ЕСЕПТЕУ ТЕОРИЯСЫ
2-БӨЛІМ

Dexter C. Kozen

Theory of Computation

With 75 Illustrations

Қазақстан Республикасы
Білім және ғылым министрлігі

ДЕКСТЕР С. КОЗЕН

ЕСЕПТЕУ ТЕОРИЯСЫ

2-бөлім

Оқулық

Алматы, 2014

ӘОЖ
КБЖ
К

Қазақстан Республикасы Білім және ғылым министрлігінің «Оқулық» республикалық ғылыми-практикалық орталығы бекіткен

Қазақ тіліне аударған:
физика-математика ғылымдарының докторы Рысбайұлы Болатбек, Рысбаева Назерке, Рысбаева Әйгерім және Рысбаева Қорлан

Козен Д.С.
К Есептеу теориясы. 2-бөлім: Оқулық / ауд. Б. Рысбайұлы, т.б. – Алматы, 2014. – 320 бет.

ISBN

Бұл, американдық ғалым әрі ұстаз Декстер Козеннің көп жылдық еңбегінің нәтижесінде дүниеге келген "Есептеу теориясы" оқулығының екінші бөлімі. 2013 жылы оқулықтың бірінші бөлімі жарық көрген еді. Ағылшын тілді, рейтингі өте жоғары университеттерде кең түрде қолданыс тапқан оқулық болғандықтан, министрлік шешімімен қазақ тіліне аударылып отыр.

Бұл бөлімге 30 бен 41 арасындағы лекциялар, және қосымша лекциялар F, G, H, I, J енгізілді. Бөлімде үй жұмыстары, аралас жаттығулар, үй жұмыстарының шешімдері, аралас жаттығулардың кейбіреуіне көмек, таңдап алынған аралас жаттығулардың шешімдері беріледі. Есептеу теориясының дамуы мен қалыптасуына еңбек сіңірген ғалымдар еңбегінің тізімі де келтіріледі.

Оқулық жоғары оқу орындарының студенттеріне, магистранттарына, докторанттарына және оқытушыларына арналған.

ӘОЖ
КБЖ

ISBN

Френсиске арналған

Аудармашының алғы сөзі

Декстер Козеннің "Есептеу теориясы" атты еңбегін екі бөлім етіп аудару жайлы шешім былтыр қабылданған еді. Оқулықтың бірінші бөлімі жарық көріп, оқушылардың қолына тиді. Қолдараңыздағы кітап "Есептеу теориясы" оқулығының екінші бөлімі. Алғашқы 29 лекция бірінші бөлімге енген болатын, бұл бөлімге енген лекциялар 30-дан басталып 41-мен аяқталады және қосымша лекциялар F, G, H, I және J қамтылған. Үй жұмыстары мен аралас жаттығулардың болуы бұл бөлімді ерекшелейді. Алдымен үй жұмыстары мен аралас жаттығулардың тапсырмалары келтіріледі, тапсырмалар аяқталысымен үй жұмыстарының шешімдері толығымен көрсетіледі. Ал аралас жаттығулардың кейбіреуіне көмек ретінде берілген нұсқаулардан кейін ғана таңдап алынған аралас жаттығулардың шешімдері беріледі. Оқулықта сілтеме жасалған әдебиеттерді толығымен түпнұсқада қалай жазылса солай келтірдік. Бұл бөлімнің тағы бір ерекшелігі болып табылатын "Белгілеулер және Аббревиатуралар" тобы толығымен аударылды. Оқулықтың салмақтылығын анықтайтын жай ол кітапта "Индекстеу" тобының болуы. Сондықтан түпнұсқадағы индекстеу тобы толығымен келтіріліп отыр. Түпнұсқада ғылыми баяндау тілі мен күнделікті өмірдегі қолданыстағы сөз саптаулар шебер ұштастыра баяндалған. Аудармада сол шеберлікті мүмкіндігінше сақтауға тырыстық. Қаншама шеберлік таныттық десек те, кемшілік кетіп жатады. Оқушылар тарапынан сондай кемшіліктерді байқаған жағдайда редакцияға хабарлауыңызды өтінеміз. Келесі басылым бола қалған жағдайда сол ескертпелерді пайдаланар едік.

физика математика ғылымдарының докторы,
профессор Рысбайұлы Б.

Мазмұны

ЛЕКЦИЯЛАР

Қосымша лекция F

Ерекше орындаушылық

Осы және келесі лекцияда, ықтималдық және есептеуді қамтитын қызықты келтірілімділік қатынасты зерттейтін боламыз және күрделіліктің структуралы ерекше екі нәтижесін дәлелдейміз. Оның алғашқысы Вэлианта және Вазирани [125] нәтижесі болып табылады. Ол нәтижеде бірден көп шешімі болатыны кепілденген кез-келген формула үшін логикалық орындалушылық жеңіл болмайтыны тұжырымдалады. Екіншісі Тод [120] нәтижесі болып табылады. Онда *NP*-толықтылық мәселесіндегі шешімді санай алу мүмкіндігі, полиномиалды-уақыт иерархиясындағы кез келген жиынды санауға мүмкіндік ашатыны айтылады. Бұл лекцияда Вэлиант-Вазирани нәтижесін дәлелдейтін боламыз. Ол нәтижеге Тода теоремасы біршама сүйенетін айта кетейік.

10.2-теоремада көрсетілгендей *NP-дағы* мәселелерді полиномиалды өлшемді *куәгер* (*witnesses*) арқылы анықтауға болады, сонда олар жеңіл танылатын болады. Мысалы, логикалық орындалушылық (SAT) ақиқатты қанағаттанарлық меншіктеу арқылы сипатталады.

Проблеманың әртүрлі данасы үшін куәгерлердің саны экспоненциалды диапозанда өзгере алады. Сондықтан осы өзгеріс *NP-толықтылық* проблемасының күрделілігіне кішкене болса да жауапты бола ма деп сұрақ қоюға болады.

Вэлиант және Вазирани [125] бұл сұраққа «болмайды» деп жауап берді. Олар SAT-тың жалпы орындалушылық мәселесі тиімді USAT-ке ықтимал келтірілетінін көрсетті, яғни айтылған мәселе қанағаттанарлық мешіктеу бірден аспайтын логикалық орындалушылық мәселесіне келтірілді.

Дәлелдеме келесі идеяға негізделген. Шектеулі $\mathbb{F}$ өрісінде n өлшемді $\mathbb{F}^n$ кеңістігін қарастырамыз. Келесі түрде сызықты кеңістіктердің ықтимал мұнарасын тұрғызуға болады

$$\{0\} = \mathrm{E}_0 \subset \mathrm{E}_1 \subset \mathrm{E}_2 \subset \ldots \subset \mathrm{E}_n = \mathbb{F}^n , \qquad (\mathrm{F.}\ 1)$$

мұндағы E_i -дің өлшемі i-ге тең және барлық осындай мұнаралар тең ықтималды болуы қажет. Оған қол жеткізу үшін, $\mathbb{F}^n$-нен кез келген $x_1, \mathrm{x}_2, \ldots \mathrm{x}_n$ базис алып $E_i = \{x_1, \ldots, \mathrm{x}_{n-i}\}^{\perp}$-ді анықтаймыз, мұндағы

$$A^{\perp} \overset{def}{=} \{y \mid \forall \mathrm{x} \in \mathrm{A}\ \ \mathrm{x} \bullet \mathrm{y}{=}0\}$$

таңбасы A-ның ортогонал толықтауышы және $\bullet$ арқылы ішкі көбейтінді таңбаланған. Кез келген базисті тиімді түрде тауып алуға болатыны бізге белгілі (Аралас жаттығулар 60 (b)).

Екі элементті өріс $\mathbb{F} = \mathbb{GF}_2$ үшін Вэлиант және Вазирани [125], Теорема 4 (ii)], кез келген бос емес жиын $\mathrm{S} \subseteq \mathbb{GF}_2^n$ және кездейсоқ таңдалған (F.1) мұнарасын қарастырып, кейбір $S \cap \mathrm{E}_i$ жиынында жоқ дегенде 1/2 ықтималдықпен бір вектор бар болатынын көрсетеді. Бұл жағдай жалпы келтірімділік проблемасын ерекше келтірімділік проблемасына айналдырады, яғни ықтимал келтірімділікке аударады.

[125]-те берілген бұл тұжырымның дәлелдемесі өлшемге қатысты индуктивті. Біз төменгі шегі жақсартылып 3/4 болатын алтернативті дәлелдемені келтіреміз.

F.1-лемма $\mathbb{GF}_2^n$ *-тің бос емес жиыны* S *болсын.* $\mathbb{GF}_2^n$ *кеңістігінің*

$\dim \mathrm{E}_i = \mathrm{i}$ *болатын ішкі кеңістіктері де* $E_0, \ldots, \mathrm{E}_n$ *болсын. Сонда*

$$\Pr\left(\exists \mathrm{i} \mid \mathrm{S} \cap \mathrm{E}_i \mid = 1\right) \geq \frac{3}{4} \tag{F. 2}$$

Дәлелдеу. Егер $0 \in \mathrm{S}$ болса, онда $S \cap \mathrm{E}_0 = \{0\}$ болғандықтан, ықтималдық 1-ге тең болады. Ал егер $|S| = 1$ болса, онда $|S| = |S \cap \mathrm{E}_n| = 1$ болғандықтан, тағы да ықтималдық 1-ге тең болады. Осы екі жағдайды шығарып тастап, $|S \cap \mathrm{E}_n| \geq 2$ және $|S \cap \mathrm{E}_0| = 0$ деп алуымызға болады. Барлық $0 \leq \mathrm{i} \leq \mathrm{n}\text{-}1$ үшін $S \cap \mathrm{E}_i \subseteq S \cap \mathrm{E}_{i+1}$ болғандықтан, осы $0 \leq \mathrm{i} \leq \mathrm{n}\text{-}1$ үшін $|S \cap \mathrm{E}_i| \leq |S \cap \mathrm{E}_{i+1}|$ болады. Сондықтан $|S \cap \mathrm{E}_k| \geq 2$ шарты орындалатын ең кіші $k \geq 1$ саны табылады. Мұндағы k кездейсоқ сан, оның мәні кездейсоқ таңдалған S_i және S -ке тәуелді болады.

Егер А векторлар жиыны болса, сонда $< A >$ деп А жиынының $\mathbb{GF}_2^n$ кеңістігіндегі сызықты қабықшасы белгіленсін. $\mathbb{GF}_2^n$ кеңістігінде тек үш сызықты тәуелсіз комбинация бар, атап айтсақ x, y және $x + \mathrm{y}$. Егер тек егер $x = \mathrm{y}$ болса ғана $x + \mathrm{y}$ нөлге тең болады. Соның әсерінен $\mathbb{GF}_2^n$ кеңістігінде кез келген нөлден өзгеше екі жұп векторлар x, y сызықты тәуелсіз болады. Сондықтан $\dim < S \cap E_k >$ жоқ дегенде 2-ге тең болады.

Сонымен (F.2)

$$\Pr\left(S \cap E_{k-1} = \varnothing\right) \leq \frac{1}{4}, \tag{F.3}$$

тұжырымына эквивалентті болып шықты. Біз дәлелдемекші болған тұжырым осы болатын. x_{n-k+1} векторына перпендикуляр барлық векторлардан құралған гипержазықтық $H = \{x_{n-k+1}\}^{\perp}$ болсын және

$S \cap E_k$-ның максималды сызықты тәуелсіз ішкі жиыны T болсын. Сонда $E_{k-1} = \mathrm{E}_k \cap H$ және

$$T \cap H \subseteq \mathrm{S} \cap \mathrm{E}_k \cap H = \mathrm{S} \cap \mathrm{E}_{k-1},$$

сондықтан

$$\Pr\left(T \cap H\right) \le \frac{1}{4}$$

болатынын көрсетсек жеткілікті болады.

Бірақ $\dim < \mathrm{T} > = \dim < \mathrm{S} \cap \mathrm{E}_k > \ge 2$, болатындықтан, 73-аралас жаттығуға сүйеніп

$$\Pr\left(T \cap H = \varnothing\right) \le \max_{d \ge 2} \Pr\left(T \cap H = \varnothing \,\middle|\, \dim < \mathrm{T} > = \mathrm{d}\right) \quad (\mathrm{F}.4)$$

қатынасын жазамыз. Осының өзі $d \ge 2$ болғанда

$$\max_{d \ge 2} \Pr\left(T \cap H = \varnothing \,\middle|\, \dim < \mathrm{T} > = \mathrm{d}\right) \quad (\mathrm{F}.5)$$

шектеуін қойсақ жеткілікті болатынын көрсетеді. Шектеулі өлшемді A және B ішкі кеңістіктер үшін $co\dim \mathrm{A} \cap \mathrm{B} \le \mathrm{codim}\, \mathrm{A} + \mathrm{codim}\, \mathrm{B}$ қатынасы орындалатындықтан (61-аралас жаттығу), егер $\dim < \mathrm{T} > = \mathrm{d}$ болса, онда $\dim\left(< \mathrm{T} > = \mathrm{H}\right)$-ның мәні не d немесе $d-1$ болатынын аламыз.

Тағы да 73-аралас жаттығуды пайдаланып, (F.5)-нің мәні

$$\Pr\left(T \cap H = 0 \,\middle|\, \dim < \mathrm{T} > = \mathrm{d} \wedge \dim\left(< T > \cap \mathrm{H}\right) = \mathrm{d}\right), \quad (\mathrm{F}.6)$$

$$\Pr\left(T \cap H = 0 \,\middle|\, \dim < \mathrm{T} > = \mathrm{d} \wedge \dim\left(< T > \cap \mathrm{H}\right) = \mathrm{d}\text{-}1\right) \quad (\mathrm{F}.7)$$

өрнектерінің максимумымен шектелгенін байқаймыз. (F.6)-ның мәні нөлге тең, бірақ $\dim < T > = \dim\left(< T > \ \cap\ H\right) = \mathrm{d}$ теңдігі $T \subseteq H$ қатынасын білдіреді, сондықтан $T \cap H$ құр жиын болуы мүмкін емес. Енді (F.7)-нің шегін табумен айналысамыз. Оны басқаша келесі түрде анықтауға болады. Сонымен, $d \geq 2$ жағдайында сызықты тәуелсіз векторлар жиыны Q берілсін, және ықтималды G гипержазықтық <Q> -да жатсын, әрі Q-дың ешқандай векторы G-да жатпайды делік, сонда ізделінді ықтималдық

$$\Pr\left(Q \cap G = \varnothing\right) \tag{F.8}$$

болады. $G = \left\{x\right\}^{\perp}$ орындалатын етіп, <Q> -дан x векторын алайық. Сонда (F.8)-ді

$$\Pr\left(\forall y \in Q \ \ \mathrm{y} \bullet \mathrm{x} \ \neq \ 0\right) \tag{F.9}$$

түрінде жаза аламыз. Бұл өрнекті бағалау үшін қосу-ажырату принципін қолданамыз (13-лекцияны қараңыз). Енді q элементті $\mathbb{GF}_2^n$ өрісінде

$$\begin{aligned}
\Pr\left(\forall y \in Q \ \ \mathrm{y} \bullet \mathrm{x} \ \neq \ 0\right) \ &= \ 1 - \Pr\left(\exists y \in Q \ \mathrm{y} \ \bullet \ \mathrm{x} = 0\right) \\
&= \ 1 - \left(\sum_{m=1}^{|Q|} \left(-1\right)^{m+1} \sum_{\substack{A \subseteq Q \\ |A| = m}} \Pr\left(x \in A^{\perp}\right) \right) \\
&= \ 1 - \left(\sum_{m=1}^{d} \left(-1\right)^{m+1} \binom{d}{m} \ \mathrm{q}^{-m} \right) \\
&= \ \sum_{m=0}^{d} \binom{d}{m} \ \left(-1\right)^{m} \mathrm{q}^{-m} \\
&= \ \left(1 - \frac{1}{q}\right)^{d} .
\end{aligned}$$

Бұл өрнектің d = 2 болғанда максимумы ізделінеді, q = 2 болғанда, ол 1/4-ге тең болады.

Ерекше орындалушылық және жалпы орындалушылық

RP класы (кездейсоқ полиномиалды уақыт үшін) 13-лекцияда анықталған болатын. Бұл ықтимал есептеумен полиномиалды уақытта біржақты қателікпен қабылдайтын жиындар класы еді. *RP* класы бейдетерминирлі Тьюринг машинасына ұқсас ықтимал машина арқылы анықталатынын естеріңізге сала кетейік. Ықтимал машина таңдауды бейдетерминирлі түрде емес ықтимал түрде жүргізеді, бұл оның айырмашылығы болып табылады. Әрбір таңдау жасау кезінде, машина келесі конфигурацияны біртекті ықтималдықпен кездейсоқ таңдайды.

Осыған эквивалентті жағдай: біз детерминирлі полиномиалды уақыт машинасына полином ұзындықты кездейсоқ битті көрсете аламыз, ол жолды машина есептеу барысында қабылдайды.

Ерекше орындалушылық үшін оракул USAT көрсеткен *RP* есептеу жалпы орындалушылықты кездейсоқ біржақты аз қателікпен анықтай алатынын, Вэлиант және Вазирани нәтижесінен байқаймыз. USAT оракулы, тек бір ғана қанағаттанарлық меншіктеуі бар логикалық формуладағы сұратымға оң (растап) жауап беретін oracl болып табылады. Ал қанағаттанарлық меншіктеуі жоқ логикалық формуланың сұратымына теріс жауап береді және басқа сұратымдар үшін әртүрлі жауап қайтарады.

F.2-теорема (Вэлиант және Вазарини [125]) $NP \subseteq \mathrm{RP}^{USAT}$.

Дәлелдеу. Біз USAT оракулды *RP* есептеудің көмегімен логикалық орындалушылықты қалай шешуге болатынын көрсетеміз. Яғни,

$$\varphi \text{ орындалады} \Rightarrow \Pr\left(M\text{-де } \varphi\text{-ге қабылдау-ашық}\right) \geq \frac{3}{4},$$

$$\varphi \text{ орындалмайды} \Rightarrow \Pr\left(M\text{-де } \varphi\text{-ге қабылдау-ашық}\right) = 0.$$

Эквивалентті жағдай. Кездейсоқ таңдаманы енгізу мәліметтерінің бөлігі ретінде кодтай отырып, детерминирлі полиномиалды уақытты N оракул машинасы табылатынын көрсетеміз. N машинасында, ω үшін

$$\varphi \text{ орындалады} \Rightarrow \Pr_w\left(\mathrm{N}\text{ -де } \varphi\#\omega \text{ -ға қабылдау-ашық }\right) \geq \frac{3}{4},$$

$$\varphi \text{ орындалмайды} \Rightarrow \Pr_w\left(\mathrm{N}\text{ -де } \varphi\#\omega \text{ -ға қабылдау-ашық }\right) = 0$$

шарттары орындалатын полином ұзындықты кездейсоқ биттер жолынан біртекті ықтималдықпен жол таңдап алынады.

F.1-леммадағыдай, сызықты ішкі кеңістіктің $H_i \subseteq \mathbb{GF}_2^n$ кездейсоқ мұнараларын құрастыру үшін N машинасы өзінің барлық кездейсоқ ω биттерін пайдаланады, мұндағы n саны φ-дің логикалық айнымалыларының саны. Бұл жағдай $O\left(n^2\right)$ кездейсоқ битті қажет етеді. F.1 лемманың дәлелдемесінде келтірілгендей, мұнара сызықты тәуелсіз емес n вектордың тізбесі ретінде көрсетілуі мүмкін.

φ-дің айнымалылары $x_1, \ldots, \mathrm{x}_n$ болсын делік. Әрбір $0 \leq i \leq n$ үшін N машинасы, $(x_1, \ldots, \mathrm{x}_n)$-ді $\mathbb{GF}_2^n$ -нің H_i -де жататын n- векторы деп түсінетін, ψ_i формуласын құрастырамыз. Бұл конструкция H_i -ді сипаттайтын кездейсоқ вектормен $(x_1, \ldots, \mathrm{x}_n)$ векторының скаляр көбейтіндісін тура кодтау деп түсінеміз. Осыдан кейін N машинасы конъюнкция $\varphi \wedge \psi_i$ үшін ораклдан сұратым жасатады, егер осы сұратымға кез келгеніне ораклдан “иә”деген жауап түссе, онда ол қабылдау-ашық күйге енеді.

S арқылы φ-ді қанағаттандыратын ақиқатты меншіктеу жиынын белгілейік. Сонда, F.1-леммаға сүйенсек

$$\Pr_w\left(\mathrm{N}\text{ -де } \varphi\#\omega \text{ -ға қабылдау ашық }\right) = \Pr\left(\exists i \;\; \varphi \wedge \psi_i \;\in\; \mathrm{USAT}\right)$$

болады. Егер $S \neq \varnothing$ болса, онда

$$\Pr\left(\exists i \;\middle|\; \mathrm{S} \cap \mathrm{H}_i\right| = 1\Big) \geq \frac{3}{4}$$

болады. Егер $S = \varnothing$ болса, онда ықтималдық нөлге тең.

Күшейтудің көмегімен қателікті шексіз аз жасауға болады (14.1-лемма).

Қосымша лекция G

Тод теоремасы

Бұл лекцияда Тод теоремасымен танысатын боламыз (G.1 теорема). Ол теоремада, NP толық есебінің үлгісінің шешімін санай алу, полиномиалды-уақыт иерархиясы PH-тағы кез келген шешілу мәселесін шешуге мүмкіндік беретіні айтылады. Бұған дейін, шешімнің қуатын есептей алу күшті нәтиже екендігіне ешкім назар аудармап еді. Сондықтан 1989 жылы [120, 121] жарық көрген бұл теорема, нағыз күтпеген жай болды. Бұл нәтиже әртүрлі күрделіліктер класының қатысуындағы, көптеген маңызды идеялар мен конструкциялардың жинақталуы болып табылады. Оған қоса, структуралы күрделіліктер теориясы мысалының озық үлгісі болып саналады.

Кластарды санау

Шешімді санау қуаты $P^{\#P}$ класына тиеселі болады. Біз $\#P$ класын барлық функциялар класы $f_M : \{0,1\}^* \to \mathrm{N}$ ретінде анықтаймыз. Оның қасиеті: полиномиалды уақытта жұмыс жасайтын М детерминирлі Тьюринг машинасы болады (бейдетерминирлі полиномиалды-уақытты Тьюринг машинасы) және x енгізуінде М-нің үздік есептеулерін f_M(x) функциясы қамтамасыз етеді. Оны басқаша айтса да болады. Яғни, егер тек егер барлық x $\in$ $\{0,1\}^*$ және $k \geq 0$ үшін A $\in$ P жиыны табылып

$$f(x) = \left| \left\{ y \in \{0,1\}^{|x|^k} \;\middle|\; x\#y \in \mathrm{A} \right\} \right| \qquad \text{(G.1)}$$

қатынасы орындалса, онда $f \in \#P$ болады (64-аралас жаттығу). Мысалы, берілген логикалық формулада орындалатын ақиқатты меншіктеу санын қайтаратын функция $\#P$-да жатады. Бұған дейін біз қарастырған басқа күрделілік класымен салыстырсақ, $\#P$ шешілімділік мәселесінің класына жатпайды, бірақ бүтін санды функциялар класы болып табылады. Осыған назар аударыңыздар. $P^{\#P}$ класы белгілі бір оракулды f $\in$ #P пайдаланатын, полиномиалды уақытта шешілетін шешілімділік мәселесінің класы болып табылады. $\#P$ және $P^{\#P}$ кластарын Вэлиант [124] енгізген болатын.

$\#P$ класын куәлер жиынының қуатын экзистенциалды түрде беретін функциялар жиыны ретінде қарастыруға болатынын, (G.1) формалдауы ұйғарады. Егер $p:\ \mathbb{N} \to \mathbb{N}$ және A жолдар жиыны болса, онда x#y $\in$ A қатынасы орындалып, x-тің p мен A-ға қарағандағы куәсі, ұзындығы p(|x|) болатын бинарлы жол y-ті береді. Енді x-тің p мен A-ға қарағандағы барлық куәсін $\mathrm{W}(p, A, x)$ арқылы белгілейміз. Сонда

$$\mathrm{W}(p, A, x) \overset{def}{=} \left\{ \mathrm{y} \in (0,\ 1)^{p(|x|)} \;\middle|\; \mathrm{x}\#y \in A \right\}.$$

Курста, осыған дейін қарастырылған күрделілік кластарының көпшілігін, әртүрлі параметрлер A мен p үшін құрастырылған куәлер жиынының қуаты $\mathrm{W}(p, A, x)$ арқылы анықтауға болады. Анықтаушы шарт көбіне $|\mathrm{W}(p, A, x)|$-ны толық беруді талап етпейді, бірақ тек шекара немесе белгілі бір биттерді сұрайды. Мысалы:

$$L \in NP \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x} \;\; \mathrm{x} \in \mathrm{L} \Leftrightarrow \left| W\left(n^c, A, x\right) \right| > 0$$

$$L \in \Sigma_{k+1}^p \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \Pi_k^p \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x} \;\; \mathrm{x} \in \mathrm{L} \Leftrightarrow \left| W\left(n^c, A, x\right) \right| > 0$$

$$L \in RP \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x}$$

$$\mathrm{x} \in \mathrm{L} \Rightarrow \left| W\left(n^c, A, x\right) \right| \geq \frac{3}{4} 2^{|x|^c}$$

$$\mathrm{x} \notin \mathrm{L} \Rightarrow \left| W\left(n^c, A, x\right) \right| = 0$$

$$L \in BPP \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x}$$

$$\mathrm{x} \in \mathrm{L} \Rightarrow \left| W\left(n^c, A, x\right) \right| \geq \frac{3}{4} 2^{|x|^c}$$

$$\mathrm{x} \notin \mathrm{L} \Rightarrow \left| W\left(n^c, A, x\right) \right| \leq \frac{1}{4} 2^{|x|^c}$$

$$L \in PP \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x} \;\; \mathrm{x} \in \mathrm{L} \Leftrightarrow \left| W\left(n^c, A, x\right) \right| \geq 2^{|x|^c - 1}$$

$$L \in \oplus P \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x} \;\; \mathrm{x} \in \mathrm{L} \Leftrightarrow \left| W\left(n^c, A, x\right) \right| \text{ is odd}$$

$$L \in \# P \overset{def}{\Leftrightarrow} \exists \mathrm{A} \in \mathrm{P} \;\; \exists \mathrm{c} \geq 0 \;\; \forall \mathrm{x} \;\; \mathrm{L(x)} = \left| W\left(n^c, A, x\right) \right|.$$

PP және $\oplus$ P жиындары $\left| \mathrm{W}\left(\mathrm{n}^c, A, x\right) \right|$-ның бірінші және соңғы биттері арқылы анықталып тұрғанына назар аударыңыздар.

Осыдан кейін, күрделілік класына $\mathcal{C}$ қарай, BP, R, # операторларын жалпылай аламыз. Бұл операторларды келтірімділік қатынастары ретінде қарастырса да болады. Төмендегі кестеде жинақталған анықтамаларды оқу жолымен таныстырайық. Сол жақ бағанда келтірілген күрделілік класы жиындар класы немесе функция L арқылы анықталады. Ол үшін $\mathrm{A} \in \mathcal{C}$ және и $k \geq 0$ табылып, барлық x үшін бағанның сол жағындағы шарт орындалуы қажет.

Кластар	*Анықтайтын шарттар*
$R\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Rightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert \geq \frac{3}{4}2^{\lvert x\rvert^k}$ $\mathrm{x}\notin\mathrm{L} \Rightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert = 0$
$BP\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Rightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert \geq \frac{3}{4}2^{\lvert x\rvert^k}$ $\mathrm{x}\notin\mathrm{L} \Rightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert \leq \frac{1}{4}2^{\lvert x\rvert^k}$
$P\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert \geq \frac{1}{2}2^{\lvert x\rvert^k}$
$\oplus\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert = 1(\mathrm{mod}\,2)$
$\Sigma^p\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert > 0$
$\Pi^p\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(n^k,A,x\right)\right\rvert = 2^{\lvert x\rvert^k}$
$\Sigma^{\log}\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(\left[k\log n\right],A,x\right)\right\rvert > 0$
$\Pi^{\log}\cdot\mathcal{C}$	$\mathrm{x}\in\mathrm{L} \Leftrightarrow \left\lvert W\left(\left[k\log n\right],A,x\right)\right\rvert = 2^{\left[k\log\lvert x\rvert\right]}$
$\#\cdot\mathcal{C}$	$\mathrm{L}(x) = \left\lvert W\left(n^k,A,x\right)\right\rvert$

Сондықтан $RP = \mathrm{R}\cdot\mathrm{P}$, $\mathrm{BPP} = \mathrm{BP}\cdot\mathrm{P}$, $\mathrm{PP} = \mathrm{P}\cdot\mathrm{P}$, $\oplus\mathrm{P} = \oplus\cdot\mathrm{P}$, $\mathrm{NP} = \Sigma^p\cdot\mathrm{P}$, $co-NP = \Pi^p\cdot\mathrm{P}$, $\Sigma^p_{k+1} = \Sigma^p\cdot\Pi^p_k$ және $\#\cdot P = \#\mathrm{P}$. Осы операторлардың барлығы қамту жиынына қатысты монотонды екенін айта кетейік.

Операторлар және куәлер арқылы күрделілік кластарын сипаттау, адам таңғаларлық кластар арасындағы негізгі ұқсастықтарды ашып көрсетеді. Және Тод теоремасын дәлелдеуде қажетті конструкцияда орындалатын жалпы негізбен қамтамасыз етеді. Біз мұқтаж болатын тұйықтаудың негізгі қасиеттері, осы оператордың коммутативтілік және идемпотенттілік сияқты алгебралық қасиеттерімен өрнектеледі.

Тод теоремасы

G.1-теорема (Тод [120]) $\mathrm{PH} \subseteq \mathrm{P}^{\#\mathrm{P}}$.

Тод теоремасының дәлелдемесінің негізгі бөлігі бірнеше қамтуға ажыратылады (G.2 (i) - (vi) лемма). Алдыңғы бөлімде анықталған күрделілік кластарының операторларының негізгі алгебралық қасиеттерін осы қамтулар анықтайтын болады. Олар $PH \subseteq BP \cdot \oplus P$ болатынын көрсету үшін біріктірілетін болады. Бұған қосымша $\mathrm{BP} \cdot \oplus P \subseteq P^{\#P}$ қатынасын көрсететін шектеулі ерекше аргумент табылады.

G.2 лемма $\mathcal{C}$ арқылы полиномиалды-уақыт Тьюринг келтірімділігімен $\leq_T^p$ төменнен тұйықталған күрделілік класын белгілейік. Сонда

(i) $\Sigma^p \cdot \mathcal{C} \subseteq \mathrm{R} \cdot \Sigma^{\log} \cdot \oplus \cdot \mathcal{C}$;

(ii) $\Pi^{\log} \cdot \oplus \cdot \mathcal{C} \subseteq \oplus \cdot \mathcal{C}$;

(iii) $\oplus \cdot \mathrm{BP} \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \oplus \cdot \mathcal{C}$;

(iv) $\mathrm{BP} \cdot \mathrm{BP} \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \mathcal{C}$;

(v) $\oplus \cdot \oplus \cdot \mathcal{C} \subseteq \oplus \cdot \mathcal{C}$;

(vi) $\mathrm{BP} \cdot \mathcal{C}$ и $\oplus \cdot \mathcal{C}$ төменнен $\leq_T^p$ арқылы тұйықталады.

Дәлелдеу. Мұндағы (i) пункті Вэлиант және Вазиранидің F лекциясында (F.1 лемма) баяндалған нәтижесінің екінші реттегі болымсыз нәтижесі екендігіне сенсеңіз де сенбесеңізде болады. Осы пункті тәптіштеп дәлелдейтін боламыз, ал қалған пункттерді (ii) - (vi) жаттығу ретінде қалдырамыз (66-аралас жаттығу).

Анықтама бойынша, $\mathrm{A} \in \mathcal{C}$ және $k \geq 0$ табылып, кез келген x үшін, егер тек егер

$$\mathrm{x} \in \mathrm{L} \;\Leftrightarrow\; W\left(\mathrm{n}^{k}, A, x\right) \neq \varnothing$$

орындалса, онда $\mathrm{L} \in \Sigma^{p} \cdot \mathcal{C}$ қатынасы орындалады. $N = n^k$ деп алайық. F.1-леммада қолданғандай, $\dim H_i^{\omega} = \mathrm{i}$ болатын, $\mathbb{GF}_2^N$ -ның сызықты ішкі кеңістіктерінің кездейсоқ мұнарасы H_i^{ω}, $0 \leq i \leq \mathrm{N}$ болсын. Мұнара, $\mathbb{GF}_2^N$ -нің кездейсоқ нұқсансыз (бейсингулярлы) матрицасы арқылы анықталатынын еске түсірейік, мұндағы H_i^{ω} мұнарасы матрицаның алғашқы N-i бағанының ортогонал толықтауышы болады. Кезегінде, кездейсоқ нұқсансыз матрица ұзындығы $O(N^2)$ болатын кездейсоқ биттерден құралған ω жолынан тұрады. F.1-лемма бойынша,

$$\mathrm{x} \in \mathrm{L} \;\Rightarrow\; W(\mathrm{N}, A, x) \neq \varnothing$$

$$\Rightarrow\; \Pr_{\omega}\left(\exists\, i \leq N \;\middle|\; W(N, A, x) \cap \mathrm{H}_i^{\omega} \;\middle|\; = 1\right) \geq \frac{3}{4}$$

$$\Rightarrow\; \Pr_{\omega}\left(\exists\, i \leq N \;\middle|\; W(N, A, x) \cap \mathrm{H}_i^{\omega} \;\middle|\; \text{is odd}\right) \geq \frac{3}{4},$$

$$\mathrm{x} \notin \mathrm{L} \;\Rightarrow\; W(\mathrm{N}, A, x) = \varnothing$$

$$\Rightarrow\; \Pr_{\omega}\left(\exists i \leq N \;\middle|\; W(N, A, x) \cap \mathrm{H}_i^{\omega} \middle|\; \text{is odd}\right) = 0.$$

Енді $p(x\#\omega\#i) = |x|^k$ орындалатын етіп, p функциясын аламыз. Ал i санын $[\,\mathrm{k}\log|x|\,]$ битті екілік жүйеде өрнектейміз деп шарт қойып, p функциясын тек $x\#\omega\#i$-ның ұзындығынан ғана тәуелді болатын етуге болады. Осы тектес, $q(x\#\omega) = \left[k\log|x|\right]$ орындалатын етіп, q функциясын аламыз. Анықтаймыз

$$A' \overset{def}{=} \left\{ x\#\omega\#i\#y \;\middle|\; \mathrm{x}\#y \in A \wedge y \in H_i^{\omega} \right\}$$

$$A'' \overset{def}{=} \left\{ z \;\middle|\; \left| W(p, A', z) \right| \text{ is odd} \right\}$$

$$A''' \overset{def}{=} \left\{ \mathrm{u} \;\middle|\; \left| W(p, A'', \mathrm{u}) \right| > 0 \right\}.$$

A $\in \mathcal{C}$ және $A' \leq_T^p$ A болғандықтан, кластардың анықтамасын пайдаланып $\mathrm{A}' \in \overline{\mathcal{C}}$, $\mathrm{A}'' \in \oplus \cdot \mathcal{C}$ және $\mathrm{A}''' \in \Sigma^{\log} \cdot \oplus \cdot \mathcal{C}$ деп жаза аламыз. Сонда

$$\begin{aligned} W(p, A', x\#\omega\#i) &= \left\{ y \in \{0,\ 1\}^{|x|^k} \;\middle|\; x\#\omega\#i\#y \ \in A' \right\} \\ &= \left\{ y \in \{0,\ 1\}^{|x|^k} \;\middle|\; x\#y \ \in A \right\} \cap H_i^{\omega} \\ &= \mathrm{W}(N,\ A,\ x) \cap H_i^{\omega} \end{aligned}$$

сондықтан

$$\begin{aligned} &\exists\, \mathrm{i} \leq \mathrm{N} \left| \mathrm{W}(N, A, x) \cap H_i^{\omega} \right| = 1 \\ &\qquad \Rightarrow \ \exists\, \mathrm{i} \leq \mathrm{N} \left| \mathrm{W}(N, A, x) \cap H_i^{\omega} \right| \text{ тақ} \\ &\qquad \Leftrightarrow \ \exists\, \mathrm{i} \leq \mathrm{N} \left| \mathrm{W}(N, A', x\#\omega\#i) \right| \text{ тақ} \\ &\qquad \Leftrightarrow \ \exists\, \mathrm{i} \leq \mathrm{N}\ x\#\omega\#i \ \in \mathrm{A}'' \\ &\qquad \Leftrightarrow \ \left| \left\{ \mathrm{i} \in \{0,\ 1\}^{\lceil k \log|x| \rceil} \;\middle|\; x\#\omega\#i \ \in \mathrm{A}'' \right\} \right| > 0 \\ &\qquad \Leftrightarrow \ \left| \mathrm{W}(q, A'', x\#\omega) \right| > 0 \\ &\qquad \Leftrightarrow \ x\#\omega \ \in \mathrm{A}'''. \end{aligned}$$

Осының әсерінен

$$\begin{aligned} \mathrm{x} \in \mathrm{L} \ &\Rightarrow \ W(\mathrm{N}, A, x) \ \neq \ \varnothing \\ &\Rightarrow \ \Pr_{\omega}\left(\exists\, i \leq N \left| W(N, A, x) \ \cap \ \mathrm{H}_i^{\omega} \right| = 1 \right) \ \geq \ \frac{3}{4} \\ &\Rightarrow \ \Pr_{\omega}\left(\ x\#\omega \ \in \mathrm{A}''' \ \right) \ \geq \ \frac{3}{4}, \\ \mathrm{x} \notin \mathrm{L} \ &\Rightarrow \ W(\mathrm{N}, A, x) = \varnothing \\ &\Rightarrow \ \forall\, \mathrm{i} \leq \mathrm{N} \left| \mathrm{W}(N, A,\ x) \cap H_i^{\omega} \right| \text{ жұп} \\ &\Rightarrow \ \Pr_{\omega}\left(\ x\#\omega \ \in \mathrm{A}''' \ \right) = 0\,. \end{aligned}$$

Сонымен $\mathrm{L} \in \mathrm{R} \cdot \Sigma^{\log} \cdot \oplus \cdot \mathcal{C}$.

G.1-теоремасының дәлелдемесі. Алдымен, G.2 лемманың әртүрлі қамтулары бірігіп, PH $\oplus$ BP$\cdot$ $\oplus$ P орындалады деп жорамалдайтынын дәлелдейміз. Полиномиалды уақыт иерархиясы деңгейіндегі индукциямен дәлелдеу. $\Sigma_0^p = \mathrm{P} \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$ орындалатыны күмәнсіз. Енді $\Sigma_0^p = \mathrm{P} \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$ болады деп жорамал қоямыз. G.2 (vi) леммаға сүйеніп, BP $\cdot$ $\oplus$ P толықтауышқа қатысты тұйық деп айта аламыз, сондықтан $\Pi_k^p \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$. Келесіде

$$\Sigma_{k+1}^p = \Sigma^p \cdot \Pi_k^p$$

$\subseteq$ $\Sigma^p \cdot BP \cdot \oplus P$ - монотондықтан

$\subseteq$ $\mathrm{R} \cdot \Sigma^{\log} \cdot \oplus \cdot BP \cdot \oplus P$ - G.2(i) леммадан

$\subseteq$ $\mathrm{R} \cdot \oplus \cdot BP \cdot \oplus P$ - G.2(ii) және (vi) леммалардан

$\subseteq$ $\mathrm{BP} \cdot \oplus \cdot BP \cdot \oplus P$ өйткені $\mathrm{R} \cdot \mathcal{C} \subseteq \mathrm{BP} \cdot \mathcal{C}$ тривиал

$\subseteq$ $\mathrm{BP} \cdot BP \cdot \oplus \cdot \oplus P$ - G.2(iii) леммадан

$\subseteq$ $BP \cdot \oplus P$ - G.2(iv) және (v) леммалардан.

Барлық k үшін $\Sigma_2^k \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$ болатынын көрсеттік, сондықтан

$$PH = \bigcup_k \Sigma_k^p \subseteq \mathrm{BP}\cdot\oplus\mathrm{P}$$

орындалады.

Енді BP $\cdot$ $P \oplus P^{\#P}$ орындалатынын дәлелдеу ғана қалды. $L \oplus BP \cdot \oplus P$ болсын. Сонда, барлық x үшін $A \in \oplus P$ және $k \geq 0$ табылып,

$$x \in L \Rightarrow \Pr\left(\mathrm{x}\#\omega \in \mathrm{A}\right) \geq \frac{3}{4}$$

$$x \notin L \Rightarrow \Pr\left(\mathrm{x}\#\omega \in \mathrm{A}\right) \leq \frac{1}{4}$$

ω - ұзындығы $|x|^k$ болатын барлық бинарлық жолдан ықтимал біртекті таңдап алынады. Эквивалентті,

$$x \in L \Rightarrow \left| W\left(n^k, A, x\right) \right| \geq \frac{3}{4} \cdot 2^{|x|^k}$$

$$x \notin L \;\Rightarrow\; \left| W\left(n^k, A, x\right) \right| \;\le\; \frac{1}{4} \cdot 2^{|x|^k}. \tag{G.2}$$

Бұған қосымша, $A \in \oplus P$ болғандықтан, егер тек егер $x\#\omega \in$ A болса, онда $f(x\#\omega)$ тақ болатын полиномиалды уақытты бейдетерминирлі Тьюринг машинасы M табылады. Мұндағы $f(x\#\omega)$ - енгізу $x\#\omega$-да[1] M машинасының табысты жасалған есептеу жолдарының саны.

Енді жаңа N машинасын алу үшін M машинасына өзгерту енгіземіз. N машинасында $x\#\omega$ енгізуі үшін бұрынғы табысты есептеулер саны $f(x\#\omega)$-тің орнына $p\left(f(x\#\omega)\right)$-ны аламыз, мұндағы p - $\mathbb{N}\left[z\right]$ -тің белгілі бір көпмүшелігімен өрнектелген полиномиалды функция. Жеке жағдайда, бізді қызықтыратын p полиномы $h^m(z)$-де жатады, мұндағы

$$h(z) \overset{def}{=} 4z^3 + 3z^4 \;\in \mathbb{N}\left[z\right],$$

$h^i(z)$ - h-тын өзіне өзі i-рет жасаған композициясы, яғни

$$h^0(z) \overset{def}{=} z$$

$$h^{i+1}(z) \overset{def}{=} \mathrm{h}\left(h^i(z)\right)$$

және $m = \log(n^k + 1)$. $h^m(z)$ -нің дәрежесі $4^m = \left(n^k + 1\right)^2$ болатынын және $\frac{2}{3}\left(4^m - 1\right) = \frac{2}{3}\left(\left(n^k+1\right)^2 - 1\right)$ санынан аспайтын битпен өрнектелетін коэффициенттері болатынын индукцияны қолданып дәлелдей аламыз. Бұдан басқа, $x\#\omega$ -ны пайдаланып полиномиалды уақытта $h^m(z)$ көпмүшелігін құрып шығуға болады. M-нен N-ді құрастыру конструкциясы 67 (d) аралас жаттығуда жете баяндалған.

Енді $h^m(z)$-ның келесі жарамды қасиеттерін индукциямен дәлелдеуге болады:

$$z \text{ тақ} \;\Rightarrow\; \mathrm{h}^m(z) \equiv -1 \left(\operatorname{mod} 2^{2^m}\right)$$

$$z \text{ жұп} \;\Rightarrow\; \mathrm{h}^m(z) \equiv 0 \left(\operatorname{mod} 2^{2^m}\right)$$

---

[1] *Бұл $\oplus$ P-ның сипаттамасы және куәлер жиындарын қамтитын сипаттамалар эквивалентті; ол #P үшін анықталған сәйкес эквиваленттіліктен бірден алынады (64- аралас жаттығу).*

(65-аралас жаттығу), сондықтан

$$z \text{ тақ} \Rightarrow \ \mathrm{p}(z) \equiv -1 \left(\bmod 2^{n^k+1}\right) \qquad \text{(G.3)}$$

$$z \text{ жұп} \Rightarrow \ \mathrm{p}(z) \equiv 0 \left(\bmod 2^{n^k+1}\right).$$

Енді $P^{\#P}$ есептеу арқылы L-де жатушылықты келесі түрде анықтаймыз. Ұзындығы n -ге тең x енгізуінде жаңадан K машинасын құрастырамыз. Ол жағдайда:

(i) $|\omega| = \mathrm{n}^k$ болатын етіп, бейдетерминирлі тармақпен барлық мүмкін $x\#\omega$ жолдарды құрастырады және әр жолға жалғыз есептеу жолы белгіленеді;

(ii) әрбір $x\#\omega$ үшін, $x\#\omega$-ді пайдаланып N -ді іске қосады.

Енді x енгізуі үшін К-ның табысты есептеу жолдарының саны

$$\sum_{|\omega|=n^k} p\left(f\left(x\#\omega\right)\right)$$

болады. Бұл шама 2^{n^k+1} модулінде келесідей:

$$\sum_{|\omega|=n^k} p\left(f\left(x\#\omega\right)\right) \equiv \sum_{\substack{|\omega|=n^k \\ f(x\#\omega)\text{ odd}}} -1 \qquad \text{(G.3)}$$

тен алынады

$$\equiv 2^{n^k+1} - \left|\ \left\{\omega \mid |\omega| = \mathrm{n}^k \wedge f\left(x\#\omega\right) \text{ odd} \right\}\ \right|$$

$$\equiv 2^{n^k+1} - \left|\ \left\{\omega \mid |\omega| = \mathrm{n}^k \wedge x\#\omega \in \mathrm{A} \right\}\ \right|$$

$$\equiv 2^{n^k+1} - \left|\ \mathrm{W}\left(n^k, A, x\right)\ \right|.$$

Десек те (G.2)-ге сәйкес,

$$
\begin{aligned}
x \in L \;&\Rightarrow\; \frac{3}{4}\cdot 2^{n^k} \le \left|W\left(n^k, A, x\right)\right| \le 2^{n^k} \\
&\Rightarrow\; 2^{n^k} \le 2^{n^k+1} - \left|W\left(n^k, A, x\right)\right| \le \frac{5}{4}\cdot 2^{n^k} \\
x \notin L \;&\Rightarrow\; 0 \le \left|W\left(n^k, A, x\right)\right| \le \frac{1}{4}\cdot 2^{n^k} \\
&\Rightarrow\; \frac{7}{4}\cdot 2^{n^k} \le 2^{n^k+1} - \left|W\left(n^k, A, x\right)\right| \le 2^{n^k+1},
\end{aligned}
$$

және бұлар өзара қиылыспайтын кесінділер. Сонымен, 2^{n^k+1} модулімен азайтылған К-ның табысты есептеулер саны L- де детерминирлі жатады. Оны $L \in P^{\#P}$ қатынасы орындалады деп айтамыз.

30-лекция

Схеманың төменгі шекарасы және релятивтелген PSPACE = PH

Осы және бұдан кейінгі лекцияларда, біз тереңдігі тұрақты логикалық схема тобының төменгі шекарасын зерттейміз және олардың релятивті күрделілікте қолданысымен танысамыз. Нәтижелер дизайндар көзқарасы тұрғысынан және алгоритмдерді талдаумен ғана қызықты емес және полиномиалды уақыттағы иерархия құрылымының әсерімен де өзіне назар аудартады.

Төменгі шекаралар жайлы алғашқы нәтиже 1980 жылдың ортасында Furst, Saxe, мен Sipser [46] және Ajtai [5] еңбектерінде алынды. Олар PARITY функциясы үшін полиномиалды уақытты және тұрақты тереңдікті схемалар тобының жоқ екенін дәлелдеді, осындай нәтиже басқа да функциялар үшін көрсетілді. Осыдан кейін жарияланған жұмыстар сериясында, кейбір функциялар үшін тұрақты тереңдікті схемалар өлшемінде дәл супер полиномиалды төменгі шекара көрсетіліп берілді, осының әсерінен тиімділікке өте жақын нәтиже, Håstad [56] дүниеге келді. Осы теорияның баяндалуы мен толық тарихы шолғыншы мақалада, Boppana және Sipser [20] жақсы жазылған.

Бұл лекцияда логикалық функцияларға жататын күрделілік класы AC^0 қарастыруға ұсынылады, ол тұрақты тереңдік тобында, полиномиалды өлшемді схемалар мен шексіз indegree енгізулерде есептелінуі мүмкін. Кейіннен, тұрақты тереңдікті схемалардың жеткілікті қатаң супер полиномиалды төменгі шекарасы дегеніміз, ол PH-тың PSPACE oracle бөлігі бар дегенді білдіретінін көрсетеміз. Бұл қатынас бірінші рет Furst, Saxe, және Sipser [46] жұмысында зерттелді және тұрақты тереңдікті схемаларды зерттеуге алғашқы қызықтырушы болып табылады.

Лекция соңында PARITY-дің AC^0-де жатпайтындығын дәлелдейтін механизмді құрастырамыз, ол дәлелдемені деталдарымен тереңдетіп, 31-лекцияда қарастырамыз. Біздің дәлелдемеміз, *кездейсоқ жеке бағалау* идеясын ұсынған Furst, Saxe, and Sipser [46] ұсынған бастапқы дәлелдеудің біршама өзгерген варианты болады. Бұл, алынған қорытынды ықтималды болмаса да, төменгі шекара қасиеттерін дәлелдеуде ықтималдықтың бірінші рет қолданысы.

Өкішішке орай, PH-тан PSPACE –ті oracle бөліп алатындай бұл нәтиже жеткілікті дәрежеде қатаң емес. Қатаң шекаралық жеткілікті бөлікті алғаш рет алған ғалымдар Yao [126] және Håstad [56]. Håstad нәтижесінің толық дәлелдемесі H қосымша лекцияда берілген.

Жұптық және AC^0 класс

Жұптық функция ол енгізу деректерінің біреуі өзгерсе, мәні дереу өзгеретін $\{0, 1\}^n \rightarrow \{0, 1\}$ логикалық функция. Кез келген *n* үшін осы қасиетке ие болатын екі функция бар, дәлірек айтсақ, өзінің енгізуінің модуль 2-дегі қосындысын есептейтін функция және айтылған функция нәтижесінің толықтауышын есептейтін функция. Осы функцияға коллективті түрде PARITY деп сілтеме жасаймыз.

Біз біртекті логикалық схемалар тобымен алдынғы 11 және 12 лекцияларда танысқан едік. Осы лекцияларда қарастырылған NC күрделілік класы, полиномды логорифмдік тереңдік және полиномдық өлшемде, logspace біртекті логикалық схемамен анықталғанын еске түсірейік. Осында, дәрежеге шектеулер қойылған еді: схемалар қақпасы $\wedge$ және $\vee$ бинарлы, немесе унитарлы $\neg$ қақпа болып еді.

AC^0 класын анықтау үшін біз тұрақты тереңдікпен шектелеміз, бірақ ∧ және ∨ қақпалар бір қадамда, сәйкес конъюкция және дизъюнкция амалдарын есептей алатындай етіп, дәрежеге қойылатын шектеуді әлсіретеміз. Аталған амалдар логикалық шексіз енгізу деректерінде жататын болады. (Енгізілген барлық деректерге тәуелді, шектеулі жалғыз шығаруда анықталған ешқандай функция, тереңдігі $o\left(\log n\right)$ болатын шектеулі дәрежелі схемада есептелуі мүмкін емес екендігін жеңіл дәлелдеуге болады).

Мысалы, тиянақты k үшін 1-ді қайтаратын логикалық функция, егер k-дан көп емес енгізулер үшін 1-ді қайтарса, онда оны өлшемі $\binom{n}{k}+1$ ал тереңдігі 2-ге тең AC^0 схемасының көмегімен есептеуге болады:

$$\bigvee_{\substack{S\subseteq\{x_1,\ldots,x_n\}\\ |S|=k}} \wedge S .$$

NC-ны анықтаған сияқты AC^0-ны ашық түрде анықтау бірқалыпты шартты қажет етеді. Дегенмен, бұл жерде біз тек төменгі шекараға қызығып отырмыз, және біз бейқалыпты схемамен анықтайтын кез келген төменгі шекара, бейқалыпты схеманың қасиеттерін сақтайды, сондықтан мұндай жағдайларды саналы түрде ескермейміз.

Біз жалпылықты бұзбай отырып барлық теріске шығару тек енгізу деректеріне қолданылады деп шарт қоямыз және қақпалар енгізу деректерінің деңгейіне сай орналасқан және теріске шығару 0-деңгейде, конъюнкция мен дизъюнкция қатаң түрде деңгейден деңгейге өткенде кезекпен ауысып отырады. Теріске шығару жайлы ұйғарымды, 12-лекцияда көрсетілгендей дуалды схема құрастыру арқылы іске асыруға болады. ∧ және ∨ амалдарының кезектесуі жайлы ұйғарым, қажет болса, тереңдік пен уақытты көп арттырмай-ақ, жалған қақпаларды қосу арқылы іске қосылады.

PH-ты PSPACE-тен ажырату

Осы лекцияның негізгі нәтижесі: PARITY-де тұрақты тереңдіктегі схема өлшемінде қажетті қатаң төменгі шекара, иерархиялы полиномды уақытта oracle-де PSPACE үлестірім бар деп санайды.

30.1 теорема (Furst, Saxe, және Sipser [46]). Кез келген c және d үшін, PARITY-*де тереңдігі d, ал өлшемі* $2^{(\log n)^c}$ *болатын схемалар тобы жоқ деп ұйғарамыз. Сонда* $PH^A \neq \mathrm{PSPACE}^A$ *болатындай oracle* $A \subseteq \{0, 1\}^*$ *табылады.*

Дәлелдеме. $A \subseteq \{0, 1\}^*$ болсын. A-да жататын сипаттама теңдеуді де A деп белгілейміз, бұл функция

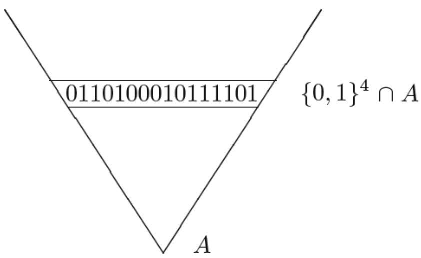
$$A(x) \overset{def}{=} \begin{cases} 1, & \text{if } x \in \mathrm{A} \\ 0, & \text{otherwise.} \end{cases}$$

Жиынды анықтаймыз

$$P(A) \overset{def}{=} \left\{ x \in \{0, 1\}^* \;\middle|\; \sum_{y \in A,\, |y| = |x|} \mathrm{A}(y) \equiv 1 \pmod 2 \right\}$$

$$= \left\{ x \in \{0, 1\}^* \;\middle|\; \text{A -дағы ұзындығы } |x| \text{ болатын жол саны тақ сан} \right\}.$$

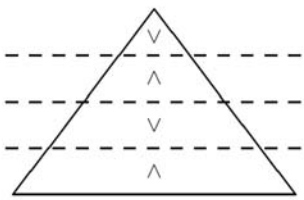
Төменгі диаграммада көрсетілген A жиыны үшін P(A) ұзындығы 4-ке тең барлық жолдарды қамтиды, оның себебі ұзындығы 4-ке тең A-да жататын жолдардың саны 9, ал ол тақ сан.



Мұндағы P(A) жиынында қабылдау-ашық кез келген oracle A үшін oracle-ды Тьюринг машинасы табылады, және ол машина сызықты кеңістікте жұмыс жасайды. Машина, енгізу x үшін ұзындығы $|x|$ -ке тең барлық жолмен белгілі бір тәртіппен өтіп шығады және әрқайсысына сұратым жасатады, және mod 2-ге қатысты оң жауаптарды тізіп отырады. Сондықтан кез келген A үшін $P(A) \in \mathrm{PSPACE}^A$.

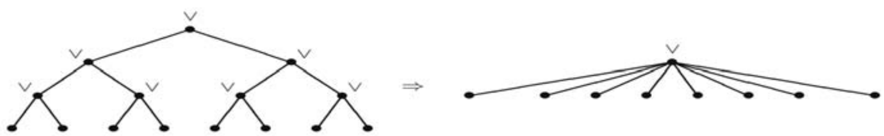
Енді, oracle кез келген машина үшін, егер M кез келген oracle A және енгізу дерегі x-пен жабдықталған болса, онда M машинасы *n-де корректі* деп айтамыз. Мұнда, егер тек егер $x \in P(A)$ болса, онда M машинада x-ке қабылдау-ашық; яғни кез келген A және ұзындығы n-ге тең болатын x енгізу дерегі үшін орындалады. M машинасы A-ның ұзындығы *n*-ге тең элементтер санына қарай жұп корректі есептейді. Біз барлық жағдайда корректі, бірақ n-мен шектеулі жиындар үшін, полиномиалды уақытты Σ_d машина жоқ деп санаймыз, кері жағдайда теорема ұйғарымын бұзатын схемалар тобын құрастыра аламыз.

Кері жағдайды туындату үшін осындай машина бар деп ұйғарайық. Ұзындығы *n*-ге тең енгізу деректері үшін, M машинасы n^c уақыт жұмыс жасайды және ұзындығы d-дан аспайтын алтернатив универсалды және экзистенциалды күйге кез келген есептеу жолында енеді, мұндағы c пен d тұрақты сандар. Кез келген енгізу дерегіне байланысты M машинасына тиеселі есептеу бұтағын, d деңгейге келесі шартпен бөлуге болады: әрбір деңгейдегі конфигурация не барлығы универсал конфигурация, не барлығы экзистенциалды конфигурация, және бірінші деңгей тек экзистенциалды конфигурацияны қамтиды.



Жалпылықты сақтай отырып, *x* енгізу дерегінде M машинасы өзінің oracle-ын, ұзындығы *x*-ұзындығынан өзгеше кез келген жолда ешқашан сұратпайды деп ұйғарамыз, өйткені осы жолдар *x* енгізуі P(A)-да жата ма деген анықтамаға сыймайды. Егер M осыны жасаса, онда біз басқа машина N-ді құрастыра аламыз. Ол машинаның қызметі мынада, егер M машинасы ұзындығы ауытқыған жолда oracle сұратым жасағысы келсе, онда N машинасы M-ді модельдеп кез келген ақиқаттықты беріп отырады. Егер M машинасы *n* корректі болса, онда N машинасы да *n* корректі болады.

Біз енді әрбір *n* үшін M корректі болатындай етіп C_n жай схеманы тұрғызамыз. O^n енгізу деректерінде N-нің есептеу бұтағынан және A-ның барлық мүмкін oracle-дерінен, C_n схемасы құрастырылған. Қақпа дегеніміз ол O^n енгізуіндегі N машинасында жататын конфигурация; мұнда белгілі бір тұрақты c үшін 2^{n^c} -санынан аспайтын қақпа бар. Енгізу деректері $A(y)$, $|y| = \mathrm{n}$ жататын, ақиқаттықты көрсететін логикалық айнымалы. Осындай 2^n енгізу бар. Бастапқы қақпа ол N-де жататын алғашқы конфигурация. Oracle сұратым сұратым жолына сәйкес келетін енгізу қақпасын оқиды. Экзистенциалды конфигурация α шектелмеген жартылай дәрежелі V-қақпа, ол барлық универсалы және тоқтатылған конфигурациялардың β енгізу деректерін алады. Мұнда $\alpha \rightarrow \beta$ есептеу жолы бар және барлық конфигурация экзистенциалды болады, кейде тоқтатылған конфигурация экзистенциалды болмай қалуы мүмкін. Айтылған жолмен, универсалды конфигурация шектелмеген indegree Λ-қақпа болады. Сондықтан есептеу бұтағының әрбір экзистенциалды және әрбір универсалды деңгейін, тереңдігі 1-ге тең схемаға теңестіреміз:



Тереңдігі c және өлшемі болатын 2^{n^c} соңғы C_n схеманың 2^n енгізуі бар және өзінің енгізу деректерінің жай болуын есептейді. Егер өлшем енгізу деректерінің санымен $m = 2^n$ анықталса, онда ол өлшем $2^{(\log m)^c}$ санына тең. Біз енгізу деректерінің өлшемі 2-нің дәрежесі ретінде өрнектелетін схемаларды ғана қарастырамыз, Бірақ біз енгізу деректерінің басқаша m өлшемінде, өлшемді келесі 2-нің дәрежесіне ұлғайтып, барлық m өлшемді енгізулерді 0-ге теңестіріп, жай схема алуымызға болады. Сонымен біз PARITY үшін, тұрақты d тереңдікті және $2^{(\log m)^c}$ өлшемді схемалар тобын алдық. Бұл жағдай теорема ұйғарымына қайшы.

Енді осыны пайдаланып, PH-ты PSPACE-тен ажырататын oracle A-ны құрастырамыз. Біз қайшылықтан, M машинасы барлық шектеулі жиынды n үшін коррект болады деген ұйғарымның қате екендігін білдік. Сондықтан, мұнда M машинасы бейкоррект болатын үлкен n табылуы қажет. Диогоналдау жолымен жалғастырамыз, барлық тұрақты d үшін M_0, M_1, ... барлық полиномиалды уақытты Σ_d машинасының тізімі болсын. Біз А-ны этаппен құрастырамыз. Яғни, i-ші стадияда (сатыда), бұған дейінгі барлық стадиядағы таңдалған сандардан үлкен және n_i-де M_i бейкорректі болатын, n_i санын алайық. 0^{n_i} енгізуінде M_i бейкорректі жауап беретін oracle-ді B_i деп белгілейік, қосымша $A_i = B_i \cap \{0, 1\}^{n_i}$ және $A = \bigcup_i A_i$ болсын. Сонда А oracle-ды M_i машинасы n_i-де бейкорректі болады, өйткені ол А-дан қандай жауап алса, B_i-ден де сондай жауап алады. Сондықтан ешқандай M_i машинасы Р(А)-ны есептемейді.

PARITY үшін төменгі шекара

31-лекцияда біз AC^0 -де PARITY жатпайды деген Furst, Saxe және Sipser [46] нәтижесімен танысатын боламыз. Оның дәлелдемесі егер жай схеманың кейбір енгізу деректерін 0 немесе 1 мәндерін қабылдаса, онда соңғы схема басқа енгізулерге көңіл бөлместен жай функцияны есептей береді. Егер бізде тұрақты d тереңдікті схема бар болса, онда кездейсоқ және тәуелсіз түрде әрбір енгізу дерекке 0 немесе 1-ді меншіктейміз. Меншіктеудің әрқайсысы анықталған ықтималдықпен $(1 - p) / 2$ теңестіріледі, екінші жағынан бұл *1/2*-ден қатаң кіші, немесе оны p ықтималдықпен ештеңеге теңестірмейміз. Бұл *кездейсоқ жеке бағалау* деп аталады. Мұндағы p-ны абайлап таңдай отырып, айнымалылардың белгіленген мәнін үлкен пайызбен сақтай отырып, тереңдігі d-дан кем схемаға эквивалентті екендігін дәлелдеуге болады. Осы процесті қайталай отырып бірінші деңгейде indegree-і шектеулі, тереңдікті 2-ге дейін азайта аламыз; бірақ біз тәуелсіз түрде ешқандай схема тобы PARITY-ды есептей алмайтынын дәлелдей аламыз.

Байқасақ, қақпа дәрежесі және схема тобының бірқалыптылығы мұндай проблема тудырмайды екен. Шексіз indegree-лі қақпаларды қамтитын бейқалыпты схема үшін де және бастапқы indegree шектесіз болса да, төменгі шекара өзгермейді екен.

31-лекция

Тұрақты терең схеманың төменгі шегі

Бұл лекцияда PARITY -дің AC^0 -де жатпайтындығын көрсететін Ферст, Саксен және Сипсер [46] нәтижесін егжей-тегжейлі баяндайтын боламыз.

Егер формула немесе схема клоздар (сөйлемдер) конъюкциясы болып табылса, онда ол t-конъюнктивті қалыпты форма (t-*CNF*) деп аталатынын еске түсірейік. Мұндағы әрбір клоз t -дан көп емес литералдан тұратын дизъюнкция, ал әрбір литерал айнымалы немесе оның теріске шығарылуы. Дуальды түрде, егер формула немесе схема мүшелер дизъюнкциясы болып табылса, онда ол t-дизъюниктивті қалыпты форма (t-*DNF*) деп аталады. Мұндағы әрбір мүше t литералдан аспайтын конъюнкция болуы шарт. Ешқандай мүше немесе клоз (сөйлем) екі өзара кері литералды (контрарлы литералдар) қамтымайды деп шарт қоямыз. Келісім бойынша бос конъюнкция 1-ге эквивалентті, ал бос дизъюнкция 0-ге эквивалентті.

Егер тереңдігі тұрақты d болатын жұптыққа тексеру схемасы берілсе, онда жалпылық ережесін сақтай отырып, қақпалар айнымалылармен бір деңгейде және олардың керіге шығарылуы 0-ші деңгейде орналасқан деп ұйғарамыз, ал басқа деңгейлер дизъюнкция және конъюнкцияның арасында кезектеседі дейміз. Және де қақпа 1-ші деңгейде дизъюнкция болады деп ұйғарамыз, болмаса, оның орнына жұптыққа тексеру дуалды схемасын қарастыруға да болады.

М мүшесінің немесе клозының ұзындығын $|M|$ арқылы белгілейміз, ол M-дегі литералдар санын береді. Біз мүшелердегі $\wedge$ таңбасын жиі тастап жазатын боламыз және $\neg x$ орнына $\overline{x}$ деп жазамыз. Сондықтан $\overline{x}y\overline{z}$ өрнегі мен $\neg x \wedge y \wedge \neg z$ өрнегі бірдей.

X логикалық айнымалылар жиынының дара бағалауы деп X жиынының кейбір айнымалыларына мәндер меншіктеу, ал кейбір айнымалыларға мәндерді меншіктемеуді (мүмкін) айтамыз. Формалды түрде, X логикалық айнымалылар жиынының дара бағалауын, кез келген $x \in X, \rho(x) \in \{x, 0, 1\}$ үшін анықталған $\rho : X \to X \cup \{0,1\}$ бейнелеуі деп түсінеміз. Егер $\rho(x) = x$ болса, онда x айнымалысы ρ -да меншіктелмеген деп айтамыз.

X-тегі кез келген ρ дара бағалауын X-тегі логикалық формулалар функциясына табиғи түрде ауыстырылады. Ол үшін әрбір x -ті $\rho(x)$ -пен алмастырамыз, одан кейін мүмкін жерлерде логикалық алгебра аксиомаларын $0 \vee x = x$, $0 \wedge x = 0$, $1 \vee x = 1$, $1 \wedge x = x$ пайдаланып ықшамдаймыз. Мысалы, егер $\rho(x) = 1$, $\rho(y) = 0$ және $\rho(z) = z$ болса, онда

$$\rho\Big((x \vee y) \ \wedge \ (\overline{x} \vee z)\Big) = z.$$

Осы лекцияның қалған бөлігінде, кездейсоқ таңдалған дара бағалауларды қарастырамыз. Мұнда әрбір айнымалыға $\left(1 - 1/\sqrt{n}\right)/2$ ықтималдықпен 0 немесе 1 меншіктеледі немесе $1/\sqrt{n}$ ықтималдықпен мән меншіктелмей қалады.

Дәлелдеудің орталық идеясы мынада: 2-деңгейдегі барлық CNF ішкі схемалар, осы үлестірімге сай келетін шектеулі рет кездейсоқ жасалған дара бағалаудан кейін, DNF схемасына үлкен ықтималдықпен эквивалентті болады. Сонымен 2-деңгейдегі барлық CNF қақпаларды DNF қақпалармен алмастырып шығудың мүмкіндігі өте үлкен. Осыдан кейін 2-деңгейдегі дизъюнкцияны 3-деңгейдегі дизъюнкциямен алмастырамыз, яғни тереңдікті бір деңгейге азайтамыз. Осы әдісті қолдана отырып, екі деңгейден басқа деңгейлерді толығымен жойып шыға аламыз.

31.1-лемма. Дара кездейсоқ бағадан кейін, меншіктелмеген айнымалылардың санының $\sqrt{n}/2$-дан аз болу ықтималдығы $(2/e)^{\sqrt{n}/2}$ санынан көп емес.

Дәлелдеу. Әрбір айнымалы $1/\sqrt{n}$ ықтималдықпен меншіктелмей қалады. Бүтіндей барлығы *n* айнымалы бар, сондықтан меншіктелмеген айнымалылардың орташа мәні $n/\sqrt{n}=\sqrt{n}$. Керекті нәтиже Чернов (I.7) теңсіздігінен $\mu=\sqrt{n}$ және $\delta=1/2$ алсақ шыға келеді.

31.1-лемманың тұжырымы бойынша, кездейсоқ дара бағалауды қолданғаннан кейін, схеманың өлшемі бұрынғыдай көпмүшелік санды енгізу болатын, үлкен ықтималдықпен жеткілікті түрде меншіктелмеген айнымалылар қалады. Сондықтан бұл лемма өте маңызды болып табылады.

Біздің негізгі дамытуымызда қолданыс табатын екі лемма төменде келтірілген.

31.2-лемма *c* тұрақты болсын және A өлшемі жоқ дегенде *c*-ға тең кез келген жиын болсын, бірақ ол өлшем $o\left(n^{1/c}\right)$ делік. Сонда жеткілікті үлкен *n* саны үшін кездейсоқ дербес бағалау *n*-нен көп айнымалыны меншіктемей қалдыру ықтималдығы $n^{1-c/2}$ санымен жоғарыдан шектеледі.

Дәлелдеу. A-ның кездейсоқ дербес бағалаудан кейін меншіктелмей қалған айнымалыларының санын сипаттайтын, кездейсоқ айнымалы шама X болсын. Біз Pr ($X\geq c$) ықтималдығын бағалағымыз келеді. Егер $s=|A|$ болса, онда меншіктелмей қалған айнымалылардың күтілетін саны $\mu=sn^{-1/2}$ болады. Осы санды жеткілікті үлкен *n* үшін алынған Чернов (I.6) леммасына қоямыз. Сонда

$$\Pr\left(X\geq c\right)<e^{c-\mu}\left(\mu/c\right)^{c}\ \leq\ \left(e\mu/c\right)^{c}$$

$$=\left(esn^{-1/2}/c\right)^{c}=n^{-c/2}\left(es/c\right)^{c}\ \leq\ n^{1-c/2}\ .$$

Мұндағы соңғы теңсіздік, *s* саны $o\left(n^{1/c}\right)$ болады деген ұйғарымды пайдаланып алынды.

31.3-лемма a мен *b* тұрақтылар болсын. *S* өзара қиылыспайтын айнымалылар жиыны болсын. *S-тің өлшемі жоқ дегенде* $b\log n$ болсын және *S-тің барлық элементінің өлшемі a*-дан аз болсын. $A \in S$ үшін *кездейсоқ дара бағалау* A -ның кез келген айнымалысына 0 меншіктейтін оқиға E (A) болсын. Сонда жеткілікті үлкен *n* үшін, кез келген $A \in S$ үшін *E (A)* оқиғасының орындалмау ықтималдығы жоғарыдан $n^{b\log\left(1-2^{-a}\right)}$

Дәлелдеу. $A \in S$ және $s = | A |$ болсын. Жеткілікті үлкен *n* үшін, *E(A)* оқиғасының орындалу ықтималдығы үшін $2^{-s}\left(1-n^{-1/2}\right)^{s} \geq 2^{-s} / 2 \geq 2^{-a}$ болады. Сондықтан *E(A)*-ның орындалмау ықтималдығы $1-2^{-a}$ санымен шектелген. Ал S-тің элементтері өзара қиылыспайтын болғандықтан, *E(A)* тәуелсіз оқиға. Сондықтан *E(A)* оқиғасы кез келген $A \in S$ үшін орындалмайды деген оқиғаның ықтималдығы әрбір орындалмаудың ықтималдықтарының көбейтіндісіне тең және ол $\left(1-2^{-a}\right)^{|S|}$ санымен шектеледі. Бірақ

$$\left(1-2^{-a}\right)^{|S|} \leq \left(1-2^{-a}\right)^{b\log n} = n^{b\log\left(1-2^{-a}\right)}.$$

Енді біз үш леммаға бөліп тасталынған ой түйіннің (аргумент) негізгі бөлігін көрсете аламыз. Сонымен *t* тұрақты болсын. Егер 1-ші деңгейдің әрбір қақпасының дәрежесі *t* -дан көп болмаса, онда схеманы *t*-схема деп айтамыз; басқаша айтсақ, егер 2-ші деңгейдің кез келген қақпасы *t-CNF* схема болса.

31.4-лемма. Егер *PARITY-дің тереңдігі d-ға тең өлшемі полиномиалды схемасы бар болса, онда белгілі бір тұрақты t* үшін *PARITY-дің тереңдігі d-ға тең өлшемі полиномиалды t*-схемасы бар болады.

31.5 -лемма. Егер *t ≥ 1* болып және *PARITY-дің* полиномиалды өлшем тереңдігі $d \geq 3$ болатын *t*-схемасы бар болса*, онда PARITY-дің тереңдігі d* болатын *(t-1)*-схемасы бар болады.

31.6-лемма. Егер *PARITY-дің полиномиалды өлшем тереңдігі* $d \geq 1$ болатын *1*-схемасы бар болса, онда *PARITY-дің полиномиалды өлшем тереңдігі* d - 1 болатын схемасы бар болады.

31.4-лемманың дәлелдемесі. PARITY-дің тереңдігі d және өлшемі n^k болатын схемасы бар деп жориық. *n*-ші схеманың айнымалысының кездейсоқ дара бағалауы *ρ*-ны қарастырайық. *t=2k+4* және *b=(k+1) /* log*(3/2)* болсын. 1-деңгейдің белгілі бір C клозы үшін $|\rho(C)| > t$ оқиғасын қарастырайық. Басқаша айтсақ *C-ның t*-дан көп литералдары меншіктелмей қалады деген оқиға. Жеткілікті үлкен *n* үшін бұл оқиға $n^{-(k+1)}$-ден артық ықтималдықпен орындалатынын көрсетеміз.

C -ның өлшеміне байланысты үш түрлі жағдай туындайды.

1-жағдай. Егер $|C| \le t$ болса, онда ықтималдық 0-ге тең, яғни дәлелдеме аяқталды.

2-жағдай. Егер $t \le |C| \ge b \log n$ болса, онда 31.2 лемма бойынша жеткілікті үлкен *n* үшін *C*-да кездейсоқ дара бағалау *t* литералдан көп болу оқиғасының ықтималдығы жоғарыдан $n^{1-t/2} = n^{-(k+1)}$ шектеледі.

3-жағдай. Соңғы жағдай $|C| \ge b \log n$. Егер С-ның кез келген литералына 1-ді меншіктесек, онда $\rho(C) = 1$ болады, одан $|\rho(C)| = 0$ болатыны шығады. Сонымен, жоқ дегенде *t* литералдың меншіктелмеу оқиғасының ықтималдығы *C*-да ешқандай литералға 1 меншіктелмейді деген оқиғаның ықтималдығынан аспайды. Жеткілікті үлкен *n* үшін белгілі бір тиянақты литералға 1-ді меншіктемеу оқиғасының ықтималдығы

$$1 - \frac{1}{2}\left(1 - \frac{1}{\sqrt{n}}\right) = \frac{1}{2}\left(1 + \frac{1}{\sqrt{n}}\right) \le \frac{2}{3}$$

болатыны, бізге қажетті ықтималдықты есептеуге қажет болатынына назар аударыңыз. Бұл оқиғалар тәуелсіз, сондықтан *C*-да ешқандай литералға 1 меншіктелмейді деген оқиғаның ықтималдығы

$$\left(\frac{2}{3}\right)^{|C|} \le \left(\frac{2}{3}\right)^{b \log n} \le n^{b \log(2/3)} = n^{-(k+1)}$$

санынан аспайды.

Жеткілікті үлкен *n* үшін клоз С-ның әрбір 1-деңгейі үшін, $|\rho(C)| > t$ оқиғасының ықтималдығы $n^{-(k+1)}$-дан аспайтындығын көрсеттік. Қосындылар заңы бойынша, литералдар саны t-дан артық 1-деңгейлі клоздың табылу ықтималдығы осы ықтималдықтардың қосындысымен шектеледі. Сонымен 1-деңгейлі клоздар саны n^k-дан аспау ықтималдығы

$$\Pr\left(\exists C \mid \rho(C)\mid > t\right) \le \sum_C \Pr\left(\mid \rho(C)\mid > t\right) \le n^k n^{-(k+1)} = n^{-1}$$

болады және ол шексіз аз сан.

Енді 31.1-леммасына сүйеніп, меншіктелмеген айнымалылар саны $\sqrt{n}/2$-дан кем болу ықтималдығы да шексіз аз сан болады. Тағы да қосындылар заңы бойынша, кездейсоқ бір дербес бағалауда бір оқиғаның болу ықтималдығы да шексіз аз сан болады. Яғни, кездейсоқ дербес бағалау арқылы жоқ дегенде $\sqrt{n}/2$ айнымалыны меншіктеусіз қалдырады және барлық 1-деңгейдегі қақпаларды t-дан асып кетпейтін дәрежеге, 1-ге шексіз ұмтылатын ықтималдықпен апарады. Бұл оқиғаның ықтималдығы нөлге тең болғандықтан, оны түсінетін дербес бағалау бар болуы қажет. Осыны дербес бағалау арқылы іске асырып (және қажет болып жатса бірнеше енгізулер тағайындап), *PARITY* үшін $\sqrt{n}/2$ айнымалылы дәрежесі t-дан аспайтын қақпасы 1-ші деңгейлі схема аламыз. n-ішіндегі полином әлі де $\sqrt{n}/2$ ішінде полином болып табылатындықтан, бұл схемалар әлі де полиномиалды өлшемді.

31.5-лемманың дәлелдемесі. PARITY-дің тереңдігі $d \ge 3$ және өлшемі n^k болатын t-схемасы бар делік, мұндағы $t \ge 1$. 2-ші деңгейдегі конъюкцияның белгілі бір клоздар жиыны S болсын. T арқылы S-тің максимал ішкі жиынын белгілейік. T жиынының кез келген екі клозы бірдей айнымалыны теріс немесе оң болса да қамтымайды. Таңба maximal арқылы T-ны қамтитын айтылған қасиетке ие болатын, сыртқы жиынның жоқ екендігін айтамыз. S-тің осындай жиыны T-ны S-тің барлық клоздарын белгілі ретпен қарастыра отырып құрастыруға болады. Айталық, S-тің келесі клозын T-ға қабылдау үшін ол клоздың осыған дейінгі T-ға қабылданған клоздардың ешқайсысымен ортақ айнымалысы болмауы қажет.

Енді $a=2k+4$ және $b=-(k+1)/log(1-2^{-a})$ болсын. T-ның айнымалыларына кездейсоқ дербес бағалаудың әсерін қарастыратын боламыз. Екі жағдайда қарастырамыз.

1-жағдай. Егер Т-ның элементтері $b\log n$-нен аз емес болса, онда біздегі клоздар саны $b\log n$-нен кем емес болады, оның ешқайсысы айнымалы бөліспейді. 31.3 лемма бойынша, жеткілікті үлкен n үшін Т-ның белгілі бір клозы $1-n^{b\log\left(1-2^{-a}\right)}=1-n^{-(k+1)}$ санынан кем емес ықтималдықпен барлық 0-ді қабылдайды, және бұл жағдайда 2-ші деңгейдегі барлық конъюкциялар жоғалып кетеді.

2-жағдай: Егер Т-ның элементтері $b\log n$-нен көп емес болса, онда $\bigcup T$-ның элементтер саны $bt\log n$-дан артпайды және S-тің барлық клоздарамен айнымалы бөліседі (олай болмаса Т maximal болмас еді). 31.2 лемма бойынша, жеткілікті үлкен n үшін кездейсоқ дербес бағалау $\bigcup T$-ның a-дан артық элементін меншіктемей қалдырып кету оқиғасының ықтималдығы $n^{1-a/2}=n^{-(k+1)}$ санымен жоғарыдан шектелген. Сонымен өте үлкен ықтималдықпен $\bigcup T$-ның $2a$-дан көп емес меншіктелмеген литералдары $\ell_1,\dots,\ell_{2a}$ табылады. Және бұрынғыдай өлшемі t болатын $\rho(S)$-тің әрбір клозы, осы литералдардың біреуін қамтыуы қажет. Өлшемі $(t-1)$-ден аспайтын барлық $\rho(S)$ клоздардың конъюнкциясы φ_0 болсын. $\rho(S)$-тің қалған клоздарының конъюнкциясы φ_j болсын, олар ℓ_j литералын қамтиды және егер $i<j$ болса, онда ℓ_i литералын қамтымайды. Және де φ'_j арқылы ℓ_j -дің φ_j -де жатуын жоятын конъюкция болсын. Сондықтан φ_j мен $\ell_j \vee \varphi'_j$ эквивалентті болады және алғашқы конъюнкция дербес бағалаудан кейін

$$\bigwedge_{j=0}^{2a} \varphi_j \;\equiv\; \varphi_0 \wedge \bigwedge_{j=1}^{2a}\left(\ell_j \vee \varphi'_j\right)$$

өрнегіне эквивалентті. Логикалық алгебраның үлестрімділік заңын пайдаланып, 2^{2a} -дан аспайтын $\varphi_0 \wedge \psi_1 \wedge \dots \wedge \psi_{2a}$ формадағы (t–1)-CNF схеманың дизъюнкциясы ретінде өрнектеуге болады, мұндағы әрбір ψ_i не ℓ_i не φ'_j.

Жеткілікті үлкен n үшін, $1-n^{-(k+1)}$ санынан кем емес ықтималдықпен кез келген 2-ші деңгейдегі t-CNF қақпа, тұрақты дизъюнкция санды (t–1)-CNF қақпаға, кездейсоқ дербес бағалау мағынасында эквивалентті болады екен. Мұнда (t–1)-схема алу үшін бұл дизъюнкция 3-деңгейдегі дизъюнкциямен бірігу мүмкін.

Дәлелдеменің қалған бөлігі 31.4 леммада көрсетілген жолмен жүреді. Қысқаша айтсақ, әрбір қарастырылған оқиғаның ықтималдығы 1-ге аса жылдам ұмтылады. Және жоқ дегенде $\sqrt{n}/2$ меншіктелмеген айнымалы қалатын жағдайдың да ықтималдығы 1-ге аса жылдам ұмтылады. Осының барлығы нөлге тең емес ықтималдықпен атқарылады.

31.6-лемма дәлелдемесі. Бұл жеңіл жұмыс. Полиномиалды өлшемді тереңдігі $d \geq 1$ болатын 1-схема, бір элементті 1-деңгейлі қақпаларды айналып өту арқылы, тереңдігі $d-1$ болатын схемаға эквивалентті. (Негізінде, бұл 31.5-лемманың $t=1$ болғандағы жеке жағдайы ғана)

31.7-лемма n енгізу үшін (*n–1)-CNF* немесе (*n–1)-DNF* жұптыққа тексеру схемасы табылмайды.

Дәлелдеме. *n* енгізуде бір клозда *n–1* литералдан аспайтын клоздардың конъюнкциясынан (*n–1)*-CNF схема құралады. Схеманы жұптыққа тексерудің кез келген дербес бағалауы ол басқа қалған айнымалыларды жұптыққа тексеру схемасы. Десек те, *n–1* айнымалыны іріктеп алып, белгілі бір клозда барлық литералды 0 жасай аламыз. Сонымен, басқа айнымалылардың мәндеріне тәулсіз түрде барлық схеманың мәні 0-ге тең болады, сондықтан ол жұптыққа тексеру схемасы бола алмайды. *n–1*-DNF схема үшін ой түйін (аргумент) дәл осылай жүргізіледі.

31.4 - 31.7 леммаларды біріктірсек, алынатын тұжырым

31.8-теорема (Ферст, Саксен, и Сипсер [46]) PARITY $\notin AC^0$.

Дәлелдеме. 31.4-31.6 леммалар конструкциясын қайталай отырып, тереңдігі тұрақты PARITY және полиномды өлшем үшін кез келген схемалар тобынан бастай аламыз. Және оларды 2-тереңдікті PARITY-дің кез келген схемалар тобына айналдыра аламыз. Ол – полиномиалды өлшемді және 1-ші деңгейдегі тұрақты жарты дәрежелі кіру схемасы. (Бағытталған графта жарты дәрежелі шығу (шығатын доғалар саны) және жарты дәрежелі кіру (кіретін қабырғалар саны) атты терминдер қалыптасқан). Белгілі бір тұрақты *t* үшін бұл схемалар *t*-DNF немесе *t*-CNF схемалар болып табылады. Бірақ 31.7-лемма бойынша бұл мүмкін емес.

Қосымша лекция Н

Алмастыру-қосу жайлы лемма

Бұл лекцияда Хастад [56] ауыстыра-қосуын дәлелдейміз, бұл ауыстыра-қосу 30.1-лекцияда баяндалғандай, PH^A мен $PSPACE^A$-ны ажыратуға қол жеткізуге қажетті, схема өлшемінің төменгі шекарасын алу үшін қолдануға болады.

Айнымалылардың кез келген X жиыны, еркін логикалық алгебра $\mathcal{F}_X$-ты жасайды, негізінде ол алгебра логикалық алгебра модуліндегі X-тің логикалық формулалар жиыны. Егер $|X| = n$ болса; онда $\mathcal{F}_X$-тің 2^{2^n} элементі болады. Пропозициялық қамту мағынасында, $\mathcal{F}_X$-те табиғи дербес реттілік $\leq$ бар; сонымен егер тек егер $\varphi \leq \psi$ болса, онда $\varphi \rightarrow \psi$ пропозициалы тавтология болады.

X-тыңі термы литералдар конъюнкциясы екенін еске саламыз, олардың екеуі ешқашан комплементарлы жұп құрастырмайды. М және N термдері үшін, егер N - де пайда болатын әрбір литерал М-де де пайда болса, онда $M \leq N$. $\mathcal{F}_X$-тің $\leq$ – максималды элементі бос терм 1 болып табылады. Егер М термі әрбір клоз φ-ден жоқ дегенде бір литерал қамти алса, онда M терм және CNF формула φ үшін $M \le \varphi$ деп жазамыз.

Егер $M \le \varphi$ болса, онда φ фомуласының минтермі M-нің $\le$ – максималы болады. Егер φ клозы CNF формула болса, онда φ-дың минтермы M болуы үшін, егер тек егер φ-дің әрбір клозынан терм M жоқ дегенде бір литерал қамтуы қажет және ешқандай ішкі терм M-а мұндай қасиетке ие болмауы қажет (71-аралас жаттығу). Логикалық алгебраның дистрибутивті заңдылығын пайдаланып, φ-дің барлық минтермдерінің дизъюнкциясы φ-ге эквивалентті DNF формула болатынын дәлелдеуге болады (72-ші әртүрлі жаттығу). φ-дің минтермдерінің жиынын $m(\varphi)$ арқылы белгіленеді.

Дербес баға $\rho : X \to X \cup \{0,1\}$ логикалық алгебраның $\mathcal{F}_X \to \mathcal{F}_Y$ гомоморфизімін туындатады, мұндағы $Y \subseteq X$ жиыны ρ-ның көмегімен меншіктелмеген айнымалылар. Егер $A \subseteq X$ болса, онда $\rho(A) = A$ деп жазамыз, егер A-ның ешқандай элементі ρ-мен меншіктелмесе, яғни $\rho(x) = x$ барлық $x \in A$ үшін. Кез келген терм M үшін екінің бірі орындалады, не $\rho(M) = 0$ немесе $M \le \rho(M)$.

ρ гомоморфизм болғандықтан, ол $\le$-ті сақтайды. Сондықтан, егер $M \le \varphi$ болса, онда $\rho(M) \le \rho(\varphi)$. $\rho(\varphi)$-дің кез келген N минтермі, φ-дің белгілі бір M минтермі үшін $\rho(M)$ болатынын дәлелдеуге болады (80-аралас жаттығу). Дегенмен, M минтермі φ-ден алынса, онда $\rho(\varphi)$-дың минтермі $\rho(M)$ болады десек дұрыс болмайды. Мысалы, егер $\psi = (x \vee z) \wedge (y \vee \bar{z})$ және $\rho(x) = x$, $\rho(y) = y$, және $\rho(z) = 0$ болса, онда ψ-дың минтермі xy болады, бірақ xy өрнегі $(\rho(\psi) = x)$-тің минтермі бола алмайды. Десек те, бізде келесі жеке жағдай бар.

H.1-лемма. Айталық, φ формула, C клоз және $M \in m(\varphi \wedge C)$ болсын. M және C екеуінде де кездесетін айнымалылар жиыны $var(M, C)$ болсын (полярлықты сақтау міндетті емес). $\sigma : var(M, C) \to \{0,1\}$ дегеніміз бір мәнді $\sigma(M) \ne 0$ мағынадағы баға болсын. Сонда $\sigma(\varphi \wedge C) = \sigma(\varphi)$ мен $\sigma(M)$ бағалары $\sigma(\varphi)$-дың минтермдері болады.

Дәлелдеме. M-дегі сәйкес литералдар 1 мәнін алуы үшін $var(M, C)$-дағы айнымалыларға мәнді меншіктеудің тек бір ғана жолы болғандықтан, баға σ бір мәнді болады. Терм M литералдарды

С-дан қабылдайтын болғандықтан, $\mathrm{var}(M, C)$ жиыны бос емес және $\sigma(C) = 1$, сондықтан $\sigma(\varphi \wedge C) = \sigma(\varphi) \wedge \sigma(C) = \sigma(\varphi)$.

Терм *M*-ді термдер конъюнкциясы $M = \sigma(M) M'$ ретінде жазуға болады, мұндағы M’ литералы $\mathrm{var}(M, C)$-тің барлық айнымалысын қамтиды. $\sigma(M)$-нің $\mathrm{var}(M, C)$-ның айнымалысын қамтымайтынына назар аударыңыз. Дәл осылай, φ-ді конъюнкция $\varphi_0 \wedge \varphi_1$ түрінде жазуға болады, мұндағы φ_1 әрбір клозы *M′* литералын қамтитын CNF формула және φ_0 ешқандай клозы *M′* литералын қамтымайтын CNF формула. Сонда $\sigma(\varphi) \le \varphi_0$ және $M' \le \varphi_1 \wedge C$.

$\sigma(\varphi)$-дің $\sigma(M)$ минтермі болатынын дәлелдейік. $M \le \varphi$ болғандықтан, $\sigma(M) \le \sigma(\varphi)$ болатыны түсінікті. Егер *N* басқа терм болып $\sigma(M) \le N \le \sigma(\varphi)$ қатынасы орындалса, онда

$$M = \sigma(M) M' \le N M' \le \sigma(\varphi) \wedge \varphi_1 \wedge C \le \varphi \wedge C.$$

Дегенмен M термі $\varphi \wedge C$-ның минтермі, сондықтан $M = \sigma(M) M' = N M'$. Ал $\sigma(M)$ мен *M′* айнымалылардың қиылыспайтын жиынында жататындықтан, $\sigma(M) = N$ болады.

Н.2-лемма φ формула және *W* айнымалылар жиыны болсын. Және

$$\varphi^{-W} \overset{\text{def}}{=} \bigwedge_{\tau : W \to \{0,1\}} \tau(\varphi)$$

болсын.

(i) Егер φ *CNF-те жазылған болса, онда* φ*-ден артық емес клоздар мағынасында* φ^{-W} *CNF формул*аға эквивалентті болады.

(ii) Егер $\mathrm{var}(M, W) = \varnothing$ болатындай $M \in m(\varphi)$ болса, онда, $M \in m(\varphi^{-W})$.

(iii) Егер $\rho(W) = W$ болса, онда $\rho(\varphi)^{-W} = \rho(\varphi^{-W})$.

(iv) Егер $\rho(W) = W$ болса, онда $\rho(\varphi) = 1$ онда тек онда $\rho(\varphi^{-W}) = 1$ ғана.

Дәлелдеме. (i) үшін, φ CNF -те жазылған болсын. Егер *W*-ның айнымалыларын қамтитын литералдардың барлығы жойылған болса, онда φ^{-W} φ-ға эквивалентті болатынын дәлелдейміз. Осыны С-ның әрбір клозына көрсетсек жеткілікті болады. Әрбір $\tau : W \to \{0,1\}$ үшін, $\tau(C)$ не 1-ге тең немесе литералы *W*-ның айнымалыларын қамтитын клоздың барлығы жойылғаны. Егер τ литералдардың біреуіне 1 берсе, онда алғашқы шарт орындалады. Егер τ барлық литералға 0 берсе, онда екінші шарт орындалады және жоқ дегенде осындай бір τ табылуы қажет. Сонымен конъюнкция C^{-W} осы клозға эквивалентті.

(ii) үшін, біз (i)-ді пайдаланамыз. φ CNF -те жазылған болсын. $\mathrm{var}(M, W) = \varnothing$ болғандықтан, M -нің барлық клоздармен ортақ литералы болады. W-ның ортақ айнымалылары жойылған болса да, бұл шарт орындалады, сонымен $M \leq \varphi^{-W}$. Бұдан басқа, әрбір клоз литералдар дизъюнкциясы болғандықтан, $C^{-W} \leq C$, сондықтан $\varphi^{-W} \leq \varphi$. $M \leq \varphi^{-W} \leq \varphi$ болады және M φ-дың минтермі екендігі белгілі, сонда ол φ^{-W} -дын да минтермі болады (78-әртүрлі жаттығу).

Біз (iii) пен (iv)-ті жаттығу ретінде қалдырамыз (79-әртүрлі жаттығу).

H.3-лемма (Алмастыра-қосу леммасы [56]) *X-тің* айнымалыларында φ *t-CNF* формула болсын. Әрбір айнымалыға тәуелсіз түрде 0 немесе 1-ді $(1-p)/2$*-ге* тең ықтималдықпен меншіктейтін немесе p ықтималдықпен мән меншіктемейтін, кездейсоқ дербес баға ρ болсын. Сонда

Pr ($\rho(\varphi)$ s-DNF формулаға эквивалентті емес) $\leq \alpha^s$, мұндағы $\alpha = 4pt/\ln 2 \sim 5.77pt$.

Дәлелдеу. Әрбір формула өзінің минтермдерінің дизъюнкциясына эквивалентті, сондықтан

$$\mathrm{pr}(\exists M \in m(\rho(\varphi)) \quad |M| \geq s) \quad \leq \quad \alpha^s$$

болатынын көрсетсек жеткілікті болады. Дәлелдемені φ-дың клоздарының санын пайдаланып индукциямен жүргіземіз. Негізінде біз біршама күшейтілген индукциялық гипотезаға мұқтажбыз, дәлірек айтсақ, ρ-ның әсері кез келген формула ψ -де 1-ге тең болып қалса да шек сақталады:

$$\Pr(\exists M \in m(\rho(\varphi)) \quad |M| \geq s \;\mid\; \rho(\psi) = 1) \leq \alpha^s. \quad (H.1)$$

Базистік $(\varphi = 1)-$де жалғыз минтерм 1 болып табылады, сондықтан теңсіздіктің сол жағы 0-ге тең, бірақ егер $s = 0$ болса, онда теңсіздіктің оң жағы 1-ге тең болады.

Индукция қадамы, жоқ дегенде бір бос емес клозда жататын $\varphi \wedge C$ формуланы қарастырамыз. Егер G_i өзара үйлесімсіз және толық оқиғалар үйірі болса, онда $\Pr(E \mid F) \leq a$ болатынын көрсету үшін барлық i үшін $\Pr\left(E \mid G_i \wedge F\right) \le a$ болатынын жеке көрсетсек жеткілікті болады (73-аралас жаттығу). Сонымен, (H.1)-ді дәлелдеу үшін

$$\Pr(\exists M \in m\left(\rho(\varphi)\right) \ |M| \ge s \mid \rho(C) = 1 \wedge \rho(\psi) = 1 \quad \le \quad \alpha^s \quad (H.2)$$

$$\Pr(\exists M \in m\left(\rho(\varphi)\right) \ |M| \ge s \mid \rho(C) \ne 1 \wedge \rho(\psi) = 1 \quad \le \quad \alpha^s \quad (H.3)$$

болатынын жеке көрсетсек жеткілікті болады.

Екеудің жеңілі (H.2) теңсіздігі. Бұл жерде бізге күшті индукциялы гипотеза қажет. $\rho(C)=1$ шарты орындалған жағдай үшін орындалады $\rho(\varphi \wedge C) = \rho(\varphi) \wedge \rho(C) = \rho(\varphi)$ және $\rho(C \wedge \psi) = 1$ егер тек егер $\rho(C) = 1$ және $\rho(\psi) = 1$, сонымен (H.2)

$$\Pr\left(\exists M \in m\left(\rho(\varphi)\right) \ |M| \ge s \mid \rho(C \wedge \psi) = 1\right) \quad \le \quad \alpha^s$$

теңсіздігіне эквивалентті. Бұл тікелей индукция гипотезасынан алынады.

Енді (H.3)-ке ауысайық. Дербес баға ρ үшін

$$M \in m\left(\rho(\varphi \wedge C)\right) \ \wedge \ |M| \ge s \ .$$

деп ұйғарайық. $A = \operatorname{var}\left(M, \rho(C)\right) \ne \varnothing$ болсын. H.1 лемма бойынша $\sigma : A \to \{0,1\}$ және $N \in m\left(\sigma\left(\rho(\varphi)\right)\right)$ табылып (дәлірек айтсақ $N = \sigma(M)$), төмендегі шарттар орындалады:

- $|N| \ = \ |M| - |A| \ \ge \ s - |A|$
- $\operatorname{var}(N, C) \ne \varnothing$
- $\sigma(C) = 1$.

Қосымша, $A \subseteq \operatorname{var}\left(\rho(C)\right)$ болғандықтан, $\rho(A) = A$ болады және σ мен ρ комутацияланады, сондықтан оларды айнымалылардың қиылыспайтын жиыны деп атайды.

Осы жолмен ρ-ны $\rho_1 \circ \rho_0$ композициясы ретінде өрнектеуге болады, мұндағы, ρ_0 $\operatorname{var}(C)$-ның дербес бағасы және ρ_1 $(X - \operatorname{var}(C))$-ның дербес бағасы. Сондықтан $\rho_1(C) = C$, $\rho_0\left(X - \operatorname{var}(C)\right) = X - \operatorname{var}(C)$, және ρ_1 мен ρ_0 комутацияланады.

Сонымен бізде бары

$$\mathsf{N} \in \mathsf{m}\left(\rho\left(\sigma\left(\varphi\right)\right)\right)$$

$$\Rightarrow \mathsf{N} \in \mathsf{m}\left(\rho\left(\sigma\left(\varphi\right)\right)\right) \text{ себебі } \sigma \text{ мен } \rho \text{ коммутацияланады}$$
$$\Rightarrow \mathsf{N} \in \mathsf{m}\left(\rho_1\left(\rho_0\left(\sigma\left(\varphi\right)\right)\right)\right)$$
$$\Rightarrow \mathsf{N} \in \mathsf{m}\left(\rho_1\left(\rho_0\left(\sigma\left(\varphi\right)\right)\right)^{-\mathsf{C}}\right) \text{ H.2 лемма (ii), } \mathsf{var}\left(\mathsf{N}, \mathsf{C}\right) = \varnothing$$
$$\Rightarrow \mathsf{N} \in \mathsf{m}\left(\rho_1\left(\rho_0\left(\sigma\left(\varphi\right)\right)^{-\mathsf{C}}\right)\right) \text{ H.2 лемма (iii), } \rho_1\left(C\right) = C\,.$$

Жоғарыда қабылданған ұйғарымдар көмегімен

$$\exists M \in m\left(\rho\left(\varphi \wedge C\right)\right) \quad |\mathrm{M}| \geq \mathrm{s} \ \Lambda \ \mathrm{var}\left(M, \rho\left(C\right)\right) = A, \quad \text{(H.4)}$$

болатынын көрсеттік, мұндағы $A \subseteq \mathrm{var}\left(C\right)$ және $\mathsf{A} \neq \varnothing$. Қорытып келесі ой түйінді айтуға болады

$$\exists \sigma : \mathsf{A} \to \{0,1\} \quad \exists \mathsf{N} \in \mathsf{m}\left(\rho_1\left(\rho_0\left(\sigma\left(\varphi\right)\right)^{-\mathsf{C}}\right)\right)$$

$$|N| \geq s - |A| \wedge \rho\left(A\right) = A \wedge \sigma\left(C\right) = 1\,. \quad \text{(H.5)}$$

Енді біз (H.3) теңсіздікті дәлелдеуге дайынбыз. Қосындылар заңын пайдаланып, (H.3)-тің сол жағын келесі түрде жаза аламыз

$$\sum_{\substack{\mathrm{A} \subseteq \mathrm{var}(C) \\ \mathrm{A} \neq \varnothing}} Pr\left(E\left(A\right) \mid \rho\left(C\right) \neq 1 \wedge \rho\left(\psi\right) = 1\right) \qquad \text{(H.6)}$$

мұнда $\mathsf{E}\left(\mathsf{A}\right)$ арқылы (H.4)-тің оқиғасы белгіленген. (H.4) -тен (H.5) алынады, ол мән мынадан артық емес

$$\sum_{\substack{\mathrm{A} \subseteq \mathrm{var}(C) \\ \mathrm{A} \neq \varnothing}} Pr\left(F\left(A\right) \mid \rho\left(C\right) \neq 1 \wedge \rho\left(\psi\right) = 1\right), \qquad \text{(H.7)}$$

мұндағы, $\mathsf{F}\left(\mathsf{A}\right)$ (H.5)-тің оқиғасы. Қосындылар заңы бойынша, ол мынадан артық емес

$$\sum_{\substack{A\subseteq \mathrm{var}(C)\\ A\neq\varnothing}} \sum_{\substack{\sigma:A\to\{0,1\}\\ \sigma(C)=1}} Pr\big(G(A)\mid \rho(C)\neq 1\wedge\rho(\psi)=1\big), \qquad (H.8)$$

мұндағы, $G(A)$ келесі түрдегі оқиға

$$G(A)\overset{\mathrm{def}}{=} \exists N\in m\Big(\rho_1\Big(\rho_0\big(\sigma(\varphi)\big)^{-C}\Big)\Big)\ |N|\geq s-|A| \ \wedge\ \rho(A)=A.$$

Дербес бағалау τ арқылы $\tau(C)\neq 1$ болатындай барлық $\mathrm{var}(C)$ үшін 73-аралас жаттығуға сай $\Pr\big(G(A)\mid \rho(C)\neq 1\wedge\rho(\psi)=1\big)$ байланыстың жеткілікті шегі

$$\Pr\big(G(A)\mid \rho(C)\neq 1\wedge\rho(\psi)=1\ \wedge\ \rho_0=\tau\big)$$
$$=\ \Pr\big(G(A)\mid \rho_0(C)\neq 1\wedge\rho_1(\rho_0(\psi))=1\ \wedge\ \rho_0=\tau\big) \qquad (H.9)$$

болады. Десек те, қабылданған жаңа $\rho_0=\tau$ шарттан соң $\rho_1(\rho_0(\psi))$ $\rho_1(\tau(\psi))$-ге өтеді, ал $\rho_1\Big(\rho_0(\sigma(\varphi))^{-C}\Big)$ $\rho_1\Big(\tau(\sigma(\varphi))^{-C}\Big)$-ге өтеді де, шарт $\rho_0(C)\neq 1$ артылып қалады. Осыған қосымша H.2 (iv) лемманы пайдаланып, $\rho_1(\tau(\psi))=1$ шартын $\rho_1\big(\tau(\psi)^{-C}\big)=1$ шартымен алмастыра аламыз. Барлық алмастырулардан кейін, (H.9)

$$\Pr\Big(\exists N\in m\Big(\rho_1\Big(\tau\big(\sigma(\varphi)\big)^{-C}\Big)\Big)\ |N|\geq s-|A|\ \wedge\ \rho(A)=A\Big.$$
$$\Big.\mid \rho_1\Big(\tau(\psi)^{-C}\Big)=1\ \wedge\ \rho_0=\tau\Big). \qquad (H.10)$$

түрге ие болады. Енді ρ_0 мен байланысқан шарттар, ρ_1 мен байланысқан шарттардан тәуелсіз, өйткені ол түрлендірулер айнымалылардың қиылыспайтын жиындарына ρ-ның әсерінен пайда болған-ды. Сонымен 75-аралас жаттығуды пайдаланып, (H.10) -ды

$$\Pr\Big(\exists N\in m\Big(\rho_1\Big(\tau\big(\sigma(\varphi)\big)^{-C}\Big)\Big)\ |N|\geq s-|A|\ \mid\ \rho_1\Big(\tau(\psi)^{-C}\Big)=1\Big)$$
$$\cdot\Pr\big(\rho_0(A)=A\mid \rho_0=\tau\big).$$

түрде көшіріп жаза аламыз. Индукцияның гипотезасы (ұйғарымы) арқылы бірінші фактор $\alpha^{s-|A|}$ -пен шектеледі, ал тура есептеу арқылы екінші фактор (2p / (1 + p)) | A |-мен шектеледі. Сонымен (H.8) мәні

$$
\begin{aligned}
&\sum_{\substack{A \subseteq var(C)\\ A \neq \varnothing}} \ \sum_{\substack{\sigma : A \to \{0,1\}\\ \sigma(C)=1}} \left(\frac{2p}{1+p}\right)^{|A|} \alpha^{s-|A|} \\
&= \sum_{k=1}^{|C|} \binom{|C|}{k} \left(2^k - 1\right) \left(\frac{2p}{1+p}\right)^k \alpha^{s-k} \\
&\le \sum_{k=0}^{t} \binom{t}{k} \left(2^k - 1\right) \left(\frac{2p}{1+p}\right)^k \alpha^{s-k} \\
&= \alpha^s \left(\left(1 + \frac{4p}{(1+p)\alpha}\right)^t - \left(1 + \frac{2p}{(1+p)\alpha}\right)^t \right) \\
&\le \alpha^s \left(\left(1 + \frac{4p}{\alpha}\right) - 1 \right) \qquad \text{(H.11)}
\end{aligned}
$$

аспайды. α -ның орнына $4pt / \ln 2$ қойып және барлық оң x үшін белгілі $(1 + 1 / x)^x \le e$ теңсіздігін пайдаланамыз. Сонда (H.11)-дегі жақшаның ішіндегі өрнектің ең үлкен мәні 1-ден аспайды, осының әсерінен (H.11)-дің мәні α^s -тен үлкен болмайды.

H.4-теорема Кез келген тұрақтылар c мен *d* аламыз сонда *PARITY* үшін тереңдігі d және өлшемі $2^{(\log n)^c}$ болатын схема табылмайды.

Дәлелдеу. Осындай схемалар бар деп ұйғарайық. $t = (\log n)^{c+1}$ және $\rho = (\ln 2) / (8\, t)$ болсын. Жалпылықты сақтай отырып, 2-деңгейдегі барлық схемалар *t*-CNF болады деп санайық (керек болып жатса бір деңгей қоса салса болады). Кездейсоқ дербес бағаларды қолдансақ, 2-деңгейдегі кез келген схема, α^t мен шектелген *t*-DNF схемаға эквивалентті жасалмауы ықтимал. Қосындылар заңы бойынша, берілген схемаға эквивалентті болмайтын 2-ретті схема табылуының ықтималдығы

$$2^{(\log n)^c} \cdot \alpha^t = 2^{(\log n)^c} \cdot 2^{-(\log n)^{c+1}}$$

шамасынан аспайды. Функция ретінде қарасақ, оның мәні шексіз аз шама.

Осының әсерінен, жеткілікті үлкен n үшін меншіктелмеген айнымалының күтілетін саны әлі де жеткілікті үлкен болады:

$$np = \frac{n\ln 2}{8(\log n)^{c+1}} \geq 2n(\log n)^{-(c+2)}.$$

Меншіктелмеген айнымалының нақты саны барлық санның жартысы болу ықтималдығы да шексіз аз шама (86-әртүрлі жаттығулар).

Сонымен, жеткілікті үлкен n үшін 2-деңгейдегі t-CNF схеманың барлығын оған эквивалентті t – DNF схемалармен алмастыру ықтималдығы нөлге тең және жоқ дегенде $n(\log n)^{-(c+2)}$ айнымалы меншіктелмей қалады. Енгізудің өлшемі $m = n(\log n)^{-(c+2)}$ болғанда жаңа схеманың өлшемі бұрынғыдай $2^{(\log m)^{c+3}}$ болады және нөлден өзгеше ықтималдық бар болғандықтан, оны ақиқат жасайтын дара баға бар болуы қажет.

Белгілі бір k санын үшін осы конструкцияны d – 2 рет қайталай отырып, тереңдігі 2 және өлшемі $2^{(\log m)^k}$ PARITY үшін барлық 1-деңгейлі схемаларының дәрежесі t-дан аспайтын схемалар үйірін аламыз. Жеткілікті үлкен n үшін бұл 31.7 леммаға қайшы келеді.

Н.5-салдар. $\mathrm{PH}^{A} \neq \mathrm{PSPACE}^{A}$ орындалатын, A оракулы табылады.

Дәлелдемесі. Н.4 және 30.1-теоремалар.

Қосымша I

Құйрықты шектеу

Ықтималды анализде, кездейсоқ шаманың өзінің орта мәнінен ауытқуының ықтималдығын шектеуге жиі мәжбүр боламыз. Осы мақсат үшін әртүрлі формула құрастырылған. Олар құйрықты шектеу деп аталады. Олардың ішіндегі ең әлсізі Марков шектеуі, ол шектеуде: орта мәні $\mu = \mathcal{E}X$ болатын кез келген X кездейсоқ оң шама үшін

$$\Pr(X \geq k) \quad \leq \quad \mu / k \tag{I.1}$$

болады деп айтылады (83-аралас жаттығу). Чебышев шектеуі оған қарағанда күштілеу болып табылады, ол шектеуде: кез келген $\delta \geq 1$

үшін, орта мәні $\mu = \mathcal{E}X$ және стандартты ауытқу $\sigma = \sqrt{\mathcal{E}\left((X-\mu)^2\right)}$ болатын *X* кездейсоқ шама үшін

$$\Pr\left(|X-\mu| \geq \delta\sigma\right) \leq \delta^{-2} \tag{I.2}$$

болады деп айтылады (84-аралас жаттығу).

Марков және Чебышев шектеулері сәйкес сызықты және квадратты жинақталады және керекті бағаны алу үшін бұл жиі әлсіз болып шығады. Әсіресе, Бернулли тәжірибесінің жеке жағдайы (0 мен 1-ді ғана қабылдайтын, бірдей үлестірілген, тәуелсіз кездейсоқ айнымалылардың қосындысы) немесе Пуассонның біршама жалпыланған тәжірибесі үшін (0 мен 1-ді ғана қабылдайтын, бірдей үлестірілуі шарт емес, тәуелсіз кездейсоқ айнымалылардың қосындысы), жинақтылық экспоненциалды болады.

Қосындысы $X = \sum_i X_i$ және $\Pr(X_i = 1) = p_i$ болатын, Пуассон тәжірибесін X_i, $1 \leq i \leq n$ қарастырайық. Құйрықты шектеудің жоғарғы дәл мәні

$$\Pr(X \geq k) = \sum_{\substack{A\subseteq\{1,\ldots,n\}\\ |A|\geq k}} \prod_{i\in A} p_i \prod_{i\notin A}(1-p_i).$$

Орындалу ықтималдығы p -ға тең Бернулли тәжірибесінің жеке жағдайы үшін, жоғарыдағы теңдік биномалды үлестірімге дейін

$$\Pr(X \geq k) = \sum_{i\geq k} \binom{n}{i} p^i (1-p)^{n-i}$$

ықшамдалады. Десек те, бұл өрнектер алгебралық тұрғыда күрделі болып саналады. Ықшам формулаға Чернов шектеуі арқылы қол жеткізуге болады.

Чернов шектеуі әртүрлі формада кездеседі. Оның бір формасы, кез келген $\delta > 0$ үшін, қосындысы $X = \sum_i X_i$ және $\mu = \mathcal{E}X$ болатын Пуассон тәжірибесі үшін

$$\Pr\left(X \geq (1+\delta)\mu\right) < \left(\frac{e^{\delta}}{(1+\delta)^{1+\delta}}\right)^{\mu}. \tag{I.3}$$

болады. Оған тең дәрежеде

$$\Pr\left(X \geq (1+\delta)\mu\right) < \left(e\left(1-\frac{\delta}{1+\delta}\right)^{(1+\delta)/\delta}\right)^{\delta\mu}. \tag{I.4}$$

деп жазса болады. (I.4) өрнегінің бөлігі

$$\left(1-\frac{\delta}{1+\delta}\right)^{(1+\delta)/\delta} \tag{I.5}$$

асимтотикалық анализде жиі кездесетін функцияның $(1-1/x)^{x}$ жеке жағдайы болып табылады. Бұл функция жоғарыдан e^{-1} мен шектелгенін, және x айнымалысы шексіздікке ұмтылғанда функция осы мәнге (e^{-1}) ұмтылатынын есте ұстаған жөн. Екінші жағынан, айнымалы x оң мәндерді қабылдаған кезде функция $(1+1/x)^{x}$ жоғарыдан *e* санымен шектеледі, x шексіздікке ұмтылғанда функция осы санға ұмтылады (57 (а) аралас жаттығулар).

(I.3) және (I.4)-ке эквивалентті үшінші форма: барлық $k > \mu$ үшін,

$$\Pr\left(X \geq k\right) < e^{k-\mu}\left(\mu / k\right) \tag{I.6}$$

қатынасы орындалады. Орта мәннен қашықтық артқан сайын жинақтылық экспоненциалды болатынын (I.4) және (I.6)-дан айқын көруге болады.

Бұл формулалар құйрықты үлестірімнің үстіңгі мәнін шектейді. Құйрықты үлестірімнің астыңғы мәні үшін симметриялы нұсқа (версия) бар: $0 \leq \delta < 1$ шарты орындалатын кез келген δ және $0 < k \leq \mu$ орындалатын *k* үшін

$$\Pr\left(X \le (1-\delta)\mu\right) < \left(\frac{e^{-\delta}}{(1-\delta)^{1-\delta}}\right)^{\mu} \tag{I.7}$$

$$= \left(e^{-1}\left(1+\frac{\delta}{1-\delta}\right)^{(1-\delta)/\delta}\right)^{\delta\mu} \tag{I.8}$$

$$\Pr\left(X \le k\right) < e^{k-\mu(\mu/k)^{k}} \tag{I.9}$$

қатынастары орындалады. Құйрықты үлестірімнің төменгі мәні үшін

$$\Pr\left(X \le (1-\delta)\mu\right) < e^{-\delta^{2}\mu/2} \tag{I.10}$$

теңсіздігі арқылы төртінші нұсқаға келеміз. Бұл шектеу (I.7)-(I.9) қатынастарына қарағанда әлсіздеу, соған қарамастан формасының қарапайым болуына байланысты қолданыста өте пайдалы.

Чернов теңсіздігінің дәлелдемесі

Енді X_i Пуассон тәжірибесі үшін Чернов теңсіздігі (I.3)-ті дәлелдейміз. Басқа формалар (I.4) пен (I.6)-ның эквивалентті болатынын өте жеңіл дәлелдеуге болады және олар оқулықта жаттығу ретінде берілген (87-аралас жаттығу). Құйрықты үлестірімнің төменгі шегі үшін сәйкес (I.7)-(I.9) шектеулердің дәлелдемелері ұқсас және олар да жаттығу ретінде қалдырылған (88-аралас жаттығу). Әлсіз шектеу (I.10)-ның дәлелдемесі $\ln(1-\delta)$ функциясын Тейлор қатарына жіктеуді қажет етеді, бірақ ол күрделі емес (89-аралас жаттығу). X_i-дің орындалу ықтималдығы әртүрлі болғанымен, ол тәжірибелердің тәуелсіз болуы өте маңызды. Дәлелдеменің шешуші қадамында тәуелсіз тәжірибелердің көбейтіндісінің күтілетін мәні, күтілетін мәндердің көбейтіндісіне тең болады деген тұжырымды пайдаланамыз (82-аралас жаттығу).

X_i Пуассон тәжірибесінің орындалу ықтималдығы p_i, қосындысы $X = \sum_i X_i$, ал орта мәні $\mu = \mathcal{E}X = \sum_i p_i$ болсын. $a > 0$ деп тиянақтайық. Функцияның монотондығы және (I.1) Марков теңсіздігін пайдаланып

$$\Pr\left(X \geq (1+\delta)\mu\right) = \Pr\left(e^{aX} \geq e^{a(1+\delta)\mu}\right)$$

$$\leq \mathcal{E}\left(e^{aX}\right)\cdot e^{-a(1+\delta)\mu} \tag{I.11}$$

теңсіздігін аламыз. Тәуелсіз тәжірибелердің көбейтіндісінің күтілетін мәні, күтілетін мәндердің көбейтіндісіне тең болатындығы (82-аралас жаттығу) және егер X_i тәуелсіз болса, онда e^{aX_i} -де тәуелсіз. Осыларды пайдаланып

$$\mathcal{E}\left(e^{aX}\right) = \mathcal{E}\left(e^{\sum_i aX_i}\right) = \mathcal{E}\left(\prod_i e^{aX_i}\right) = \prod_i \mathcal{E}\left(e^{aX_i}\right)$$

$$= \prod_i \left(p_i e^{a} + (1-p_i)\right) = \prod_i \left(1 + p_i\left(e^{a}-1\right)\right) \tag{I.12}$$

деп жаза аламыз. $\left(1+1/x\right)^{x} < e$ теңсіздігінен барлық оң y үшін $1+y < e^{y}$ теңсіздігі алынады. Осыны $y = p_i\left(e^{a}-1\right)$ деп алып қолдансақ, $1+p_i\left(e^{a}-1\right) < e^{p_i\left(e^{a}-1\right)}$ теңсіздігі алынады. Сондықтан (I.12) қатаң түрде

$$\prod_i e^{p_i\left(e^{a}-1\right)} = e^{\sum_i p_i\left(e^{a}-1\right)} = e^{\left(e^{a}-1\right)\mu}$$

өрнегімен шектеледі. Осыны $e^{-a(1+\delta)\mu}$ өрнегімен байланыстырсақ, (I.11) үшін қатаң шектеу

$$e^{\left(e^{a}-1\right)\mu}\cdot e^{-a(1+\delta)\mu} = e^{\left(e^{a}-1-a-a\delta\right)\mu} \tag{I.13}$$

аламыз. Біз енді $e^{a}-1-a-a\delta$ өрнегін минимумға жеткізетін a-ны таңдап алғымыз келеді. Егер $a = \ln\left(1+\delta\right)$ болса, онда туынды нөлге айналады, осы мәнді (I.13) -тегі a -ның орнына қойып ықшамдасақ (I.3) алынады.

32-лекция

Айырмашылық жайлы теорема және басқа патология

Біз қарастырған күрделілік иерархиясының структурасы, барлық проблемалар табиғи ерекше күрделілікке ие деген көзқарасқа апаруы мүмкін, және көптеген уақытты немесе кеңістікті талап етіп, әрқашан көп есептеуді қажет етеді. Көптеген күнделікті кездесетін проблемалар және күрделіктер шектеуі үшін осы екі тұжырым да орындалатын сияқты көрінеді, бірақ негізінде ол дұрыс емес. Патологиялық (кері) мысалдар құрастырып, осы мысалдар үшін олардың орындалмайтынын көрсетуге болады.

Мысалы, асимптотикалық емес ең ұтымды алгоритммен есептелетін f функциясын көрсетуге болады. Оны мына мағынада түсінеміз: f -тің $T(n)$ уақытта орындалатын кез келген алгоритміне сай, осы f үшін logT (n) уақытта орындалатын алгоритм табылады. Сонымен f-ті есептеу шексіз үдетіледі екен. log функцияның ешқандай ерекшелігі жоқ екендігін айта кетейік, нәтиже кез келген жалпы рекурсивті функция үшін де орындалады.

Басқа мысал, $S(n)$ кеңістігінде есептелетін кез келген функцияны, log$S(n)$-де де есептеуге болатын, кеңістік S(n) табылатынын көрсетуге болады. Алғашында, бұл 3.1-теоремаға қайшы сияқты көрінеді, бірақ ол теоремада конструкциялық күй бар, ал *S(n)*-де ол күй жоқ. Тағы да, бұл қасиет log-тан басқа да кез келген рекурсиялық жақсартылым үшін орындала береді.

Осы лекцияның көптеген мысалдары, аспандатылған ұғым диогонализация арқылы құрастырылады. Олар өмірде ештеңеге сай емес және нақты ешқандай қолданысты көрсетпейді. Бірақ оларды пайдаланып күрделіліктер теориясының күші мен шектеулерін жақсы түсінуге болады. Бұл лекцияда Тьюринг машинасының уақыты мен кеңістігін пайдаланып, айтылған нәтижелерді дәлелдейміз; және олардың көпшілігі нақты өлшемінен тәуелсіз екенін айта кетейік. Бұдан да абстракциялы ұғым J аралас лекцияда беріледі.

Бірінші мысалда айырмашылық жайлы теореманы қарастырамыз, ол теоремада күрделіліктер иерархиясында өте үлкен рекурсивті жетіспеушіліктер бар екені айтылады. Бұл нәтижені тәуелсіз түрде Бородин [21] және Трахтенброт [122] алған деп мойындалады.

32.1-теорема (Айырмашылық жайлы теорема [21, 122]) $f\left(x\right) \geq x$ орындалатын кез келген жалпы рекурциялы функция $f:\omega \to \omega$ үшін, $DTIME(f(T(n))) = DTIME(T(n))$ теңдігі орындалатын $T(n)$ уақыты бойынша шектеу табылады; басқаша айтсақ, $T(n)$ уақытта қабылдау-ашық болмайтын детерминирлі M_i ТМ үшін, $f(T(n))$ уақытта қабылдау-ашық детерминирлі ТМ болып кететін жиын табылмайды.

Дәлелдеу. Енгізу x-те *ТМ* машина M_i-дің жұмыс уақыты $T_i(x)$ болсын. Әрбір n үшін, егер $T_i(n) \leq f(m)$ болса, онда $T_i(n) \leq m$ болатын $T(n)$ үшін ең кіші m-ді тауып аламыз. $T(n)$-ды есептеу үшін $m{:=}0$ тағайындауынан бастаймыз. Теңсіздіктер $m < T_i(n) < f(m)$ орындалатындай айнымалы $i \leq n$ табылса, онда $m := T_i(n)$ деп аламыз. Мұндағы $i \leq n$ шарты ақиқат болатын жағдайдың саны шектеулі болатындықтан, бұл үдеріс ұзаққа созылмауы қажет. Соңғы $T(n)$-нің мәні m-ді береді.

Енді $T(n)$ теорема шартын қанағаттандырады деп ұйғарамыз. M_i машинасы $f(T(n))$ уақыт жұмыс жасайды деп ұйғарайық. Сондықтан б.ж.д.[1] $T_i(n) \leq f(T(n))$. T-ны құрастыру шарты бойынша, өте үлкен $n \geq i$ үшін, $T_i(n) \leq T(n)$ қатынасы орындалады.

Біздің дәлелдеген дүниеміз, теорема формулировкасынан күшті болып шықты. Теорема бойынша, $f(T(n))$ уақытта жұмыс істейтін кез

[1] *«б.ж.д.» − «барлық жағдайға дерлік» немесе « n -нің кейбір мәнінен басқа барлық мәндер үшін» деген сөз . Одан басқа, «ш.ж.» десек, ол − «шексіз жиі» = «шексіз көп n үшін».*

келген детерминирлі M_i ТМ үшін оған эквивалентті $T(n)$ уақытта жұмыс істейтін M_j ТМ табылады дейді. Іс жүзінде, $f(T(n))$ уақытта жұмыс жасайтын кез келген детерминирлі ТМ-сы, $T(n)$ уақытта да жұмыс жасайтынын көрсеттік.

Бұл бағалар б.ж.д. орындалады, бірақ біз оларды барлық жағдайда орындалатындай жасай аламыз. Оны шектеулі басқарудағы аздаған енгізулер үшін мәндерді кодтау және оларды таблица арқылы есептеу көмегімен іске асыруға болады.

Алдын ала таңдап алынған кез келген рекурсивті қосынды үшін кез келген алгоритмді бірнеше рет үдетуге болатынын келесі мысалдағы жиын көрсетеді. Бұл Блюм [17] нәтижесі болып табылады.

32.2-теорема (Үдету теоремасы [17]) Енгізу x-ке сай M_i *ТМ-ның жұмыс уақыты* $\mathrm{T}_i(x)$ болсын. Шарт $f(n)\geq n^2$ орындалатын, кез келген жалпы рекурциялы функция $f:\omega\to\omega$ болсын. Сонда рекурсивті A жиыны табылып, A-ға қабылдау ашық кез келген M_i *ТМ* үшін б.ж.д. $f(T_j(x))\leq T_i(x)$ орындалатын, A жиынына қабылдау-ашық екінші бір M_j ТМ табылады.

Дәлелдеу. f-тын өз өзіне n-рет жасалған композициясы f^n болсын:

$$f^n \stackrel{def}{=} \underbrace{f\circ f\circ\cdots\circ f}_{n}\cdot$$

Сонымен f^0 теңбе-тең функция, $f^1=f$, және $f^{m+n}=f^m\circ f^n$ Мысалы, егер $f(m)=m^2$ болса, онда $f^n(m)=m^{2^n}$, және егер $f(m)=2^m$ болса, онда $f^n(m)$-биіктігі n-ге тең екілік стекті толық қамтитын итерацияланған экспоненциал болады.

Диогоналдауды пайдаланып келесі шарттарға бағынатын $A\subseteq 0^*$ жиынды құрастырамыз:

(i) A-ға қабылдау ашық кез келген машина M_i үшін, б.ж.д.[2] $T_i(0^n)>f^{n-i}(2)$, және

(ii) барлық k үшін, A-ға қабылдау ашық M_j машинасы табылып, $T_j(0^n)\leq f^{n-k}(2)$ орындалады. Осымен біздің мақсат орындалды, өйткені A-ны қабылдайтын кез келген M_i машинасы, б.ж.д.

---

[2] *Біз* $f^{n-i}(2)$ *-ні тиянақталған i тұрақтылардан тұратын n-нен тәуелді функция ретінде қарастырамыз. Сонымен бұл контексте «ш.ж.» және «б.ж.д.» қысқартулары сәйкес «шексіз көп n үшін» және «шектеулі n -нен басқа барлық сандар үшін» дегенді білдіреді.*

$T_j(0^n) \le f^{n-i-1}(2)$ орындалатын A-ға қабылдау ашық M_j машинасының бар болуына кепілдік береді; ал екінші жағынан

$$\begin{aligned} f(T_j(0^n)) &\le f(f^{n-i-1}(2)) && \text{б.ж.д. } f\text{-тің монотондығынан} \\ &= f^{n-i}(2) \\ &< T_i(0^n) && \text{(i)-ден.} \end{aligned}$$

Енді A жиынын құрастырумен айналысамыз. Енгізу алфавиті $\{0\}$ болатын барлық бір таспалы Тьюринг машинасының тізімі $M_0, M_1, \ldots$ болсын. Төмендегі симуляцияны жасайтын аудару машинасы N болсын. Ол машина осы уақытта симуляциялайтын машиналарды сипаттайтын шектеулі жанды тізімді сүйемелдейді. Универсал симуляцияға сәйкес келетін M_i-дің сипаттамасы, i индексі арқылы жеңіл табылады делік.

N-ды есептеу этаппен жүргізіледі. Алғашында жанды тізім бос болады. Стадия n де, N машина келесі M_n машинасын жанды тізімнің соңына орналастырады. Осыдан кейін ол, жанды тізімдегі машиналарды ең төменгі индекстен бастап ретімен симуляция жасайды. Әрбір M_i үшін енгізу 0^n-ді пайдаланып $f^{n-i}(2)$ қадамда, ол M_i-ді симуляция жасайды. Ол алдымен өзіне бөлінген уақытта тоқтаған бірінші машинаға тоқтайды және кері қимыл жасайды: егер M_i машинасында 0^n-ге қабылдау-жабық болса, онда N машинасы $0^n \in A$ деп жариялайды, ал егер M_i машинасында 0^n-ге қабылдау-ашық болса, онда N машинасы $0^n \notin A$ деп жариялайды. Бұл $L(M_i) \ne A$ шартын қамтамасыз етеді. Осыдан кейін ол M_i-ді жанды тізімнен алып тастайды. Егер ешқандай машина, өзіне берілген уақытта жанды тізімде бөгелмейтін болса, онда N машина $0^n \notin L(M_i)$ деп жариялайды.

Бұл конструкция, ш.ж. $f^{n-i}(2)$ уақытта жұмыс жасайтын кез келген M_i машинада A-ға қабылдау-жабық болуын қамтамасыз етеді. Стадия i- машина M_i жанды тізімге орналастырылады. Осыдан кейін, егер 0^n үшін $f^{n-i}(2)$ уақыт ішінде M_i тоқтайтын болса, бірақ жою үшін таңдалмаған болса, онда жанды тізімнен жоғары проритетті басқа машина таңдап алынғаны; бірақ бұл жағдай тек шектеулі рет орындалуы мүмкін.

Сонымен, егер ш.ж. 0^n үшін M_i машинасы $f^{n-i}(2)$ уақыт аралығында тоқтаса, онда соңында M_i жанды тізімдегі ең үлкен приоритетті машина болып шығады және ол осы n-этапта жанды тізімнен шығарылуға таңдап алынады. Осы сәтте, $L(M_i) \ne A$ шартына кепілдік бере отырып, егер тек егер $0^n \notin L(M_i)$ болса, онда 0^n-ды A-да орналастыруға болады. Бұл жоғарыда айтылған (i) шартын береді.

(ii) шартты іске асырғымыз келсе, барлық k-үшін, б.ж.д. $f^{n-k}(2)$ уақытта бір таспалы N_k *ТМ*-сында A жиынына қабылдау-ашық болатынын көрсетуіміз қажет. Белгілі бір жеткілікті m үшін, соңғы бақылау N_k-да, N-ді есептеудің алғашқы m этапын қатаң кодтау идеяның өзегі болып табылады. Назар аударайық; әрбір M_i үшін келесі екі шарттың біреуі орындалады:

(A) ш.ж. $T_i(0^n) \le f^{n-i}(2)$ болсын, осы жағдайда $m(i)$ стадия табылып N машина M_i-ді жанды тізімнен шығарады; немесе

(B) б.ж.д. $T_i(0^n) > f^{n-i}(2)$ болсын, бұл жағдайда $m(i)$ стадия табылып, M_i өзіне бөлінген уақыттан артық жұмсайды.

Енді $m = \max_{i \le k} m(i)$ болсын. Біз $m(i)$ немесе m-ді тиімді түрде анықтай алмаймыз (105-аралас жаттығу), бірақ олардың бар екендігін білеміз. N_k машинасының $n \le m$ үшін, өзінің соңғы бақылауында қатаң кодталған элементтер тізімі $0^n \in A$ бар. Осындай енгізулер үшін, $0^n \in A$ болатынын тексеру және қабылдау-ашық немесе қабылдау-жабық болатынын анықтау үшін N_k машина кесте бойынша іздеу жасайды. Және $n > m$ үшін 0^n енгізулерінде, белгілі бір жанды тізімнен бастап, ол N-нің қимылдарын $m+1,\ m+2,\ \ldots,\ n$ этаптар үшін симуляция жасайды, олар да соңғы бақылауда қатаң кодталғанын айта кетейік. Ол бастау алатын жанды тізім, N-нің де m стадиядағы жанды тізімі болып табылады, мұнда $i \le k$ болғандағы барлық машиналар M_i жойылған. Бұл $0^n \in A$ статусын өзгерте алмайды: $i \le k$ болғандағы кез келген M_i үшін, (A) жағдай үшін m-ші стадияда M_i жанды тізімнен жойылып кеткен болатын. Ал (B) жағдайда m стадияда M_i машинасы өзіне тиесілі уақыттан көп уақыт жұмсайды, сондықтан ол еш уақытта жойылуға кандидат бола алмайды. Сондықтан симуляция N машинасы m стадиясында және оның сыртында тұрғандай жүргізіле береді. Осының әсерінен, $0^n \in A$ орындала ма және оған сәйкес қабылдау-ашық немесе қабылдау-жабық бола ма соны N_k машинасы анықтайтын болады.

Енгізу 0^n-дегі N_k-ның жұмыс уақытын бағалау ғана қалды. Егер $n \le m$ болса, онда N_k сызықты уақытты қажет етеді, ол уақыт енгізуді оқу және кесте бойымен іздестіру жасауға жеткілікті. Егер $n > m$ болса, онда $n - m$ енгізуде N_k машинасы, $(n-k)$-дан аспайтын машинаны симуляция жасауы қажет және әрқайсысына $f^{n-k-1}(2)$ қадамнан артық жасалмауы қажет. Кодтау схемасы жұмсақ шартпен жұмыс жасаған кезде, яғни индекс i-дің бинарлық түрде жазылуын M_i -дің сипаттамасы ретінде қарастырамыз, сонда M_i -дің $\log i$ -ден аспайтын күйі болады, $\log i$ -ден аспайтын таспалы символы болады, және соңғы бақылауда $\log i$ -ден аспайтын тармақ болады және M_i -дің бір қадамы N_k -ның $c(\log i)^2$ қадамы арқылы симуляция жасалуы мүмкін. Сонымен барлық симуляцияға қажетті жалпы уақыт $cn^2(\log n)^2 f^{n-k-1}(2)$ -дан артпайды. Бірақ

$$\begin{aligned} cn^2(\log n)^2 &\leq 2^{2^{n-k-1}} \\ &\leq f^{n-k-1}(2) \quad \text{өйткені} \quad f(m) \geq m^2, \end{aligned}$$

сондықтан

$$\begin{aligned} cn^2(\log n)^2 f^{n-k-1}(2) &\leq (f^{n-k-1}(2))^2 \text{ б.ж.д.} \\ &\leq f(f^{n-k-1}(2)) \\ &= f^{n-k}(2). \end{aligned}$$

Біз 32.2-теореманың дәлелдемесінен бірнеше қызықты тұжырымдар жасай аламыз.

Біріншіден, кодтау схемасындағы «жұмсақ ұйғарымдар» шартының әсері болымсыз. Егер олар орындалмаса $f(m) \geq m^2$ шартын сәкес күшейте түсу қажет. Универсалды түрде Тьюринг машинасын симуляциялау жоғарыдан жалпы рекурсивті функциямен шектелетінін білуіміз қажет.

32.2-теоремасының дәлелдемесіндегі мән $m = \max_{i \leq k} m(i)$ тиімді алынуы мүмкін емес. Әрбір M_i -ге сондай m сәйкес келетінін білеміз, бірақ M_i -дің (A) жағдай немесе (B) жағдайға тап болатыны шешілмеген, сондықтан M_i -ді жанды тізімнен жою қажет пе соны білмейміз. Нақты білетініміз; k-ға -тәуелді $f^{n-k}(2)$ уақытта тиімді жұмыс жасайтын A машинасын алу мүмкін емес (105-аралас жаттығу).

33-лекция

Дербес рекурсивті функция және Гёдел нөмірлеуі

Дербес және жалпы рекурсивті функциялар

Келесі бірнеше лекция классикалық рекурсивті функция теориясына кіріспе болып табылады. Бұл тақырыпты тіпті тереңдете оқығыңыз келсе [104, 114] жұмыстарды қараңыз.

Дербес рекурсивті функция деп есептеуге келетін дербес функцияны $f:\omega \to \omega$ айтады. Ол функция барлық енгізулерде анықтала бермейтіндіктен, оны дербес деп атайды. Есептелушілік өзара эквивалентті бірнеше тәсілмен анықталуы мүмкін: Тьюринг машинасымен, Гёделдің μ-рекурсивті функциясымен, λ- есептеумен немесе C тілінде программалау арқылы. Дербес рекурсивті функция барлық жерде анықталса, онда ол жалпы деп аталады.

Мысалы, дербес рекурсивті функцияны детерминирлі M Тьюринг машинасымен есептелген, келесі түрдегі дербес функция ретінде анықтаймыз:

• егер M машина енгізу x -те тоқтамаса, онда $f(x)$ анықталмаған болады және

• егер M машина енгізу x -те тоқтаса, онда машина тоқтау күйіне енген кездегі, M -нің таспасына жазылған мән $f(x)$ болады.

Жұптасу

Біз тек унитарлы функцияны $\omega \to \omega$ қарастырамыз. Жоғары арлы функциялар бірдің-бірге түйіндес функциясын $<>: \omega^2 \to \omega$ пайдаланып кодталуы мүмкін

$$< i, j > \overset{\text{def}}{=} \binom{i+j+1}{2} + i. \tag{33.1}$$

$i \backslash j$	0	1	2	3	4	5
0	0	1	3	6	10	15
1	2	4	7	11	16	
2	5	8	12	17		
3	9	13	18		$\ddots$	
4	14	19				
5	20					

Сәйкес проекциялар π_1^2 және π_2^2 арқылы белгіленеді:

$$\pi_1^2(< x, y >) \overset{\text{def}}{=} x \qquad \pi_2^2(< x, y >) \overset{\text{def}}{=} y.$$

$n > 2$ болғанда,

$$< x_1, \ldots, x_n > \overset{\text{def}}{=} < x_1 < x_2, \ldots, < x_{n-1}, x_n >>>,$$

деп аламыз және $m \le n$ болғанда,

$$\pi_1^n \overset{\text{def}}{=} \pi_1^2 \qquad \pi_m^n \overset{\text{def}}{=} \pi_{m-1}^{n-1} \circ \pi_2^2$$

деп аламыз.

Таңбалауды экономдау үшін $f(<x,y>)$ -тің орнына біз $f(x,y)$ деп жазатын боламыз және $f(<x_1,\dots,x_n>)$ -тің орнына $f(x_1,\dots,x_n)$ деп жазылады. Бірақ, есіңізде болсын, ашып айтар болсақ барлық дербес рекурсивті функциялар унитарлы.

Бұдан басқа таңба $<>$-ны қайта жүктеу арқылы функцияда есептелетін келесі жұптасу операторын анықтаймыз:

$$< f, \mathrm{g} > \overset{\text{def}}{=} \lambda x. < f(x), \mathrm{g}(x) > .$$

негізгі қасиеттері

Дербес рекурсивті функция тұрақты функцияларды $\kappa_c \overset{\text{def}}{=} \lambda x.c$ қамтиды және проекциялар π_1^2, π_2^2 композиция және жұптасуға қатысты тұйықталған болады (тұйықталудың басқа қасиеттерінің ішінде). Яғни, егер f және g дербес рекурсивті функциялар болса, онда $f \circ \mathrm{g} \overset{\text{def}}{=} \lambda x. < f(\mathrm{g}(x))$ және $< f, \mathrm{g} > \overset{\text{def}}{=} \lambda x. < f(x), \mathrm{g}(x) >$.

Тұрақты функциялар мен проекциялар жалпы болып табылады. Егер тек егер f және g екеуі де x-те анықталса, онда композиция $f \circ \mathrm{g}$ x-те анықталады. Егер тек егер f және g екеуі де x-те анықталса, онда жұптасу $< f, \mathrm{g} >$ x-те анықталады.

Гёдел нөмірлеуі

Дербес рекурсивті функцияның барлық саны санауға келеді, өйткені Тьюринг машинасының саны да саналады (немесе С программалау немесе λ-термдер, немесе μ-рекурсивті функциялар, ...). Дербес рекурсивті функцияның тізбелеуі

$$\varphi_0,\ \varphi_1,\ \varphi_2,\ \dots$$

Гёдел нөмірлеуі немесе жарамды индексация деп аталады, ол үшін келесі үш шарт орындалуы тиіс:

(i) Белгілі бір i-де әрбір рекурциялы функция φ_i болады.

(ii) Функцияның универсал қасиеті: Барлық i және x үшін,

$$U(i,x) = \varphi_i(x)$$

қатынасы орындалатын дербес рекурсивті функция U табылады.

(iii) s_n^m қасиет: Барлық i, $x_1,...,x_n$, $y_1,...,y_m$, үшін,

$$\varphi_{s_n^m(i,x_1,...x_n)}(y_1,...,y_m) = \varphi_i(x_1,...,x_n, y_1,...,y_m)$$

орындалатын жалпы рекурциялық функция s_n^m табылады.

Сан i Гёдел саны немесе φ_i функцияның индексі деп аталады.

Индекс i тек сан болса да, бізге i-ді алгоритм сипаттамасы немесе φ_i функциясын есептейтін машина ретінде қарастыру тиімді. Мысалы, i қандай да бір Тьюринг машинаны кодтасын делік, ол машина φ_i-ді есептеу немесе С программаны есептеу немесе осыған ұқсас дүние жасауы мүмкін. Дәл форма нақты индексациядан тәуелді және біз дәл форма үшін көп мазасызданып отырғанымыз жоқ, бізге керегі (i) - (iii) қасиеттер. Бізді қатты алаңдататын жайт ол – әрбір дербес рекурсивті функцияның, жоқ дегенде бір индексі болуы қажет (қасиет (i)), функцияны индексін ескере отырып, бірқалыпты симуляция жасауға болады (қасиет (ii), функцияның универсал қасиеті) және программадағы енгізудің бөлігін қатаң кодтауға болады (қасиет (iii), s_n^m қасиет).

Мысалы, Тюринг машинасымен көрсетілген индексацияны алайық. i-ді бинарлы жол ретінде жазған кезде, біз i-ді Тьюринг машинасының кодтала сипатталуы ретінде интерпретация жасай аламыз; осылай кодтаудың арнайы түрімен танысу үшін [61, 76]-ны қараңыз. Екілік көрсетілімі Тюринг машинасын кодтамайтын кез келген санды, осы схемаға сүйенсек, жалғыз күйлі тривиалды Тюринг машинасының индексі ретінде алуға болады екен. Бірақ, әрбір дербес рекурсивті функция Тюринг машинасымен есептелетіндіктен, ал әрбір Тюринг машинасының бинарлық сан түрінде кодталған сипаттамасы бар болатындықтан, (i)-ші қасиет орындалады. Және басқа i машинаның сипаттамасын және енгізу x-ті алып, x-те i сипаттамасымен симуляция жасайтын универсалды машина бар болғандықтан, (ii)-ші қасиет орындалады. Соңында, машинадағы енгізудің бір бөлігін соңғы бақылауда кодтауға болатындықтан, оларға іздеу кестесінде қабылдау-ашық болады, бұл индексациялау схемасы (iii)-ші қасиетті қанағаттандырады.

Жарамды индексациялайтын (iii)-ші қасиет s_n^m функцияның бар болатынын ұйғарады. Іс жүзінде s_1^1 функцияны ғана бар деп ұйғарса болар еді; қалғандарының бәрі анықталады. Мысалы,

$$s_2^3 \overset{\text{def}}{=} s_1^1 \circ < s_1^1 \circ < \pi_1^3, \pi_2^3 >, \pi_3^3 >,$$

деп алсақ болады. Сонда

$$\begin{aligned}
\varphi_i(x_1, x_2, y_1, y_2, y_3) &= \varphi_i(< x_1, < x_2, < y_1, y_2, y_3 >>>) \\
&= \varphi_{s_1^1(s_1^1(i,x_1),x_2)}(< y_1, y_2, y_3 >) \\
&= \varphi_{s_1^1 \circ < s_1^1 \circ < \pi_1^3, \pi_2^3 >, \pi_3^3 > (i,x_1,x_2)}(< y_1, y_2, y_3 >) \\
&= \varphi_{s_2^3(i,x_1,x_2)}(y_1, y_2, y_3).
\end{aligned}$$

Comp, Const және Pair

Біз f және g екі дербес рекурсивті функциялардың индекстерін пайдаланып, функциялар композициясы $f \circ g$ үшін индексті тиімді[1] түрде алуымызға болады. Басқаша айтсақ, жалпы рекурсивті функция comp табылып,

$$\varphi_{\text{comp}(i,j)} = \varphi_i \circ \varphi_j$$

қатынасы орындалады. Мұнда comp конструкциясы беріледі. Дербес рекурсивті функцияның $U \circ < \pi_1^3, U \circ < \pi_2^3, \pi_3^3 >>$ индексі m болсын. Сонда

$$\begin{aligned}
(\varphi_i \circ \varphi_j)(x) &= \varphi_i(\varphi_j(x)) \\
&= U(i, U(j, x)) \\
&= U \circ < \pi_1^3, U \circ < \pi_2^3, \pi_3^3 >> (i, j, x) \\
&= \varphi_m(i, j, x)
\end{aligned}$$

[1] *тиімділік = рекурсивті функциямен*

болады. Жалпы рекурсивті функциямен comp-ты санай алуымыз үшін:

$$\begin{aligned}\mathsf{comp} &\overset{\text{def}}{=} \lambda < i, j > .s_2^1(m,i,j) \\ &= \lambda x.s_2^1(\kappa_m(x), \pi_1^2(x), \pi_2^2(x)) \\ &= s_2^1 \circ < \kappa_m, \pi_1^2, \pi_2^2 > .\end{aligned}$$

Бұл функция жалпы болады, өйткені s_2^1 , κ_m, π_1^2, және π_2^2 функциялары да жалпы.

Егер бізге керек болып жатса, онда comp үшін де индекс алуға мүмкіндігіміз бар. s_2^1 -дің индексі ℓ болсын. Түрлендірсек

$$\mathsf{comp}(i,j) = s_2^1(m,i,j) = \varphi_\ell(m,i,j) = \varphi_{s_2^1(\ell,m)}(i,j),$$

сондықтан comp индексі $s_2^1(\ell,m)$ болады.

f и g индекстерін пайдаланып, екі дербес рекурсивті функциялар жұбы $< f, g >$ үшін тиімді индекс алуымызға болады және тұрақты функция κ_c үшін тиімді индексті c-дан аламыз. Басқаша айтсақ,

$$\varphi_{\mathsf{pair}(i,j)} = < \varphi_i, \varphi_j > \qquad \varphi_{\mathsf{const}(i)} = \kappa_i$$

қатынастары орындалатын, жалпы рекурсивті функциялар pair и const табылдады. Конструкциялдар pair және const, жоғарыда келтірілген comp конструкциясы сияқты анықталады және жаттығу ретінде қалдырылды (10-үй жұмысы, 1-жаттығу).

Рекурсия теоремасы

Рекурсивті функциялар теориясының интригалы аспектісінің біреуі ол өзіне-өзі тиістілік күші болып табылады. Рекурсия теоремасы деп аталатын жалпы теорема бар, оны қозғалмайтын нүкте жайлы теореманың баламасы десе де болады. Ол теоремада, кез келген жалпы рекурсивті функционалдың (индекстерге әсер ететін функция) қозғалмайтын нүктесі бар деп айтылады. Формалды түрде:

33.1-теорема (Рекурсия теоремасы). Кез келген жалпы рекурсивті σ функция үшін, i индекс табылып $\varphi_i = \varphi_{\sigma(i)}$ орындалады. Қосымша, мұндай i σ *-ның индексінен тиімді алынуы мүмкін.*

Бұл теореманың бірнеше қолданысын 34-лекцияда келтіреміз. Қазір қозғалмайтын нүкте жайлы Гёдел леммасымен ұқсастығын көрсетеміз, ол өз кезегінде бейтолықтылық теоремасын дәлелдеуде қолданылады ([76]-ны қараңыз). Қозғалыссыз нүкте жайлы лемма тұжырымын еске түсірейік: сандар теориясының бір тәуелсіз x айнымалылы кез келген формуласы $\Phi(x)$ үшін,

$$\mathbb{N} \models \Psi \leftrightarrow \Phi(\ulcorner\Psi\urcorner)$$

орындалатын сөйлем Ψ табылады, мұндағы $\ulcorner\Psi\urcorner$ таңбасы белгілі бір ақылға сыйымды кодтау жүйесінде Ψ-дің цифрлық коды болып табылады. (Ойланыңыз: «код» = «Гёдел саны»). Дұрысын айтсақ, қозғалыссыз нүкте жайлы лемма және рекурсия жайлы теорема әртүрлі формалдауда айтылған бірдей құбылыс, сондықтан олардың дәлелдемелері де өте ұқсас. Және де λ-есептеуде қозғалмайтын нүкте комбинаторымен тығыз байланыс бар $\lambda f.(\lambda x.f(xx)\ \lambda x.f(xx))$.

Рекурсия теоремасы Клинидікі [72] деп саналады ([73]-ті де қараңыз).

33.1-теореманың дәлелдемесі. h жалпы рекурсия функция болсын, ол v енгізуде функция индексін өндіреді, ал енгізу x-те:

(i) $\varphi_v(v)$-ды есептейді;

(ii) егер $\varphi_v(v)$ анықталған болса, σ $\varphi_v(v)$-ге қолданады; және

(iii) $\sigma(\varphi_v(v))$ индекс сияқты интерпретация жасайды және функцияны осы индекспен x-ке қолданады.

Сонымен, егер $\varphi_v(v)$ анықталған болса, онда

$$\varphi_{h(v)}(x) = \varphi_{\sigma(\varphi_v(v))}(x),$$

кері жағдайда анықталмаған. h-тың өзі жалпы рекурсиялы функция екендігін білген маңызды; ол (i) - (iii) қадамдардың ешқайсысын орындамайды, ол тек қадамдарды орындайтын функциялардың индекстерін есептейді.

Енді h-тың индексі u болсын. Сонда $h(u) = \varphi_u(u)$ ізделініп отырған қозғалыссыз σ нүкте болады: барлық x үшін,

$$\varphi_{h(u)}(x) = \varphi_{\sigma(\varphi_u(u))}(x) = \varphi_{\sigma(h(u))}(x).$$

Біз теореманың екінші тұжырымдамасын қалдырамыз, ол фактыны айтсақ: σ үшін қозғалыссыз нүкте индекстен тиімді алынады; оны жаттығу ретінде қалдырамыз (10-үй тапсырмасы, 2-жаттығу).

Біз рекурсия теоремасының бірнеше қолданысын келесі жолы келтіретін боламыз.

34-лекция

Рекурсия теоремасының қолданысы

Дербес рекурсивті функцияның $\varphi_0, \varphi_1, \ldots$ Гёдел нөмірлеуі болсын. Өткен жолғы рекурсия теоремасының тұжырымын еске түсірейік: кез келген рекурсивті функционал σ-ның қозғалмайтын нүктесі i бар болады, яғни $\varphi_i = \varphi_{\sigma(i)}$ болатын индекс i бар. Одан басқа, σ-ның индексі үшін i-ді тиімді таба аламыз.

Рекурсия теоремасы басында рекурсиямен анықталатын функцияның бар болатынын дәлелдеу тәсілі ретінде ойлап табылған еді (аты осыдан шыққан). Мысалы, факториалды функция жалпы рекурсиялы функцияның қозғалмайтын нүктесі болады.

$$\mathrm{P} \ \to \ \lambda x.\begin{cases} 1, & \text{егер } x{=}0 \text{ болса} \\ x \ \cdot \ \mathrm{P}(x-1), & \text{болмаса} \end{cases}$$

Біз бұл қолданысты 12-үй тапсырмасы, 12-жаттығуда зерттейтін боламыз. Дегенмен, рекурсия теоремасының басқа да көптеген алысқа сермейтін әсері бар. Ол өзіне-өзі тиістілік идеясын сығымдалған түрде қамтиды.

Өзін-өзі теретін бағдарлама

Өзіне-өзі тиістілік ұғымына мысал ретінде мұнда С-де жазылған программа келтірілген, ол өзін-өзі тереді:

```
char * s = "char * s = %c%s%c; %cmain() {printf (s, 34, s, 34, 10, 10) ; }%c";
main() {printf (s, 34, s, 10, 10) ; }
```

Мұнда 34 және 10 қос жақшаның және жаңа жолдың таңбасының сәйкес ASCII кодтары болып табылады. Негізінде, s жолды алауымыз қажет және жақшаның ішіне оның көшірмесін қоя салысымен оны теруге жіберу қажет дегенді **printf** тұжырымы айтады.

UNIX оболочкасының командасы арқылы келесі скрипте сол принцип иллюстрация жасалады. Оқуға жеңіл болуы үшін, ол артық пробелдар және жол үзілісімен жазылған; іс жүзінде өзін-өзі теретін бағдарлама осы бағдарламаның нәтижесі болып шығады.

```
x ='y = ‘echo . | tr . " \ 47"‘; echo "x = $y$x$y; $x"'
y = ‘echo . | tr . " \ 47"‘; echo "x = $y$x$y; $x"
```

Мұндай бағдарламалар кейде философ Уиллард ван Орман Куайнның құрметіне куайн деп аталады.

Мақсаты жалпы кез келген бағдарлама куайн конструкцияның қасиеттерін қамтиды. Дербес рекурсиялы функцияның кез келген Гёдел нөмірлеуінде, «өзін-өзі теретін бағдарлама» индекс i болады, мұнда барлық x үшін $\varphi_i(x) = i$ болады. Осындай i-ді іздеген жағдайда, 33-лекцияда құрастырылған функционал **const**-тың қозғалмайтын нүктесін пайдаланамыз.

Райс теоремасы

Райс теоремасы [102, 103], әрбір рекурсивті тізбеленетін (р.т.) жиынның бейтривиалды қасиеті шешілмейді деп тұжырымдайды. Осының дәлелдемесін рекурсия теоремасының көмегімен дәлелдеуге тырысамыз. Интуитивті, егер бейтривиалды қасиет шешілетін болса, онда қозғалмайтын нүктесі болмайтын рекурсивті функционал құрастыруға болар еді, ол рекурсия теоремасына қайшы келеді.

Дербес рекурсивті функцияның әрбір бейтривиалды қасиеті шешілмейтінін дәлелдейміз, мұнда айтып өткен рекурсивті функцияның бейтривиалды қасиеті ол $P:\omega\to\{0,1\}$ бейнені білдіреді, ол бейнеде егер $\varphi_i=\varphi_j$ болса, онда $P(i)=P(j)$ (сондықтан ол индекстер қасиеті емес, бірақ функциялар қасиеті болады). Және егер P не универсалды өтірік болмаса немесе универсалды ақиқат болмаса, онда P бейтривиалды болады.

P-да сондай қасиет деп ұйғарайық. P бейтривиалды болғандықтан, $P(i_0)=0$ және $P(i_1)=1$ орындалатын, i_0 және i_1 индекстері табылады. P шешіледі деп жорыйық. Сонда функция

$$\sigma(j)\ \stackrel{def}{=}\ \begin{cases} i_0, \text{егер } P(j)=1, \\ i_1, \text{егер } P(j)=0 \end{cases}$$

жалпы рекурсивті функция болады. Дегенмен, σ-ның жылжымайтын нүктесі болмайды: барлық j үшін,

$$P(\sigma(j))\ =\ \begin{cases} P(i_0)=0, \text{ егер } P(j)=1, \\ P(i_1)=1, \text{ егер } P(j)=0, \end{cases}$$

сондықтан $P(j)\neq P(\sigma(j))$. Ал P функция қасиеті болғандықтан, $\varphi_j\neq\varphi_{\sigma(j)}$ болады. Бұл рекурсия теоремасына қайшы келеді.

Минималды программалар

Кез келген ақылға қонымды «ең кіші» түсініктің анықтамасы үшін берілген программаға эквивалентті ең кіші бағдарламаны табатын алгоритм болмайды. Программалау тілінен тәуелсіз түрде бұл ақиқат. Мысалы, Тьюринг машинасы үшін берілген машинаға эквивалентті ең аз күйлі Тьюринг машинасын табатын алгоритм жоқ.

Жалпы түрде, кез келген Гёдел нөмірлеуі үшін, барлық j үшін φ_j-де $\sigma(j)$ минималды индекс болатын, жалпы рекурсиялы функция σ табылмайды. Мұнда «минималды» деп ω-да табиғи реттілік $\leq$-ге қатысты айтылған. Біз одан күштірек нәтижені дәлелдейміз:

34.1-теорема Кез келген Гёдел нөмірлеуінде, шексіз минималды индекстердің р.т. тізімі табылмайды.

Дәлелдеу. Осындай тізім нақты бар екен деп ұйғарайық. Енгізу x-те x-тің индексінен үлкен индекспен соқтыққанға дейін тізімді тізбелеп үлгеретін және оны өзінің мәні ретінде қабылдайтын, жалпы рекурсиялы σ функциясын қарастырамыз. Сонда σ-ның қозғалыссыз нүктесі болмайды: барлық x үшін $\varphi_x \neq \varphi_{\sigma(x)}$ болады, өйткені $x < \sigma(x)$ және $\sigma(x)$ минималды индекс. Бұл рекурсия теоремасына қайшы.

Тиімді толтыру

Егер берілген индекске эквивалентті ең аз индексті тиімді таба алмасаңыз да, сіз одан үлкенін барлық уақытта таба аласыз. Осыны тиімді толтыру деп айтады. Тюринг машинасымен және Java программасымен машинаны немесе программаны жеңіл толтыруға болады, яғни эквивалентті үлкенін табу қажет: Тюринг машинасы үшін жалған орындалмайтын әйтеуір бір күйді қосымшалау жеткілікті және программа үшін қандайда бір жалған орындалмайтын тұжырым қосыңыз. Бір сөзбен айтсақ, кез келген Гёдел нөмірлеуі осындай толтыру қасиетіне ие:

34.2 лемма Кез келген Гёдел нөмірлеуінде жалпы рекурсивті функция σ табылып, барлық x үшін $\sigma(x) > x$ және $\varphi_x = \varphi_{\sigma(x)}$ болады.

Дәлелдеу. Бізде x бар деп айталық. $\sigma(x)$-ты есептеу үшін бірнеше этаптан қатарлап өтуіміз қажет. Біз $B := \{x\}$ деп алып 0-ші стадиядан бастаймыз. Енді, белгілі бір этапта барлық $y \in B$ үшін $\varphi_y = \varphi_x$ болатындай етіп $B \subseteq \{0, 1, 2, \ldots, x-1, x\}$-ны құрастырдық деп ұйғарайық. Жалпы рекурсиялы функцияны

$$f(z) = \begin{cases} x+1, & \text{егер } z \in B, \\ x, & \text{егер } z \notin B, \end{cases}$$

қарастырамыз және f-тің қозғалыссыз нүктесі y болсын; яғни $\varphi_y = \varphi_{f(\mathrm{y})}$ орындалатын индекс бар. Біз y-ті рекурсия теоремасының тиімді версиясы арқылы ала аламыз. Егер $y > x$ болса, онда біз $\sigma(x) = y$ қатынасын ұстап тұратын етіп аяқтадық:

$$\varphi_y \quad = \quad \varphi_{f(\mathrm{y})} \quad = \quad \varphi_x,$$

Егер $y \in B$ болса, онда $\sigma(x) = x + 1$ деп аламыз және біз тағы да аяқтадық:

$$\varphi_{\sigma(x)} \quad = \quad \varphi_{x+1} \quad = \quad \varphi_{f(\mathrm{y})} \quad = \quad \varphi_y \quad = \quad \varphi_x.$$

Соңында, егер $y < x$ және $y \notin B$ болса, онда біз $B := B \cup \{y\}$ қатынасын орындай аламыз және процесті басынан қайталаймыз. Және де

$$\varphi_y \quad = \quad \varphi_{f(\mathrm{y})} \quad = \quad \varphi_x$$

болғандықтан, инвариант сақталады. B $x+1$-ден артық элемент қамти алмайтындықтан, бұл процесс шектеулі саннан көп рет орындала алмайды. Түбінде, x-тен артық қозғалмайтын нүктені табамыз.

Изоморфизм жайлы теорема

Біз рекурсивті изоморфизмнен басқа барлық Гёдел нөмірлеуі бірдей болатынын көрсеткеннен кейін, осы лекцияны аяқтаймыз. Бұл Роджерс ([104]-ті қараңыз) нәтижесі деп танылады.

34.3-теорема. Дербес рекурсивті функцияның екі Гёдел нөмірлеуі $\varphi_0, \varphi_1, \varphi_2, \dots$ және $\psi_0, \psi_1, \psi_2, \dots$ болсын. Сонда барлық i үшін $\varphi_i = \psi_{p(i)}$ орындалатын бірдің бірге және суръективті жалпы рекурсивті функция $\rho : \omega \to \omega$ табылады.

Дәлелдеу. φ нөмірлеуіне тән универсалды функция U болсын және ψ нөмірлеуде U-дың индексі ℓ болсын. Сонымен

$$\psi_\ell(i,x) \quad = \quad U(i,x) \quad = \quad \varphi_i(x)$$

орындалады. ψ нөмірлеуде s_1^1 функцияны қолдансақ

$$\psi_{s_1^1(\ell,i)}(x) \quad = \quad \psi_\ell(i,x) \quad = \quad \varphi_i(x)$$

болады. Сондықтан жалпы рекурсивті $\sigma \overset{\text{def}}{=} \lambda i.s_1^1(\ell, i)$ функция φ нөмірлеудегі индексті ψ нөмірлеудегі эквивалентті индекске бейнелейді; яғни барлық i-де $\varphi_i = \psi_{\sigma(i)}$ орындалады. Басқа бағыттағы тура осы конструкция, универсалды ψ нөмірлеудегі және φ нөмірлеудегі s_1^1 функцияны пайдаланып, барлық j үшін $\psi_j = \varphi_{\tau(j)}$ орындалатын етіп жалпы рекурциялы функция τ-ды өндіреді.

Біз қазір барлық i үшін $\varphi_i = \psi_{p(i)}$ орындалатын етіп, σ мен τ-ды жалғыз бірдің бірге жалпы рекурсивті және суръективті $\rho : \omega \to \omega$ функцияға біріктіреміз. Алға және артқа аргументін пайдаланып ρ-ны этаппен құрастырамыз. n-ші этаптан кейін, соңғы сәйкестікті $\rho : A_n \to B_n$ құрастырдық деп айтайық, мұндағы A_n және B_n-дер ω-ның n-элементті ішкі жиындары, ал A_n-да ρ бірдің бірге қатысы және барлық $i \in A_n$ үшін $\varphi_i = \psi_{p(i)}$ болады. Егер n жұп болса, онда $\sim A_n$-ның ең кіші элементі m болсын. $\sigma(m)$ бастап, ψ нөмірлеудің тиімді толтыру функциясын, бірінші индекспен $k \in \sim B_n$ соқтыққанға дейін қолданамыз (34.2-лемма). B_n шектеулі болғандықтан, бұл жағдай міндетті түрде орындалуы қажет. Дәл осылайша, егер n тақ болғанда, k ең кіші элемент болсын. $\tau(k)$-дан бастап φ нөмірлеудің тиімді толтыру функциясын, бірінші индекспен $m \in \sim A_n$ соқтыққанша қолданамыз. Сонда, барлық жағдайда $\varphi_m = \psi_k$, $m \notin A_n$ және $k \notin B_n$ болады. $\rho(m) = k$, $A_{n+1} = A_n \cup \{m\}$, және $B_{n+1} = B_n \cup \{k\}$ деп бекітейік. Сонмен, ρ-ның анықталу аймағының өлшемін бірге арттырдық және инвариантты сақтап қалдық.

Кезектесіп отыратындықтан және екі жақта да әрқашан сәйкес келмейтін ең кіші элементті өңдейтін болғандықтан, ең соңында әрбір элемент сәйкестеніп шыға келеді.

Изоморфизм жайлы Роджерс теоремасының альтернативті дәлелдемесі 108-аралас жаттығуда берілген.

Изоморфизм жайлы тағы да бір Майхилл [90] ([104]-ті де қараңыз) атымен белгілі теорема бар, ол изоморфизм жайлы Майхилл теоремасы деген атпен белгілі. Бұл жиындар теориясындағы Кантор-Шрёдер-Бернштейн теоремасының тиімді версиясы; егер бірдің бірге функциялар $A \to B$ және $B \to A$ бар болса, онда A мен B-ның қуаттары бірдей болады. Изоморфизм жайлы Майхилл теоремасы, бірдің бірге келтірімділігінің көмегі арқылы бір-біріне келтірілетін кез келген екі жиын рекурсивті изоморфты деп айтады (109-аралас жаттығу).

Изоморфизм жайлы Роджерс және Майхилл атымен аталатын екі теореманы да, олардан да жалпыланған басқа теоремадан жеке жағдай ретінде алуға болады (107-ші аралас жаттығу).

Қосымша J лекция

Абстракциялы күрделілік

Есептеу күрделілігін өлшейтін әртүрлі тәсілдер бар, соның ішінде ең кең таралғаны уақыт және кеңістік өлшемдерін пайдалану. Бұдан басқа мүмкін өлшемдер түрі: Тьюринг машинасының уақыт нөмірін таспалы ұяшыққа жазу («сиямен белгілеу») немесе логикалық схеманың өлшемі мен тереңдігі. Бұл күрделілік өлшемдерінің, дәл өлшемнен тәуелсіз белгілі бір ортақ қасиеттері бар. Мысалы, уақыт және кеңістік үшін орындалатын үдету теоремасы (32.2-теорема) және айырмашылық теоремасы (32.1-теорема). Бұл теоремалар, жеңіл түрде формалданатын аксиомаларды қанағаттандыратын, кез келген жалпы күрделілік өлшемі үшін де орындалады.

Күрделілік теориясы дамуының бастапқы уақытында, Блюм [16] осы құбылысты байқайды да, абстракты күрделілік өлшемін формалдауға әрекет жасап көрді, Оның мақсаты керекті қасиеттерді таза автоматты түрде алу болды. Өзінің қарапайым түрінде Блюм аксиомалары әрбір дербес рекурсивті функция φ_i үшін келесі қасиеттер орындалатындай етіп,

(i) барлық x үшін егер тек егер $\Phi_i(x)\downarrow$ болса, онда $\varphi_i(x)\downarrow$ болады;

(ii) $\Phi_i(x)=m$ теңдеуі *i, x*, және *m*-де біркалыпты шешіледі, функциялар жиынтығы Φ_i -ге шарттар бекітеді.

Шарт (ii) арқылы мынадай жағдайды білдіреміз: үш айнымалылы жалпы рекурсивті функция f табылып,

$$f(i,x,m) \;=\; \begin{cases} 1, \text{егер}\, \Phi_i(x)=m \;\text{ болса} \\ 0, \;\text{егер кері болса} \end{cases}$$

қатынасы орындалады. Жиынтық *Ф* абсракциялы күрделілік өлшемі деп аталады. Тюринг машинасының уақыт күрделілігі сөзсіз бұл аксиомаларды қанағаттандырады, осы тұжырымды Тюринг машинасының кеңістігі үшін де айтуға болады. Бұл жерде мынадай шарт қойылады: егер машина белгілі бір енгізуде тоқтамаса, онда біз пайдаланған кеңістік көлемінде осы енгізу анықталмаған деп санаймыз.

Φ_i функциясы дербес рекурсивті функция болып табылады; осыған қосымша *i*-ден Φ_i үшін тиімді индекс алуға болады (116-аралас жаттығу).

Егер абстракты күрделілік мерасы Ф берілсе, онда кез келген жалпы рекурсиялық функция f келесі күрделілік класын

$$\mathcal{C}_f^{\Phi} \;\overset{def}{=}\; \{\varphi_i \mid \Phi_i(n)\le f(n)\}$$

анықтайды. Бұларды есептейтін программа емес, функциялар класы екендігіне назар аударыңыздар; сонымен φ_i $\mathcal{C}_f^{\Phi}$ -да жатуы мүмкін, десек те ш.ж. $f(n)$ -нен $\Phi_i(n)$ асып түседі.

32-лекцияда баяндалған үдету және айырмашылық теоремаларын осы абстракциялы тұрғыдан қарап, басқаша формалдауға (айтуға) болады. Осы түрде абстракциялы айтылған айырмашылық теоремасының дәлелдемесі, 32.1-теоремасының тура жалпылама дәлелдемесі болып шығады және біз оны жаттығу ретінде қалдырамыз

(120-аралас жаттығу). Дегенмен, күрделілік мерасының негізгі қасиеттерін, Блюм аксиомасы қалайша тартып алды, соны анық көрсету үшін үдету теоремасының дәлелдемесін біршама өзгертеміз.

J.1-теорема (айырмашылық теоремасы [21]) Абстракциялы күрделілік мерасы Φ болсын. Кез келген жалпы рекурсивті функция $f(x) \geq x$ үшін, $\mathcal{C}_f^{\Phi} = \mathcal{C}_{f \circ t}^{\Phi}$ орындалатын жалпы рекурсивті функция t табылады. Басқаша айтсақ, кез келген жалпы рекурсивті функция $f(x) \geq x$ үшін, егер б.ж.д. $\Phi_i(x) \geq f(t(x))$ болса, онда $\varphi_j = \varphi_i$ және б.ж.д. $\Phi_j(x) \leq t(x)$ орындалатын индекс j үшін жалпы рекурсивті функция t табылады.

Дәлелдеу. 120-аралас жаттығу.

J.2-теорема (Үдеу жайлы теорема [17]) Абстракциялы күрделілік өлшемі Φ болсын. Барлық жалпы рекурсивті f үшін жалпы рекурсивті g табылып, *g-дің барлық индекстері i* үшін б.*ж.д.* $f(n, \Phi_j(n)) < \Phi_i(n)$ орындалатын, *g-дің басқа бір индексі j* табылады.

Дәлелдеу. Бұл дәлелдеу көбіне белгілі дәрежеде уақыт үшін айтылған үдеу теоремасының (32.2) дәлелдемесін имитация жасайды, айтқанымыз орындалмайтын жағдай да бар, ол абстракциялы күрделік өлшемінің аксиомалары деп пайымдаймыз. Бұл жағдай сол дәлелдемені оқыуға тырыспай-ақ, терең түсінуге көмектеседі.

Әрбір φ_r үшін g_r-ді алу үшін алдымен диогоналдаймыз. Диогоналдау тәртібі мынау, егер φ_r тоталды және g_r-дің i индексі болса, онда

$$\Phi_i(n) > \varphi_r(n-i) \text{ а.е.}^1 \tag{J.1}$$

болады.

0-ші стадия $g_r(0) = 0$ және $D_0 = \varnothing$ болсын (32.2-теорема дәлелдемесінде белсенді тізімнен алынып тасталған машиналарға мұндағы D сәйкес келеді).

---

[1] *32.2-теорема сияқты «б.ж.д.» мен «ш.ж.» n айнымалысына қатысты айтылады. Басқа айнымалылар, осы өрнектегі i сияқты айнымалылар, қозғалыссыз тұрақты ретінде қарастырылады.*

$n \geq 1$ стадия Ең аз i-ді таңдаймыз, егер ол бар болса, онда

(i) $i \leq n$
(ii) $i \notin D_{n-1}$
(iii) $\Phi_i(n) \leq \varphi_r(n-i)$.

Егер осындай i бар болса, онда $g_r(n) = \varphi_i(n) + 1$ және $D_n = D_{n-1} \cup \{i\}$ болсын. Егер ондай i жоқ болса, онда $g_r(n) = 0$ және $D_n = D_{n-1}$ болсын.

Жоғарыда келтірілген программаны g_r үшін $\varphi_{h(r,0,0)}$ арқылы белгілейік. Егер барлық $0 \leq i \leq n$ үшін $\varphi_r(i)\downarrow$ болса, онда $\varphi_{h(r,0,0)}(n)\downarrow$ болатынына назар аударыңыздар. Сондықтан егер φ_r тоталды болса, онда $\varphi_{h(r,0,0)}$-де тоталды және $\varphi_{h(r,0,0)}$-мен саналған g_r функциясы (J.1)-ді қанағаттандырады.

Осыдан кейін, егер φ_r тоталды болса, онда жиындағы барлық программалар тоталды болатын р.с. жиындар программасын құрастырамыз және соның ішінде g_r «ш.ж.» ретінде көрсетілген.

$1 \leq k \leq m$ үшін $\varphi_{h(r,k,m)}$-ді:

$0, \ldots, m-1$ стадиялармен анықтаймыз. $\varphi_{h(r,k,m)}$ -ді $\varphi_{h(r,0,0)}$ сияқты анықтап шығамыз.

$n \geq m$ стадия Ең аз i-ді таңдаймыз, егер ол бар болса, онда

(i) $k \leq i \leq n$
(ii) $i \notin D_{n-1}$
(iii) $\Phi_i(n) \leq \varphi_r(n-i)$.

Егер ондай i табылса, онда $\varphi_{h(r,k,m)}(n) = \varphi_i(n) + 1$ және $D_n = D_{n-1} \cup \{i\}$ болсын. Егер ондай i жоқ болса, онда $\varphi_{h(r,k,m)}(n) = 0$ және $D_n = D_{n-1}$ болсын.

Тағы да еске саламыз, егер $0 \leq i \leq n-k$ үшін $\varphi_r(i)\downarrow$ болса, онда $\varphi_{h(r,k,m)}(n)\downarrow$ болады.

Сондықтан, егер φ_r тоталды болса, онда $\varphi_{h(r,k,m)}$-де тоталды. Осыған қосымша, біз

$$\forall k \overset{\infty}{\forall} m \ \varphi_{h(r,k,m)} = g_r$$ [2]

болады деп тұжырымдаймыз. Бұл ақиқат, өйткені белгілі бір этап m-де $\varphi_{h(r,0,0)}$-ді есептеу кезінде, барлық $i \le k$ болады, олар D_m -де тұрғанымен әйтеуір бір уақытта D_j-да болады. Осының әсерінен, барлық D_j-ға кандидаттар $\varphi_{h(r,k,m)}$-мен таңдалады, және $\varphi_{h(r,0,0)}$-мен де таңдалады, сондықтан g_r функция қайта құрылады.

Түбінде жарамды φ_r-ді таңдап аламыз. Рекурсивті оператор σ-ны анықтайық:

$$\varphi_{\sigma(r)}(0) \overset{\text{def}}{=} 0$$

$$\varphi_{\sigma(r)}(n+1) \overset{\text{def}}{=} 1 + \max_{k,m \le n} f(n+k, \Phi_{h(r,k,m)}(n+k)).$$

Рекурсия теоремасына сай, σ-ның қозғалыссыз нүктесі болады, яғни

$$\varphi_r(0) \overset{\text{def}}{=} 0$$
$$\varphi_r(n+1) \overset{\text{def}}{=} 1 + \max_{k,m \le n} f(n+k, \Phi_{h(r,k,m)}(n+k)).$$

Индукция көмегімен φ_r тоталды болатынын көрсете аламыз. Анықтама бойынша $\varphi_r(0)\downarrow$ болатыны күмән туғызбайды, егер барлық $0 \le i \le n$ үшін,

---

[2] *Мұндағы* $\overset{\infty}{\forall}$ *«шектеулі көптен басқа, барлық үшін ...» дегенді білдіреді немесе «барлық жеткілікті үлкен үшін ...». Және де* $\exists$ *таңбасы бар, оның мағынасы «шексіз көп болады ...».*

$\varphi_r(i)\downarrow$ болса, онда $\varphi_{h(r,k,m)}(n+k)\downarrow$ болады, сондықтан $\varphi_r(n+1)\downarrow$. Бірақ барлық *k, m* және жеткілікті үлкен *n* үшін,

$$\varphi_r(n+1) \ > f(n+k,\Phi_{h(r,k,m)}(n+k)),$$

сондықтан

$$f(n,\Phi_{h(r,k,m)}(n)) \ > \ \varphi_r(n-k+1) \text{ б.ж.д.}$$

Ерекше жағдайда, φ_i g_r-ді есептейтін кез келген *i* үшін (J.1.)-ге сүйеніп

$$f(n,\Phi_{h(r,i+1,m)}(n)) \ < \ \varphi_r(n-i) \ < \ \Phi_i(n) \text{ б.ж.д.}$$

қатынасын аламыз.

Соңғы мысал ретінде қызықты жалпы теореманы келтірейік, ол барлық абстракциялы күрделілік өлшемін қамтиды, біз МакКрейт және Мейер [83] екеуінің біріктіру теоремасын айтып отырмыз. Бұл теорема, кез келген тиімді иерархияның бірігуі бір функциямен ғана анықталатын күрделілік класы болады деп тұжырымдайды.

Мысалы, бірігу теоремасының салдарының біреуін келтірейік, ол салдарда $DTIME(p(n)) = P$ орындалатын есептелетін функция *p* табылады дейді. Тағы да, *p* функциясы табиғи болмай шықты, $2^{(\log n)^2}$ сияқты немесе $n^{\log\log n}$ сияқты немесе ештеңеге ұқсас емес. Айырмашылық теоремасындағы *t* сияқты және үдеу теоремасындағы g сияқты ол күрделі диогоналдау арқылы құрастырылған.

J.3-теорема (Біріктіру жайлы теорема [83]). Кез келген *i* және *n* үшін $f_i(n) \le f_{i+1}(n)$ болатын, жалпы рекурсиялы функциялар $f_0, f_1, \ldots$ р.с. тізімі болсын. Сонда

$$\mathcal{C}_f^{\Phi} \ = \ \bigcup_i \mathcal{C}_i^{\Phi}$$

орындалатын жалпы рекурсиялы функция f табылады. Басқаша айтсақ, кез келген рекурсивті функция g үшін $\Phi_i(n) \ \le f(n)$ б.ж.д. орындалатын g функциясы үшін *i* индексі табылады, сонда тек сонда егер g үшін индекс j және сан k табылып, $\Phi_j(n) \le f_k(n)$ б.ж.д. орындалса.

Дәлелдеу. Диоганалдауды пайдаланып, жалпы рекурциялы f функциясын келесі екі шарт орындалатын етіп құрастырамыз:

(i) Барлық k үшін, $f(n) \le f_k(n)$ б.ж.д.

(ii) Барлық i үшін, егер барлық k үшін $\Phi_i(n) > f_k(n)$ ш.ж. болса, онда $f(n) < \Phi_i(n)$ ш.ж.

Бұл екі шарт f-ті екі қарама-қарсы бағыққа тартады: (i) шарт f үлкен болса екен дейді, ал (ii) шарт f кіші аз болса екен дейді. Десек те, бізге (i) -ді тек б.ж.д., ал (ii)-ні тек ш.ж. қанағаттандыру қажет болғандықтан, ол жай конструкция жасауда бізге аздаған жеңілдік береді. Егер екі шартты да қанағаттандыратын f-ті құрастыра алсақ, онда өз мақсатымызға жетеміз, өйткені (i) шарты $\mathcal{C}^{\Phi}_{f_j} \subseteq \mathcal{C}^{\Phi}_{f}$ болуына кепілдік береді, ал (ii)-нің айтатыны: егер $\Phi_i(n) \le f(n)$ б.ж.д. болса, онда белгілі бір j үшін $\Phi_i(n) > f_j(n)$ б.ж.д. болады; сондықтан $\mathcal{C}^{\Phi}_{f} = \bigcup_i \mathcal{C}^{\Phi}_{f_i}$.

Енді біз f конструкциясын қарастыруға көшеміз. J.2-теоремасының дәлелдемесіндегі сияқты, f-ті диогоналдау арқылы құрастырамыз. Біз конъюкция ретінде қарастыруға болатын, (i,k), $i \le k$ жұбының кезегін сақтаймыз және ол $\Phi_i(n) \le f_k(n)$ б.ж.д. Конъюкция бұзылған кезде, f-тің анықтамасын пайдаланып, түзету амалдарын жасаймыз және оны әлсіздеу конъюкциямен алмастырамыз.

0 стадия *f(0):=0*-ді анықтаймыз және жалғыз (0,0) жұбын қамтитын етіп, кезекті инициализация жасаймыз.

$n \ge 1$ стадия. Кезекте бірінші тұрған, n-де бұзылған конъюкция (i,k)-ны табамыз; яғни $\Phi_i(n) > f_k(n)$ болады. Егер ондай (i,k) табылса, онда $f(n) := f_k(n)$ анықтап, (i,k)-ны кезектен алып тастаймыз және кезек соңына $(i,k+1)$-ны тіркейміз. Егер ондай (i,k) табылмаса, онда $f(n) = f_n(n)$ анықтаймыз. Барлық жағдайда кезек соңына (n,n)-ді тіркейтін боламыз.

Кез келген m үшін кезекте $k \le m$ болғанда шектеулі санды конъюкция (i,k) табылады (дәлірек айтсақ $\binom{m+2}{2}$). Конъюкция кезектен жойылғаннан кейін, ол ешуақытта қайтып оралмайды. Егер кезекте конъюкция шексіз жиі бұзыла берсе, онда ол соңында жойылу үшін таңдауға алынады. Егер конъюкция белгілі бір этапта бұзылған болса, онда осы этапта оның жойылуға таңдап алынбай қалуының тек бір жолы бар, ол үшін осы конъюкция үшін жойылуға таңдалып кеткен ағымдағы конъюкция алдында тұратын конъюкция бар болуы қажет, бірақ мұндай жағдай шектеулі сан рет мүмкін. Белгілі бір кезеңде, кезектен жойылуы қажет $k \leq m$ болатын, барлық конъюкция (i,k) жойылып бітеді, сондықтан $f(n) = f_m(n)$. Бұл (i)-ді қалыптастырады.

(ii) үшін, егер барлық k үшін $\Phi_i(n) > f_k(n)$ ш.ж. болса, онда $i \leq k$ болғанда (i,k) кон юкциялар соңында кезекке барып түседі және жойылады. Конъюкция (i,k) жойылған кезде, $f_k(n) < \Phi_i(n)$ орындалуы үшін $f(n)$ анықталады, сондықтан $f(n) < \Phi_i(n)$. Бұл шексіз сан рет орындалады, сондықтан $f(n) < \Phi_i(n)$ ш.ж.

Егер f_k-да монотондық шарты болмаса, онда бірігу теоремасы орындалмайды (126-аралас жаттығу).

Абстракциялы күрделілік мерасының көптеген теориясы 116-127 аралас жаттығуларда қарастырылады.

35-лекция

Арифметикалы иерархия

A, B жолдар жиыны болсын. Егер оракул B мен бірге белгілі бір оракул M ТМ үшін $A = L(M^B)$ орындалса, онда B-да A р.с. (рекурсивті саналады) деп айтамыз. Егер оракул B мен бірге белгілі бір оракул M ТМ үшін $A = L(M^B)$ орындалса және M^B толық болса, онда B-да A рекурсивті деп айтамыз. Басқаша айтсақ, егер A-да жатушылық B оракулы арқылы шешілсе, онда B-да A рекурсивті дейміз. Егер A жиын B-да рекурсивті болса, онда $A \leq_{\text{T}} B$ деп жазамыз. Қатынас $\leq_{\text{T}}$ - Тьюринг келтірімі деп аталады.

Біз р.с. жиында кластар иерархиясын полиномиалды иерархияға сәйкестендіріп, келесі түрде анықтаймыз. Алфавитті $\{0,1\}$ деп белгілейміз және натурал сандармен қоса $\{0,1\}^*$-де жолдарды да анықтаймыз.

Анықтаймыз

$$\Sigma_1^0 \overset{\text{def}}{=} \{\text{р.с. жиын}\},$$

$$\Delta_1^0 \overset{def}{=} \{\text{рекурсивті жиын}\},$$

$$\Sigma_{n+1}^0 \overset{def}{=} \left\{L(M^B) \mid B \in \Sigma_n^0\right\}$$

$$= \{\text{белгілі бір } B \in \Sigma_n^0 \text{ -да } A \mid \text{А р.с.}\},$$

$$\Delta_{n+1}^0 \overset{def}{=} \left\{L(M^B) \mid B \in \Sigma_n^0,\ M^B - \text{толық}\ \right\}$$

$$= \left\{A \mid A,\ \text{белгілі бір В-да рекурсивті}\, B \in \Sigma_n^0\right\}$$

$$= \left\{A \mid A \leq_{\text{T}} B \text{ белгілі бір В үшін } B \in \Sigma_n^0\right\},$$

$$\Pi_n^0 \overset{def}{=} \left\{\Sigma_n^0\text{-де қосымша жиын}\right\}.$$

Сонымен, Π_1^0- ко-р.с. (корекурсивті саналады) жиындар класы болып шықты. Ал Σ_n^0, Π_1^0 және Δ_n^0 арифметикалық иерархия деген атпен белгілі кластарды құрайды.

Кванторлар кезектесуін пайдаланып арифметикалық иерархияның түсініктілеу сипаттамасын береміз. Бұл сипаттамалар 10.2-теоремада берілген полиномиалды иерархия сипаттамасын еске түсіреді.

Егер тек егер A р.с. болса, онда

$$A = \left\{x \mid \exists y\ R(x,y)\right\} \tag{35.1}$$

болатын бинарлы предикат R табылатынын еске сала кетейік. Мысалы, тоқтау және қамту проблемасы келесі түрде жазылады

$$\text{HP} = \left\{M\#x \mid \exists t\ M\ \ x\text{-те } t\text{-қадамнан кейін тоқтау}\right\},$$

$$\text{MP} = \left\{M\#x \mid \exists t\ M\ \ t\text{-қадамнан кейін } x\text{-ке қабылдау ашық}\right\}.$$

«x -те M тоқтайды» деген предикат шешілмейді, ал «М машинасы x-те t-қадамнан кейін тоқтайды» десек ол шешіледі, соған назар аударайық. Өйткені енгізу x-те t қадам үшін универсал машина көмегімен M-ді симуляция жасаймыз да, осы уақытта ол тоқтайды ма соны көре аламыз. Альтернативті түр

$$\mathrm{HP} = \{M\#x \mid \exists v \;\; v - x\text{-те } M\text{-ді есептеуді тоқтату тарихы}\},$$

$\mathrm{MP} = \{M\#x \mid \exists v \;\; v - x$-те M-ді есептеу тарихына қабылдау-ашық $\}$.

(35.1) формасы арқылы өрнектеуге болатын жиындар тобының (үйірі) класы Σ_1^0 арқылы жазылады.

Дәл осы жолмен, элементар логикадан Π_1^0 ко-р.с. жиындар тобы екендігі шығады. Яғни, оны шешетін бинарлы предикат R табылып,

$$A = \{x \mid \exists y \; R(x, y)\} \tag{35.2}$$

қатынасы орындалатын, барлық A жыйындар тобының класы болып шығады.

Егер тек егер жиын р.с. және ко-р.с. болса, онда ол рекурсивті болатыны бізге белгілі. Жаңа белгілеуді қолдансақ,

$$\Delta_n^0 = \Sigma_1^0 \cap \Pi_1^0 .$$

Бұл нәтижелер төменде дәлелденетін теореманың жеке жағдайы, ол 32-аралас жаттығуда келтірілген PH сипаттамасына қатты ұқсайды.

35.1-теорема (i) Егер тек егер $(n+1)$-арлы шешетін предикат R табылып

$$A = \{x \mid \exists y_1 \; \forall y_2 \; \exists y_3 \; \cdots \; \mathrm{Q} y_n \; R(x, y_1, \ldots, y_n)\}$$

қатынасы орындалса, онда A жиыны Σ_n^0*-де жатады, мұндағы* $\mathrm{Q} = \exists$ егер *n* тақ болса, $Q = \forall$ егер *n* жұп болса.

(ii) Егер тек егер $(n+1)$-арлы шешілетін предикат R табылып

$$A = \{x \mid \forall y_1 \; \exists y_2 \; \forall y_3 \; \cdots \; \mathrm{Q} y_n \; R(x, y_1, \ldots, y_n)\}$$

қатынасы орындалса, онда A жиыны Π_n^0*-де жатады, мұндағы* $\mathrm{Q} = \forall$ егер *n* тақ болса, $Q = \exists$ егер *n* жұп болса

(iii) $\Delta_n^0 = \Sigma_n^0 \cap \Pi_n^0 .$

Дәлелдеу. 128-аралас жаттығу.

Мұндағы PH -тан айырмашылықтың біреуі, бұл жерде бізге PH -та қарастырылатын альтернативті ТМ сияқты арифметикалық иерархияның сипаттамасы жетіспейді, осы жағдай айырмашылықтың біреуі болып саналады. Бұл жетіспеушілік 39-лекцияда түзетіледі.

35.2-мысал. EMPTY $\stackrel{def}{=} \{M \mid \forall x \forall t$ M t қадамда x-ті қабылдамайды $\}$ болғандықтан, EMPTY $\stackrel{def}{=} \{M \mid L(M) = \varnothing\}$ жиыны Π_1^0 -де орналасады.

33-лекцияда баяндалған, (33.1)-де көрсетілген функциялар жұбының бірдің бірге есептеуін қолдана отырып, екі $\forall x \forall t$ кванторды бір кванторға біріктіруге болады. Сонымен

EMPTY $\stackrel{def}{=} \{M \mid \forall z$ M $\pi_1^2(z)$ -қа $\pi_2^2(z)$ қадамда қабылдау-жабық$\}$.

35.3-мысал. TOTAL $\stackrel{def}{=}$ $\{M \mid \forall x \exists t$ x-те t-қадамнан кейін M тоқтайды $\}$ болатындықтан, TOTAL $\stackrel{def}{=}$ $\{M \mid M$ толық$\}$ толығымен Π_2^0-де орналасады.

35.4-мысал. FIN $\stackrel{def}{=}$ $\{M \mid \exists n \forall x$ үшін егер $|x| > n$ болса, онда $x \notin L(M)$ болады$\}$
$= \{M \mid \exists n \forall x \forall t$ $|x| > n$ немесе t қадамда x-ке M-де қабылдау-жабық$\}$
болатындықтан, FIN $\stackrel{def}{=} \{M \mid L(M)$ шектеулі$\}$ жиыны Σ_2^0 -де орналасады.

Тағы да екі универсал кванторды (33.1)-дегі функциялар бірдің бірге қатынасын пайдаланып, бір кванторға біріктіруге болады.

35.5-мысал. Егер жиынның толықтауышы шектеулі болса, онда ол жиынды ко-*шектеулі* дейміз.

COF $\stackrel{def}{=}$ $\{M \mid \exists n \forall x$ егер $|x| > n$ болса, онда $x \in L(M)\}$
$= \{M \mid \exists n \forall x \exists t$ $|x| > n$ немесе t қадамда x-ке M-де қабылдау-ашық$\}$

болғандықтан, $\mathrm{COF} \overset{def}{=} M \mid L(M)$ ко-шектеулі$\}$ жиын Σ_3^0-де орналасады.

35.1-суретте ең төменгі бірнеше иерархия деңгейіндегі қамтулар бейнеленген. Әрбір иерархия деңгейі қатаң түрде келесіде орналасады; яғни $\Sigma_n^0 \cup \Pi_n^0 \cup \Delta_{n+1}^0$, бірақ $\Sigma_n^0 \cup \Pi_n^0 \neq \Delta_{n+1}^0$. Біз ко-р.с. жиын болмайтын р.с. жиындар бар екенін білеміз (мысалы, HP) және р.с. жиын болмайтын ко-р.с. жиындар (мысалы, $\sim\mathrm{HP}$). Сондықтан белгіленген қамтуларды пайдаланып, Σ_1^0 мен Π_1^0-ді салыстыра алмаймыз. Дәл осы жолды пайдаланып, кез келген n үшін белгіленген қамтулар аясында Σ_n^0 мен Π_n^0 салыстыруға келмейтінін көрсетуге болады (11-үй жұмысы, 2-жаттығу).

Келтірімділік және толықтылық

$A \subseteq \Sigma^*$ және $B \subseteq \Gamma^*$ үшін $A \leq_{\mathrm{m}} B$ қатынасын анықтаймыз; егер жалпы рекурциялы функция $\sigma : \Sigma^* \to \Gamma^*$ табылып, барлық $x \in \Sigma^*$ үшін

$$x \in A \quad \Leftrightarrow \quad \sigma(x) \in B$$

орындалса, онда $A \leq_{\mathrm{m}} B$ деп айтамыз. Қатынас $\leq_{\mathrm{m}}$ көп мәнді келтірімділік деп аталады және ол бұрын зерттелген келтірімділік қатынастар $\leq_{\mathrm{m}}^{\log}$ немесе $\leq_{\mathrm{m}}^{\mathrm{p}}$ сияқты, бір айырмашылық - мұнда ресурсқа шектеу қойылмайды.

Қатынастар мәселесі $\mathrm{MP} \overset{def}{=} \{M\#x \mid M$-де x-қабылдау ашық$\}$ бір жағынан шешілмейтін болса, екінші жағынан «өте күрделі» р.с. жиын болып табылады. Өйткені кез келген басқа $\leq_{\mathrm{m}}$ р.с. жиын, осы проблемаға келтіріледі: кез келген Тьюринг машинасы М үшін қатынас $x \mapsto M\#x \;\; L(M)$-ді MP-ға келтіретін тривиалды есептелетін қатынас.

Егер кез келген $\leq_{\mathrm{m}}$ р.с. жиынға келтірілетін жиын табылса, онда ол р.с.- күрделі жиын деп аталады. Басқаша айтсақ, егер кез келген р.с. жиын A үшін $A \leq_{\mathrm{m}} B$ қатынасы орындалса, онда жиын B р.с.-күрделі болады. Бұрын байқағанымыздай, MP-да жату мәселесі р.с.-күрделі. $\leq_{\mathrm{m}}$-ні қамту мәселесіне келтірілетін басқа кез келген мәселелер да осы қасиетке ие болады (мысалы, *HP*-ны тоқтату мәселесі), оған себеп болып отырған $\leq_{\mathrm{m}}$ қатынасының транзитивтілігі.

Егер жиын B бір уақытта р.с. жиын және р.с.-күрделі болса, онда ол р.с.-толық деп аталады. Мысалы, MP және HP екеуі де р.с. толық болады.

Жалпылау жасайық, айталық $\mathcal{C}$ кластар жиыны болсын. Егер барлық $A \in \mathcal{C}$ үшін $A \leq_\mathrm{m} B$ орындалса, онда B жиыны $\mathcal{C}$ үшін $\leq_\mathrm{m}$ *-күрделі* (немесе қысқаша $\mathcal{C}$-күрделілік) деп аталады. Егер $\mathcal{C}$ үшін B $\leq_\mathrm{m}$-күрделі және $B \in \mathcal{C}$ болса, онда $\mathcal{C}$ үшін B жиыны $\leq_\mathrm{m}$-толық (немесе қысқаша $\mathcal{C}$-толық) деп айтамыз.

5.3-леммамен сәйкестендірілген теореманы дәлелдеуге болады. Онда былай делінеді: егер $A \leq_\mathrm{m} B$ және $B \in \Sigma_n^0$ болса, онда $A \in \Sigma_n^0$, және егер $A \leq_\mathrm{m} B$ және $B \in \Delta_n^0$ болса, онда $A \in \Delta_n^0$ болады. Өйткені біздің білуімізше арифметикалық иерархия қатаң (әрбір деңгей келесі деңгейде дұрыс орналасады), егер Σ_n^0 үшін B $\leq_\mathrm{m}$-толық болса, онда $B \notin \Pi_n^0$ (немесе Δ_n^0 немесе Σ_{n-1}^0) болады.

Жоғарыда көрсетілген әрбір мәселе, иерархия деңгейі үшін $\leq_\mathrm{m}$ -толық болып шықты, оған ол өздігінен барып түседі:

(i) HP Σ_1^0 үшін $\leq_\mathrm{m}$ – толық,
(ii) MP Σ_1^0 үшін $\leq_\mathrm{m}$ – толық,
(iii) EMPTY Π_1^0 үшін $\leq_\mathrm{m}$ – толық,
(iv) TOTAL Π_2^0 үшін $\leq_\mathrm{m}$ – толық,
(v) FIN Σ_2^0 үшін $\leq_\mathrm{m}$ – толық және
(vi) COF Σ_3^0 үшін $\leq_\mathrm{m}$ – толық.

Иерархия қатаң сақталатындықтан, осы мәселелердің ешқайсысы өзінен төмен орналасқан иерархияның класында жатпайды, немесе өзінен төмен орналасқан иерархияның класының кез келген мәселесінде толығымен келтірілмейді. Егер осылай бола қалса, онда осы деңгейде иерархия бұзылады. Мысалы, EMPTY HP-ге келтірілмейді және COF FIN-ге келтірілмейді.

Біз (ii)-ні жоғарыда дәлелдеп өттік. (v)-ті осы лекцияда, ал (vi)-ны 36-лекцияда дәлелдейміз; қалғандарын жаттығу ретінде қалдырамыз (130-шы аралас жаттығу).

Аяқталу $\exists\forall$ предикатының көмегімен өрнектелетін болғандықтан, біз $\mathrm{FIN} \in \Sigma_2^0$ болады деп ұйғарған болатынбыз. Σ_2^0 үшін FIN $\leq_\mathrm{m}$ -күрделі болатынын көрсету үшін, Σ_2^0-нің кез келген жиыны оған келтірілетінін көрсетуіміз қажет. Біз 35.1-теоремадағы сипаттаманы пайдаланамыз. Жиын

$$A \;=\; \left\{x \in \Gamma^* \mid \exists y \forall z\ R(x,y,z)\right\}$$

Σ_2^0-нің кез келген жиыны болсын, мұндағы $R(x,y,z)$ шешілетін үштік предикат. R-ді шешетін жалпы машина M болсын. Егер тек егер $N \in \mathrm{FIN}$ болса, сонда $x \in A$ болатын етіп, берілген x үшін тиімді түрде N машинасын құрастыруымыз қажет. Сонымен, егер тек егер $\exists y \forall z\ R(x,y,z)$ болса, онда N шектеулі жиын қабылдауын қалаймыз; осыған эквивалентті, егер тек егер $\exists y \forall z\ \neg R(x,y,z)$ болса, онда N шексіз көп жиын қабылдауын қалаймыз. N машинасы ω енгізуінде:

(i) ұзындығы $|\omega|$-дан аспайтын барлық y жолдарын жазады; содан кейін

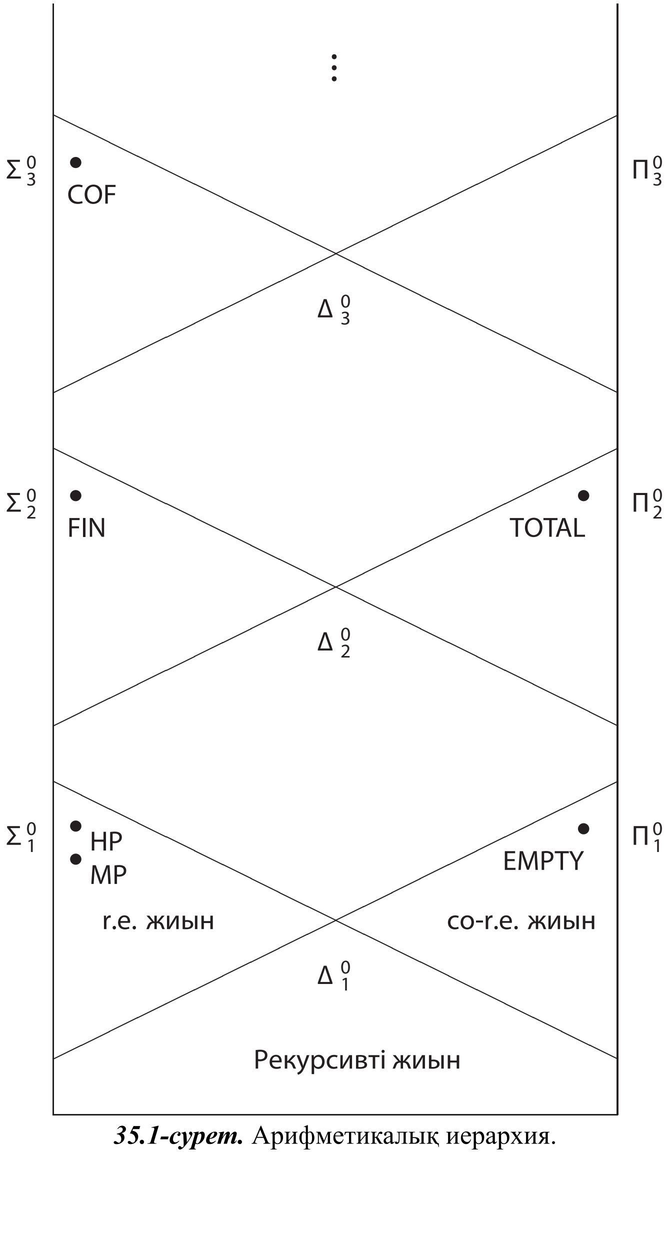
(ii) әрбір осындай y үшін $\neg R(x,y,z)$ орындалатын z тапқысы келеді (яғни, M-де $x\#y\#z$-ке қабылдау жабық), және барлық ұмтылыс оң нәтиже берсе, онда қабылдау-ашық. N машинасы x-ті біледі және M-нің сипаттамасын өзінің соңғы бақылауында сақтайды.

Қадам (ii) орындалған кезде, ұзындығы $|\omega|$-дан аспайтын әрбір y жол үшін N машина z жолын белгілі бір ретпен санап шығады және $x\#y\#z$ үшін M-ді іске қосады, ол M-де қабылдау-жабық болатын, z-ті тапқанша есептей береді. M жалпы болғандықтан, N уақытты бөлумен айналыспайды; ол алаңсыз z-тарды кезекпен өңдей береді. Егер қажетті z табылмаса, онда N тоқтаусыз есептей береді. Осындай N-ді M мен x арқылы тиімді түрде құрастырып шығуға болады.

Енді егер $x \in A$ болса, онда барлық z үшін, $R(x,y,z)$ болатын y табылады (яғни, барлық z үшін M-де $x\#y\#z$-ке қабылдау ашық); сондықтан $|\omega| \geq |y|$ болғанда, (ii)-ші қадам орындалмайды. Бұл жағдайда N шектеулі жиынды қабылдайды. Екінші жағынан, егер $x \notin A$ болса, онда барлық y үшін $\neg R(x,y,z)$ болатын z табылады, және оның бәрі (ii)-ші қадамда табылған. Бұл жағдайда N машинасында Γ^*-ға қабылдау-ашық.

Біз, егер $x \in A$ болса, онда $L(N)$ шектеулі жиын дедік және егер $x \notin A$ болса, онда Γ^* шектеулі дедік, сондықтан $x \mapsto N$ түрлендіруі A-дан FIN-ге дейін $\leq_{\mathrm{m}}$-келтірімділікті көрсетеді. Өйткені A жиыны Σ_2^0-нің кез келген элементі, ал Σ_2^0 үшін FIN $\leq_{\mathrm{m}}$-күрделі болатын.

Осындай мәлімет Π_2^0 үшін TOTAL-да $\leq_{\mathrm{m}}$-күрделі болатынын көрсетеді, соған назар аударыңыздар. Оған себеп, жоғарыда келтірілген конструкцияда, $A \in \Sigma_2^0$ болғандықтан, егер тек егер N жалпы болса, онда $x \in \ \sim A$, және $\sim A \in \Pi_2^0$ болады.



35.1-сурет. Арифметикалық иерархия.

36-лекция

Арифметикалық иерархияның толық мәселесі

Бұл лекцияда арифметикалық иерархияның үшінші деңгейі үшін табиғи толықтылық мәселесін береміз. Соңғы берген анықтамаларды еске түсірейік:

$$\Sigma_{n+1}^0 \overset{def}{=} \left\{ A \mid A \text{ белгілі бір } B \in \Sigma_n^0\text{-да р.с.} \right\}$$

$$= \left\{ L(M^B) \mid B \in \Sigma_n^0 \right\},$$

$$\Delta_{n+1}^0 \overset{def}{=} \left\{ A \mid A \text{ белгілі бір } B \in \Sigma_n^0\text{-да рекурсивті} \right\}$$

$$= \left\{L(M^B) \mid B \in \Sigma_n^0,\ M^B \text{ толық}\right\}$$
$$= \left\{A \mid A \leq_{\mathrm{T}} B \text{ белгілі бір } B \in \Sigma_n^0 \text{ үшін}\right\},$$
$$\Pi_n^0 \overset{def}{=} \left\{A \mid \ \sim A \in \Sigma_n^0\right\},$$

және $\Sigma_1^0 = \left\{\text{р.с. жиын}\right\}$ және $\Delta_1^0 = \left\{\text{рекурсивті жиын}\right\}$. Келесі белгілеулерді енгіземіз:

$M(x)\downarrow$ енгізу x-те M тоқтайды,

$M(x)\downarrow^t$ t қадам ішінде енгізу x-те M тоқтайды,

$M(x)\uparrow$ енгізу x-те M тоқтамайды,

$M(x)\uparrow^t$ t-дан кем емес қадамда енгізу x-те M тоқтамайды .

Мысалы, тоқтау мәселесі HP $\overset{def}{=} \left\{M\#x \mid M(x)\downarrow\right\}$ жиыны болады.

Соңғы лекцияда қарастырылған кез келген жиынның релятивтелген версиясын (нұсқасын) анықтай аламыз. Мысалы, оракул А үшін шектеулілік мәселесі

$$\mathrm{FIN}^A \overset{def}{=} \left\{M \mid L\left(M^A\right) \text{ шектеулі}\right\}$$

жиынымен анықталады.

36.1-лемма Σ_3^0 үшін $\mathrm{FIN}^{\mathrm{HP}}$ $\leq_{\mathrm{m}}$-толық болады. Жалпылап айтсақ, егер Σ_n^0 үшін A $\leq_{\mathrm{m}}$-толық болса, онда Σ_{n+2}^0 үшін FIN^A $\leq_{\mathrm{m}}$-толық болады.

Дәлелдеу. $\mathrm{FIN}^A \in \Sigma_{n+2}^0$ болатынын дәлелдеу үшін,

$$M \in \mathrm{FIN}^A \ \Leftrightarrow\ L(M^A) \text{ шектеулі болады} \Leftrightarrow\ \exists y \forall z \geq y \forall t\ M^A(z)\uparrow^t \quad (36.1)$$

деп ұйғарамыз (жалпылылықты сақтай отырып, ауытқу күйі болмайтын машинаны қарастырамыз, сондықтан тоқтау және қабылдау-ашық синоним болып кетеді). Оракул A Σ_n^0-толық, және предикат $M^A(z)\uparrow^t$ A-да рекурсивті болатындықтан, ол Δ_{n+1}^0-ны береді. 35.1 (iii) теорема бойынша ол Π_{n+1}^0-да да рекурсивті болады және 35.1 (ii) теоремаға сүйенсек, оны $\forall$ басталатын және рекурсивті предикатқа бағына отырып жасалатын кванторлардың $n+1$ кезектесуі арқылы өрнектеуге болады. Осыны (36.1)-дегі Σ_2 кванторлы префикспен $\exists y\ \forall z \geq y\ \forall t$ комбинациялай отырып, кезекті рекурсивті предикатпен Σ^0_{n+2} кванторлы префиксті аламыз. Осы жағдай FIN^A -тің Σ^0_{n+2} -де жататынын көрсетеді.

FIN^A -тің Σ^0_{n+2} -күрделі болатынын көрсету үшін, алдымен FIN -нің Σ^0_2 -күрделі болатындығының дәлелдемесін еске түсіреміз. Біз Σ^0_2 -дегі кез келген жиынды FIN -ге дейін амалсыздан келтірдік. Яғни, біз кез келген рекурсивті предикат $R(x,y,z)$ үшін

$$\left\{x \mid \exists y\ \forall z\ R(x,y,z)\right\} \quad \leq_{\mathrm{m}} \quad \left\{M \mid L(M) \text{ шектеулі болады}\right\}$$

мәліметін құрастыруымыз қажет болды. Соның әсерінен, жалпы рекурсивті σ функциясын анықтауымыз қажет болды. Сонымен, кез келген x үшін M машинаның $\sigma(x)$ сипаттауы болады және келесі шарт орындалады:

$$\exists y\ \forall z\ R(x,y,z) \quad \Leftrightarrow \quad L(M) \text{ шектеулі болады.}$$

Енгізу x берілген кезде, енгізу ω-да ұзындығы $|\omega|$ -дан аспайтын барлық y-ті санайтын M-ді құрастырған едік және әрбір осындай y үшін $\neg R(x,y,z)$ болатын z-ті табуға тырыстық. Сондықтан, егер $\exists y\ \forall z\ \neg R(x,y,z)$ болса, онда $L(M) = \Sigma^*$; бірақ, екінші жағынан егер $\exists y\ \forall z\ R(x,y,z)$ болса, онда ұзындығы y -ке еседен кем болатын барлық енгізу ω үшін M машинасы шексіз айналдыра есептейді, сондықтан ол шектеулі жиынды қабылдайды.

Енді біз дәл осындай конструкцияны оракула A болған жағдайға құрастыра аламыз. Егер $R^A(x,y,z)$ A -да рекурсивті болса, онда ол

$$\left\{x \mid \exists y \forall z\ R^A(x,y,z)\right\} \quad \leq_{\mathrm{m}} \quad \left\{M \mid L\left(M^A\right) \text{ шектеулі}\right\}$$

мәліметін береді. Енгізу ω-да ұзындығы $|\omega|$ -дан аспайтын барлық y-ті санайтын оракул машина M^A -ны құрастырамыз және ол әрбір осындай y үшін $\neg R^A(x,y,z)$ болатын z -ті табуға тырысады. M^A машинасы $R^A(x,y,z)$-ты анықтау қажет болғандықтан, өзінің A

оракулын сұратады. Тағы да, егер $\exists y\ \forall z\ \neg R^A(x, y, z)$ болса, онда $L(M) = \Sigma^*$; десек те басқа жағынан, егер $\exists y\ \forall z\ R^A(x, y, z)$ болса, онда M^A шектеулі жиын қабылдайды.

Енді біз Σ_n^0-толық болатын кез келген A үшін кез келген Σ_{n+2}^0 жиын

$$\{x \mid \exists y_1\ \forall y_2\ \exists y_3\ \cdots\ \mathrm{Q} y_{n+2}\ S(x, y_1,\ \ldots\ , y_{n+2})\},$$

A-дағы белгілі бір рекурсивті R үшін

$$\{x \mid \exists y\ \forall z\ R^A(x, y, z)\}$$

түрінде жазылатынын дәлелдеуіміз қажет. Бірақ A оракул Σ_n^0-күрделі болғандықтан, Σ_n^0 жиын

$$\{(x, y_1, y_2) \mid \exists y_3 \cdots\ \mathrm{Q} y_{n+2}\ S(x, y_1, \ldots\ , y_{n+2})\}$$

A-да рекурсивті болатынын пайдалансақ, дәлелдемені бірден аламыз.

Келесі проблемалар Σ_3^0 үшін $\leq_m$-күрделі болып табылады:

- $\mathrm{COF} \stackrel{def}{=} \{M \mid L(M)$ ко-шектеулі $\} = \{M \mid \sim L(M)$ шектеулі $\}$
- $\mathrm{REC} \stackrel{def}{=} \{M \mid L(M)$ рекурсивті $\}$,
- $\mathrm{REG} \stackrel{def}{=} \{M \mid L(M)$ регулярлы $\}$,
- $\mathrm{CFL} \stackrel{def}{=} \{M \mid L(M)$ контексті тәуелсіз $\}$.

Бұл проблемаларды анықтайтын предикаттарды керекті формада өрнектеу арқылы мәселелердің барлығы Σ_3^0-де жататындығын дәлелдеуге болады. Мысалы,

$$\begin{aligned}\mathrm{COF} &= \left\{M \mid \exists y\ \forall z\ |z| \geq |y| \ \rightarrow \exists t\ M(z)\downarrow^{t}\right\} \\ &= \left\{M \mid \exists y\ \forall z\ \exists t\ |z| < |y| \ \vee\ M(z)\downarrow^{t}\right\}.\end{aligned}$$

Σ_3^0 үшін 36.1 леммада баяндалған Σ_3^0-күрделі болатын $\mathrm{FIN}^{\mathrm{HP}}$-ның мәліметтерін пайдалана отырып, COF-тың $\leq_{\mathrm{m}}$-күрделі болатынын көрсетеміз. Біз

$$\left\{M \mid L\left(M^{\mathrm{HP}}\right) \text{ шектеулі }\right\} \leq_{\mathrm{m}} \ \left\{M \mid L(M) \text{ ко-шектеулі }\right\}$$

мәліметін құрғымыз келеді; басқаша айтсақ барлық M үшін $N = \sigma\left(M\right)$ машина болатын және $L\left(M^{\mathrm{HP}}\right)$ шектеулі $\Leftrightarrow$ $L(N)$ ко-шектеулі болатын жалпы σ рекурциясын құрғымыз келеді.

M-ді ескере отырып, HP оракулды K^{HP} оракул машина болсын, ол енгізу y-те келесі амалдарды іске асырады:

(i) M^{HP}-да қабылдау-ашық y-тен ұзындығы артық болатын z-ті табуға тырысады. Ол белгілі бір ретпен $|z| > |y|$ болатындай етіп, барлық z енгізулерде M^{HP}-ның уақытын бөлшектеуді симуляция жасау арқылы іске асырылады. Ең соңында, барлық $M^{\mathrm{HP}}(z)$ әртүрлі көптеген қадамдар арқылы симуляция жасалғанына көз жеткізу үшін есептеу уақыты бөлшектенеді. Егер осындай z бар болса, онда оны симуляция өзі табады. Егер жоқ болса, онда симуляция циклмен есептей береді. Қажетті z табылысымен симуляция процесі тоқтайды және машина (ii) қадамға көшеді.

(ii) Енгізу ω-да барлық жолдар $H\#\omega$ үшін H симуляциясын пайдаланып, ермек үшін (i) қадамдағы оракулдың «ия» деген барлық жауабын тексереміз. Мұндағы жолдар үшін оракулдан сұратым жасалған және оған «ия» деген жауап алынған. «Жоқ» деген жауап ескерілмейді. Оракул «ия» деп жауап берсе, онда барлық симуляция тоқтатылады, сондықтан енгізу ω-да H тоқтайды.

Құрастыру бойынша,

- егер $L(M^{\mathrm{HP}})$ шектеулі болса, онда $L(K^{\mathrm{HP}})$-да шектеулі; және
- егер $L(M^{\mathrm{HP}})$ шексіз болса, онда $L(K^{\mathrm{HP}}) = \Sigma^{*}$.

Белгілі бір енгізуде K^{HP}-ның есептеу тарихын қабылдамайтын, барлық жолдарды қабылдайтын машина N болсын (Біз есептеудің тарихын 23-ші лекцияда қарастырғанбыз). Қалыптасқан тәртіп бойынша, берілген машинаның есептеу тарихын көрсету үшін, жол төмендегі қасиеттерге ие болуы қажет.

1. Жол машина конфигурациясының тізбесін кодтайды.

2. Бірінші конфигурация белгілі бір енгізудің алғашқы конфигурациясы болады.

3. Соңғы конфигурация қабылдаушы конфигурация болып табылады.

4. Машинаның алмасу ережесі бойынша, $(i+1)$-ші конфигурация i -ші конфигурациядан алынады.

Мұндағы жалғыз айырмашылық оракул HP үшін қарастыратындығымыз. Біздің ұйғарымыздағы K^{HP}-ның есептеу тарихы болып табылатын жол, оракул сұратыммен және оған сәйкес оракул жауаптармен көмілген. Сондықтан есептеу тарихын анықтауға келесі шартты қосуымыз керек.

5. Жолда көрсетілген оракул жауаптары корректі болуы қажет.

Жол K^{HP}-тың қабылдау-ашық емес есептеу тарихы екендігін тексеру үшін машина N 1-5 шарттарының ішінен жоқ, дегенде біреуі орындалмайтынын тексеруі қажет. 1-4 шарттарды тексеру қиындық туғызбайды; бірақ N -де оракулге қабылдау-жабық, сонда ол оракул жауабының нақты екендігін қалай тексере алады?

Бұған жауап берейік: N машинасы оракулдің тек теріс жауаптарын ғана тексере алуы қажет. Оң жауаптар K^{HP}-дің өзімен тексерілген және есептеу тарихындағы оракул жауаптарының нақты болатыны дәлелденген; ол жоғарыда айтылған (ii) қадамның мақсаты болатын!

Сонымен, есептеу тарихы болмайтын жолдарға қабылдау-ашық болуы үшін N машинасы алдымен 1-4 шарттардың біреуі бұзыла ма, соны тексереді. Егер осылай болса, онда машинада қабылдау-ашық. Егер олай болмаса, онда жолда көрсетілгендей оракул жауабының дұрыстығы тексеріледі. Оң жауаптар үшін ол есептеу тарихының өзінен алып, дәлелдеуді тексере алады. Әрбір теріс жауап үшін, айталық $H\#\omega$ сұратымы үшін, H тоқтап қала соны білу үшін N машинасы енгізу ω да H-ты іске қосады. Егер осылай болса, онда ол (N)тоқтайды және қабылдау-ашық, өйткені есептеу тарихында көрсетілген $H\#\omega$ -ның сұратымына оракулдың берген «жоқ» деген жауабы дұрыс емес, сонымен 5-ші шарт бұзылды. N машина осыны, барлық осындай оракул сұратымдар $H\#\omega$ үшін уақытты бөліктеу образында жасайды.

Жиын $L(K^{\text{НР}})$ шектеулі, сонда тек сонда $K^{\text{НР}}$ есептеу тарихындағы қабылдау-ашық жиыны шектеулі болса. Өйткені есептеу тарихындағы қабылдау-ашық жиыны сияқты, қабылдау-ашық жолдар да көп болады; және егер тек егер $L(N)$ ко-шектеулі болса, онда осы жағдай іске асады. Сонымен біз құрастырған N машина, сонда тек сонда шектеулі жиын қабылдайды, егер $L(M^{\text{НР}})$ шектеулі болса.

Біршама күрделі аргументті пайдалана отырып, осы әдіспен REC, REG және CFL мәселелері Σ_3^0-күрделі болатынын дәлелдеуге болады (133- аралас жаттығу).

37-лекция

Пост мәселесі

Рекурсивті функциялар теориясының алғашқы мақсаттарының бірі р.с. жиынның m және Т - дәрежелерін түсіну болатын. A жиынының m-дәрежесі деп бірдің бірге келтірімділігіне $\leq_{\mathrm{m}}$ қатысты A-ға эквивалентті жиынды түсінеміз, ал A жиынының Т - дәрежесі немесе Тьюринг дәрежесі деп Тьюринг келтірімділігіне $\leq_{\mathrm{T}}$ қатысты A-ға эквивалентті жиынды түсінеміз. Эквивалентті жиындар бірдей есептеу ақпаратын қамтитын болғандықтан, эквивалентті кластарды қарастыру себебі осында жатыр, сондықтан есептеуде оларды идентификация жасауға болады.

Р.с. Т-дәреженің жоқ дегенде екі түрі бар, атап айтсақ рекурсивті жиындар дәрежесі (яғни, $\varnothing$-ның дәрежесі) және р.с.-толық жиындар дәрежесі (яғни, тоқтау мәселесінің дәрежесі). Әрбір m-дәреже Т -дәрежеде жататындықтан, р.с. m-дәреженің жоқ дегенде әртүрлі екі түрі бар. Эмиль Пост 1944 [96], екіден көп р.с. m -дәреже болатынын көрсетті, және осы проблеманы Т-дәреже үшін де қойып шықты. Бұл Пост мәселесі деген атпен белгілі болды. Мәселе 1956 жылға дейін 12 жыл шешілмей тұрды, тек Фридберг [45] және Мучник [88] бір-бірінен тәуелсіз түрде проблеманы шешіп шықты.

Фридберг-Мучник теоремасы рекурсивті функция теориясының классикалық нәтижесі болып табылады. Бұл нәтиженің дәлелдемесін біз 38-лекцияда баяндайтын боламыз. Дәлелдеме шектеулі зақымдалған приоритетті аргумент деп аталатын техниканы иллюстрация жасайды, ол техника басқа да қолданыста жақсы көмектеседі. Осы лекцияның басқа жақтарында m-дәреже проблемасын шешетін, Пост теоремасының дәлелдемесін көрсететін боламыз. Бұл сұрақты толық білгіңіз келсе, [104, 114] жұмыстарды қарасаңыз болады.

Осы және келесі лекцияларда біз 33 және 34-лекцияларда енгізілген рекурсивті функция теориясының стандартты нотациясына қайтып келеміз, бірақ ол 35 және 36-лекциялардағы белгілеулерден біршама өзгешелеу болады. Сонымен дербес рекурсивті функциялардың $\varphi_0, \varphi_1, \ldots$ Гёдел нөмірлеуі болсын. Рекурсивті функция φ_x үшін, егер φ_x y-пен анықталса, онда $\varphi_x(y)\downarrow$ деп жазамыз, және

$$W_x \stackrel{def}{=} \text{анықталу аймағы } \varphi_x = \{y \mid \varphi_x(y)\downarrow\},$$

деп анықтаймыз.

Әрбір жиын W_x р.с. жиын болып табылады және белгілі бір x үшін әрбір р.с. жиын W_x жиын болып табылады. Сондықтан $W_0, W_1, \ldots$ тізбесін р.с. жиындар индексациясы деп қарастыруымызға болады. Анықтаймыз

$$K \stackrel{def}{=} \{x \mid \varphi_x(x)\downarrow\} = \{x \mid x \in W_x\}.$$

K жиынның р.с.-толық болатындығын жеңіл көрсетуге болады: $\lambda y.\mathrm{comp}(x, \mathrm{const}(y))$ түрлендіруі арқылы $W_x \leq_{\mathrm{m}} K$ алынады, ал К-ның өзі W_k болады, мұндағы $k = \mathrm{comp}(u, \mathrm{pair}(i,i))$, және u мен i сәйкес универсалды және тепе-тең функция индекстері.

T - және m - дәрежелер

Келтірімділік қатынастар $\leq_{\mathrm{m}}$ және $\leq_{\mathrm{T}}$ анықтамасын еске түсірейік: егер барлық x үшін

$$x \in A \quad \Leftrightarrow \quad \sigma(x) \in B,$$

орындалатын жалпы рекурсивті функция σ табылса, онда $A, B \subseteq \omega$ үшін $A \leq_{\mathrm{m}} B$ болады және егер A жиыны B-да рекурсивті болса, онда $A \leq_{\mathrm{T}} B$ болады; яғни, егер M^B тоталды және $A = L(M^B)$ болатын B оракулді M Тюринг машинасы табылады деп қысқаша айтса болады. Егер $A \leq_{\mathrm{m}} B$ болса, онда $A \leq_{\mathrm{T}} B$ болады, өйткені біз енгізу x-те $\sigma(x)$-ты есептейтін және оракулды сұрататын M-ді құрастыра аламыз. Десек те, кері тұжырым орындалмайды: $\sim K \leq_{\mathrm{T}} K$ болғанымен, $\sim K \leq_{\mathrm{m}} K$ болмайды.

Егер $A \leq_{\mathrm{m}} B$ және $B \leq_{\mathrm{m}} A$ болса, онда $A \equiv_{\mathrm{m}} B$ деп анықтаймыз және $A \leq_{\mathrm{T}} B$ және $B \leq_{\mathrm{T}} A$ орындалса, онда $A \equiv_{\mathrm{T}} B$ деп анықтаймыз. A үшін анықталған $\equiv_{\mathrm{m}}$-эквиваленттілік класы және $\equiv_{\mathrm{T}}$-эквиваленттілік класы, A-ның сәйкес *m*-дәрежесі және *T*-дәрежесі деп аталады. Қатынас $\equiv_{\mathrm{m}}$ арқылы $\equiv_{\mathrm{T}}$ қатынасы айқындала түседі; басқаша айтсақ, кез келген A үшін A-ның *m*-дәрежесі A-ның *T*-дәрежесінің жақсартылған жиынында (setwise) жатады.

Барлық рекурсивті жиыннан құрастырылған *m*-дәрежелер $\leq_{\mathrm{m}}$ -төменгі деп аталады. *m*-дәрежелі *K*-дан құрастырылған р.с.-толық жbындар тобы *m*-дәрежелі р.с. $\leq_{\mathrm{m}}$-жоғарғы деп аталады. Пост 1944 жылы осы екеуінен басқа да *m*-дәрежелер бар екендігін дәлелдеді және осы жай Т-дәрежелер үшін де дұрыс болуы мүмкін деген ұйғарым айтты.

37.1-теорема (Пост 1944 [96]) *Р.с.-толық болмайтын бейрекурсивті р.с.жыйындар табылады.*

Иммунды, қарапайым және өндіргіш жиындар

37.1-теореманың дәлелдемесі иммунды, қарапайым және өндіргіш жиындар түсінігін қажетсінеді.
37.2-анықтама Егер

- A шексіз*,* және
- A ешқандай ішкі шексіз р.с. жиынды қамтымайды деген шарттар

орындалса, онда $A \subseteq \omega$ жиыны иммунды деп аталады.

37.3-анықтама Егер

- B р.с. болса және
- $\sim B$ иммунды болса;

онда $B \subseteq \omega$ жиыны қарапайым деп аталады.

Басқаша айтсақ, егер

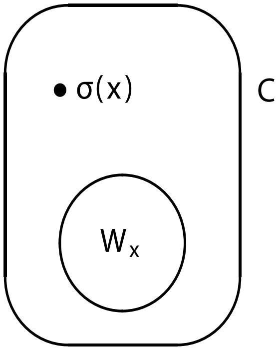
- B р.с. және
- $\sim B$ шексіз*,* және
- B әрбір *р.с. жиынмен қи*ылысады

онда $B \subseteq \omega$ жиыны қарапайым деп аталады.

37.4-анықтама Егер әрбір $W_x \subseteq C$ үшін жалпы рекурциялы функция σ табылып,

$$\sigma(x) \quad \in \quad C - W_x$$

орындалса, онда $C \subseteq \omega$ жиыны өндіргіш деп аталады. Ал функция σ C *-ның* өндіргіш функциясы деп аталады.



37.5-мысал. Мысалы, өндіргіш функциясы $\lambda x.x$ тепе-теңдік болатын, жиын $\sim K$ өндіргіш болады.

Осыны дәлелдеу үшін $W_x \subseteq \sim K$ болады деп ұйғарайық. Кез келген x үшін,

W_x -тың анықтамасы бойынша $x \in W_x \quad \Leftrightarrow \quad \varphi_x(x)\downarrow$,

K -ның анықтамасы бойынша $\Leftrightarrow \quad x \in K$.

Сондықтан, екінің бірі орындалады, немесе

- $x \in W_x$ және $x \in K$, немесе
- $x \notin W_x$ және $x \notin K$.

Ұйғарым бойынша $W_x \subseteq \sim K$ болуы қажет еді, яғни мүмкін емес алғашқы шарт алынған; сондықтан $x \in \sim K - W_x$.

Пост теоремасының дәлелдемесі

Жоғарыда енгізілген түсініктер мен анықтамаларды ескере отырып, 37.1-теореманың дәлелдемесін негізгі үш леммаға бөлуге болады.

37.6-лемма Қарапайым жиын табылады.

37.7-лемма Егер B қарапайым болса, онда $\sim B$ өндіргіш емес.

37.8-лемма Егер A р.с.-толық болса, онда $\sim A$ өндіргіш болады.

Төменде осы леммаларды дәлелдейтін боламыз. Алдымен, осыларды пайдаланып, Пост теоремасын қалай дәлелдеуге болатынын қарастырамыз.

37.1-теореманың дәлелдемесі. 37.6-леммада бойынша табылатын B қарапайым жиын болсын. B жиыны рекурсивті бола алмайды, олай бола қалса $\sim B$ р.с. болуы қажет; бұл жағдай B қарапайым деген шартқа қайшы келеді. Өйткені, $\sim B$ шексіз және B барлық шексіз р.с. жиындармен қиылысуы қажет. Сонда, 37.7 және 37.8-леммалар бойынша B р.с.-толық болмауы қажет.

37.6-лемманың дәлелдемесі. Біз қарапайым B жиын құрастырамыз. Конструкция қалыптасуы үшін бізге үш шарттың орындалғаны қажет болады:

- B р.с. болуы қажет,
- $\sim B$ шексіз болуы қажет және
- B әрбір шексіз р.с. жиынмен қиылысуы қажет.

B үшін тізбелеу процедурасын сипаттаймыз. W_x р.с. жиынның тізбелейтін тізбелеу машинасы M_x болсын. Тізбелеу машинасының оқу/жазу таспалары бар және жазуға мүмкіндігі бар шығару таспасы бар, бірақ енгізу таспасы жоқ екендігін еске түсірейік ([76] қараңыз). Ол өз жұмысын бос таспалар жасаумен бастайды және шексіз жұмыс істей береді. Кейде ол ерекше тізбелеу күйге енеді, сол сәтте шығу таспасына жазылған жол тізбеленген деп аталады және енгізу таспасы бірден өшіріледі де бас тиек таспаның басына орналасады.

Біздің B-ға жасайтын тізбелеу процедурамыз барлық тізбелеу машиналары үшін уақытты бөліктеу симуляциясын жасайды. Тізбелеу ағымдағы уақыттағы симуляциядағы машиналар және таспаның әртүрлі блогындағы әртүрлі машиналарды симуляциялайды. Ол симуляцияны бір қадамда M_0-ден бастайды, одан кейін M_0 мен M_1 әрқайсысына бір қадам, одан кейін M_0, M_1, және M_2 әрқайсысына бір қадам және осылай кете береді. Әрбір симуляция раундында, тізбелеу тізімге жаңа машинаны қосады және жаңа машинаны симуляциялау үшін өзінің жұмыс таспасынан блок бөледі. Егер белгілі бір сәтте бір симуляция үшін орын жетпей қалса, онда жаңа кеңістік құрастыру мақсатында блоктарды жылжыту үшін ол ішкі программаға енеді.

Белгілі сәтте, егер M_x-ті симмуляциялау алғашында белгілі бір $(y \geq 2x)$-ті тізбелегісі келсе, онда біздің процедурамыз y-ті тізбелейді, осыдан кейін M_x-ті симуляциялауды аяқтайды және оны симуляцияланатын машиналар тізімінен алып тастайды. $g(x)$-ты M_x-пен тізбеленген y бол деп белгілейміз, ол осыны болдырғысы келеді. M_x машинасы кез келген $(y \geq 2x)$-ті ешуақытта тізбелей алмай қалуы мүмкін, және бұл жағдайда M_x-ті симуляция жасау еш уақытта тоқтамайды және $g(x)$ анықталмай қалып қояды.

Осы процедурамен барлық уақытта тізбеленген элементтер B жиынын құрасын. Біз B қарапайым деп ұйғарамыз. Біріншіден, біз жаңа ғана ол үшін тізбелеу процедурасын көрсеткендіктен, B р.с. болады. Екіншіден, оның толықтауышы шексіз болады, өйткені $2n$-элементті жиын $\{0,\dots,2n-1\}$ B-ның n элементінен артық элемент қамти алмайды, атап айтсақ $g(0),\dots,g(n-1)$. $g(m) \geq 2m$ болғандықтан, басқа ешқандай элемент $g(m)$ бұл жиында бола алмайды. Және соңында, белгілі бір x үшін осындай кез келген жиын W_x болатындықтан, B әрбір шексіз р.с. мен қиылысады. M_x белгілі бір $y \geq 2x$ үшін өзіне жүктелген тізбелеуді жасаған кезде, W_x шексіз болғандықтан y B-ның элементі ретінде тізбеленеді.

37.7-лемманың дәлелдемесі. Кез келген өндірістік жиын $\varnothing$ бастап өндіргіш функциямен итерация жасау арқылы алынған, шексіз р.с. ішкі жиынды қамтиды. Өндіргіш функциясы σ болатын өндіргіш жиын C болсын. Толығымен анықталмаған функция индексі i_0 болсын; сонымен

$$W_{i_0} = \varnothing \subseteq C.$$

Біз $W_{i_0} \subseteq C$ жиынын құрастырып алдық деп ұйғарайық. Сонда $\sigma(i_n) \in C - W_{i_n}$. Анықтаймыз

$$W_{i_{n+1}} \overset{def}{=} W_{i_n} \cup \{\sigma(i_n)\}.$$

Сонда $W_{i_{n+1}} \subseteq C$. Одан басқа, i_n индексінен i_{n+1} индексін тиімді жолмен ала аламыз. Сондықтан, жиын

$$\{\sigma(i_0), \sigma(i_1), \sigma(i_2), \ldots\}$$

C-ның шексіз р.с. ішкі жиыны болады.

37.8-лемма дәлелдемесі. A р.с. -толық деп ұйғарайық. Сонда жалпы рекурсиялық функция σ сүйенсек, $K \leq_{\mathrm{m}} A$ болады. Сондықтан барлық x үшін,

$$x \in K \iff \sigma(x) \in A;$$

эквивалентті түрде,

$$\sim K = \sigma^{-1}(\sim A), \tag{37.1}$$

мұндағы $\sigma^{-1}(\sim A) \overset{def}{=} \{x \mid \sigma(x) \in \sim A\}$.

37.5-мысалды еске түсірейік, онда өндіргіш функция $\lambda x.x$ арқылы $\sim K$ өндіргіш деп едік. A үшін өндіргіш функция алу үшін осы факты мен σ келтірілімдігін біріктіреміз.

$W_i \subseteq \sim A$ деп ұйғарайық. σ-ның m индексі болсын және

$$\tau = \lambda i.\mathrm{comp}(i, m)$$

болсын. Сонда

$$
\begin{aligned}
W_{\tau(i)} &= W_{\mathrm{comp}(i,m)} \\
&= \{x \mid \varphi_{\mathrm{comp}(i,m)}(x)\downarrow\} \\
&= \{x \mid \varphi_i(\sigma(x))\downarrow\} \\
&= \{x \mid \sigma(x) \in W_i\} \\
&= \{\sigma^{-1}(W_i)\} \\
&\subseteq \{\sigma^{-1}(\sim A)\} \qquad \sigma^{-1}\text{-дің монотондығы бойынша} \\
&= \ \sim K \qquad \text{(37.1) бойынша.}
\end{aligned}
$$

Тепе-теңдік функция $\sim K$-ның өндіргіш функциясы болғандықтан,

$$\tau(i) \ = \ \sim K - W_{\tau(i)} \ = \ \sigma^{-1}(\sim A) - \sigma^{-1}(W_i) \ = \ \sigma^{-1}(\sim A - W_i);$$

сондықтан

$$\sigma(\tau(i)) \ \in \ \sim A - W_i.$$

Осының әсерінен $\sim A$-ның өндіргіш функциясы $\sigma \circ \tau$ болып табылады.

37.8-лемманы жалпылау үшін 112-аралас жаттығуды қараңыз.

38-лекция

Фридберг-Мучник теоремасы

Бұл лекцияда Фридберг және Мучниктің Пост мәселесін қалай шешті соған тоқталамыз. Қазіргі таңда рекурсивті функциялар теориясында жиі қолданылатын, шектеулі зақымдалған приоритетті аргумент деп аталатын өте пайдалы техниканы дәлелдемеде иллюстрация жасаймыз.

Дара рекурсивті функцияның $\varphi_0, \varphi_1, \ldots$ Гёдел нөмірлеуі болсын. Өткен жолғы белгілеулерді

$$W_n \overset{def}{=} \text{анықталу аймағы } \varphi_n = \{x \mid \varphi_n(x)\downarrow\}$$

$$K \overset{def}{=} \{n \mid \varphi_n(n)\downarrow\},$$

еске түсірейік , мұндағы $\downarrow$ “ анықталған” дегенді білдіреді, және

$$B \le_T C \quad \stackrel{def}{\Leftrightarrow} \quad B \quad C\text{-да рекурсивті.}$$

Анықтаймыз

$$B <_T C \quad \stackrel{def}{\Leftrightarrow} \quad B \le_T C \quad \text{бірақ } C \not\le_T B.$$

38.1-теорема (Фридберг [45] и Мучник [88]) $K \not\le_T A$ орындалатын, бейрекурсивті р.с. жиын A табылады. Басқаша айтсақ,

$$\varnothing \ <_T \ A \ <_T \ K$$

болатын A жиыны табылады.

Аласа жиын

38.1-теореманың дәлелдемесі [114]-тен алынады. Дәлелдеме аласа жиын концепциясын өзінде қамтиды.

38.2-анықтама. Егер A жиыны р.с. болса және $K^A \le_T K$ орындалса, онда ол аласа деп аталады.

Басқаша айтсақ, егер оракул A болғандағы шешімнің тоқтауы, оракул A болмағандағы шешімнің тоқтауынан күрделі болмаса, онда A аласа деп аталады.

38.3-лемма. Егер A аласа болса, онда $A <_T K$.

Дәлелдеу. A р.с. және K р.с.-толық болғандықтан, $A \le_T K$ болатыны талас тудырмайды. Енді, егер $K^K \le_T K^A$ болса, онда $K \le_T A$ болады. $K^A \le_T K$ аласалықты және $\le_T$ -ның транзитивтілігін пайдалансақ $K^K \le_T K$ болады. Бірақ ол мүмкін емес, өйткені Σ_2^0 үшін K^K толық және Σ_1^0 үшін K толық.

Өткен лекциядан еске түсірейік, егер

- A р.с. болса,
- $\sim A$ шексіз болса және
- A әрбір шексіз р.с. жиынмен қиылысса, онда A қарапайым болады.

38.4-лемма Қарапайым аласа жиын табылады.

38.4-лемманы төменде дәлелдейтін боламыз. Алдымен, Фридберг-Мучник теоремасы қалай дәлелденетінін көрсетуге рұқсат етіңіз.

38.1-теорема дәлелдемесі. 38.4-леммасының тұжырымы бойынша табылатын, A аласа қарапайым жиын болсын. 38.3-лемма бойынша $A <_{\mathrm{T}} K$ болады. Бірақ ешқандай қарапайым жиын рекурсивті бола алмайтындықтан, $\varnothing <_{\mathrm{T}} A$ болады.

Шектеулі зақымдалған приоритетті аргумент

Бұл жолы 38.4-лемманың дәлелдемесін беретін боламыз. Аласа қарапайым A жиын үшін оны шексіз көп шектеулі жиын бірігуі ретінде

$$A = \bigcup_{t \geq 0} A_t$$

өрнектеп, тізбелеу процедурасын қарастырамыз, мұндағы A_t-ның t элементі бар және $A_t \subseteq A_{t+1}$, $t \geq 0$.

A-ның аласалығы мен қарапайымдылығын қамтамасыз ету үшін біз бірнеше бәсекелес шарттарды қанағаттандыруымыз қажет. Элементті A-ға орналастырғысы келетін кейбір оң шарттар бар және элементті A-ға жолатқысы келмейтін теріс шарттар бар. Арасында, кейбір шартты орындау үшін, бұрын қанағаттандырылған басқа шартты талдауға тура келеді. Осы жолмен талданатын шартты зақымдалған деп атайды. Дегенмен біз шарттарға приоритет меншіктейміз, сондықтан әрбір шарт үшін, одан приоритеті артық болатын саны шектеулі шарттар табылады. Шарт өзінен приоритеті артық болатын шартпен ғана зақымдалуы мүмкін және ол тек бір рет осы шартпен зақымдалады. Сонымен шарт тек шектеулі сан рет зақымдалады және соңында қанағаттандырылатын болады.

A аласа және қарапайым болуы үшін төмендегі шарттарды қамтамасыз етуіміз қажет:

(i) A р.с. болады;

(ii) A ко-шектеулі болады;

(iii) A әрбір шексіз р.с. жиынмен қиылысады;

(iv) $K^A \leq_{\mathrm{T}} K$.

Біз A-ны тізбелеу үшін процедураны өзіміз береміз, сондықтан бірінші шарт (i) автоматты түрде ақиқат болады. Келесі (ii) шарт теріс шарт болып табылады, бірақ ол көп қиындық тудырмайды; ол шарт дәл Пост теоремасының (37.1-теорема) дәлелдемесіндегідей жолмен өңделетін болады.

Мұндағы (iii) және (iv) шарттар қызғылықты шарттар болып саналады. Әрбір n үшін екі түрлі шарт қарастырамыз, оның біреуі оң және екіншісі теріс:

P_n: Егер W_n шексіз болса, онда $A \cap W_n \neq \varnothing$.

N_n: Егер шексіз көп t үшін $\varphi_n^{A_t}(n) \downarrow^t$ болса, онда $\varphi_n^A(n) \downarrow$ болады.

Мұндағы таңба $\downarrow^t$ - « осы функцияны есептейтін машина t қадамда тоқтайды» деген мағына береді.

P_n шарттары оң шарттар болып табылады; егер олар барлық n үшін қанағаттандырылса, онда (iii)-ші шарт орындалады. Ал N_n шарттар теріс шарттарға жатады; егер олар барлық n үшін орындалса, онда (iv) орындалады (оны төменде қарастыратын боламыз). Біз шарттарға приоритеттер

$$P_0 > N_0 > P_1 > N_1 > P_2 > N_2 \cdots$$

меншіктейміз. Кез келген шартқа, приоритеті үлкен болатын бірнеше шарт табылатынына назар аударыңыз.

Бұл N_n-ның түсіндірмесі. $\varphi_n^{A_t}(n) \downarrow^t$ болады деп ұйғарайық. Соңғы стадияларда, осы есептеуде оракул сұратымына тап болған элементтерді, A-ның жаңа элементі ретінде орналастырғымыз келмейді. Егер осыны сәтті жеңе алсақ, онда бізде $\varphi_n^A(n) \downarrow$ бар болады. Өйткені A мен A_t сұратылатын элементтер жайлы келістіріледі. Дегенмен, барлық уақытта осы жағдаймен күресе алмаймыз. Сондықтан N_n зақымдалған болуы мүмкін екен. Бірақ біз, N_n-ді жоғары приоритетті P_k ғана зақымдайтын ету жағын қамтамасыз ете аламыз, және әрбір P_k үшін тек бір рет болатын етеміз. Сондықтан, шарт N_n егер болса тек шектеулі сан рет зақымдалатын болады.

Шарт N_n-ның конъюнкциясы аласалықты меңзейді. Осыған көз жеткізу үшін, егер N_n шарты орындалса, онда

$$\overset{\infty}{\exists} t\ \varphi_n^{A_t}(n)\downarrow^t \ \Rightarrow\ \varphi_n^{A}(n)\downarrow\ \Rightarrow\ \overset{\infty}{\forall} t\ \varphi_n^{A_t}(n)\downarrow^t\ \Rightarrow \exists t\ \varphi_n^{A_t}(n)\downarrow^t,$$

болатынын байқаймыз, мұндағы $\overset{\infty}{\exists}$ таңбасы «шексіз көп табылады» және $\overset{\infty}{\forall}$ таңбасы «барлық бірақ шексіз көп» деген ұғымдарды береді. Келтірілген қатынастағы алғашқы өрнек N_n-ді бейнелейді, ал қалған өрнектер жиынның базалы теориялық дәйектемесі. Сондықтан егер барлық n үшін N_n орындалса, онда

$$\begin{aligned} K^A &= \{n \mid \varphi_n^A(n)\downarrow\} = \left\{n \mid \overset{\infty}{\exists} t\ \varphi_n^{A_t}(n)\downarrow^t\right\} \\ &= \left\{n \mid \overset{\infty}{\forall} t\ \varphi_n^{A_t}(n)\downarrow^t\right\} = \{n \mid \forall k\ \exists t \geq k\ \varphi_n^{A_t}(n)\downarrow^t\} \\ &= \{n \mid \exists k\ \forall t \geq k\ \varphi_n^{A_t}(n)\downarrow^t\} \end{aligned} \tag{38.1}$$

болады. (38.1) квантификация формасынан

$$K^A \in \Sigma_2^0 \cap \Pi_2^0 = \Delta_2^0 = \{\mathrm{B} \mid \mathrm{B} \leq_{\mathrm{T}} K\},$$

болатынын көре аламыз, сондықтан $K^A \leq_{\mathrm{T}} K$ болып шығады, осы шарт (iv)-ші шарттың дәл өзі.

Енді, барлық қажетті шарттарды қанағаттандыратын, A-ның оқиғалы-бағытталған тізбелеулерін қарастырамыз. M_m машинасы W_m-ды тізбелейтіндей, тізбелейтін машиналар тізімі $M_0, M_1, M_2, \ldots$ болсын. $A_0 = \varnothing$ деп белгілейік.

37.6-лемманың дәлелдемесінде көрсетілгендей етіп, $M_0, M_1, \ldots$ үшін уақытты бөліктеудің параллельдік симуляциясын орындаймыз, мұнда ағымдағы уақыттағы симулациядағы машиналар тізімін сақтау міндетті. Белгілі бір машина M_m керекті элемент x-ті тізбелегенше, симуляция жасала береді. Осы орындала салысымен, симуляцияны тоқтатамыз және келесі амалдарды жасаймыз. A_t-ны құрастырғандай, t элементті A-ға орналастырдық деп ұйғарайық.

(a) Егер $x < 2m$ болса, онда симуляцияны қайта қосамыз.
(b) Кері жағдайда, барлық $n < m$ үшін, t қадамға $\varphi_n^{A_t}(n)$-ді іске

қосамыз. Кез келген n-де тоқтау болған кезде, егер осы есептеуде x A_t -мен сұратылған болса, онда симуляцияны іске қосамыз.

(c) Кері жағдайда, $x \in A$ орналастырамыз (яғни, $A_{t+1} := A_t \cup \{x\}$ тағайындаймыз) және M_m -ді тізімнен сызып тастаймыз.

Енді біз керекті шарттар қанағаттандырылды деп ұйғарамыз. Алдымен (i) қанағаттандырылды. 37.6-теоремадағы сияқты, (a) амалының әсерінен

$$|A \cap \{0,1,...,2m\text{-}1\}| \leq m$$

болғандықтан, (ii) қанағаттандырылады.

Енді P_n және N_n шарттары қанағаттандырылғанын көрсетеміз. Әрбір n үшін (c) бойынша әйтеуір тізімнен сызылуы керек, әрбір M_m, $m < n$ сызылып кеткен уақыт сәті болады. Осы сәттен кейін, егер $\varphi_n^{A_t}(n)\downarrow^t$ орындалатын болса, онда барлық $s \geq t$ үшін $\varphi_n^{A_s}(n)\downarrow^t$ болады. Сондықтан (b) -ға сәйкес, оракул басқа бірдеме жасауға мәжбүр болмау үшін оракулға ешқандай өзгеріс енгізуге болмайды. Сондықтан $\varphi_n^{A}(n)\downarrow^t$, осының әсерінен N_n қанағаттандырылады.

P_n шарты да қанағаттандырылған: егер W_n шексіз болса, онда $2n$-нен үлкен элемент x-ті M_n әйтеуір тізбелейтін болады. Және бұдан басқа, ол оракулдің кез келген сұратымынан үлкен болады. Ол сұратым тоқтайтын жоғары приритетті есептеу $\varphi_k^{A_t}(k)$ -мен жасалады. Осы сәтте x A -ның элементі болып бекиді.

39-лекция

Аналитикалық иерархия

Арифметикалық иерархия бірінші ретті сандар теориясына жатады, ал аналитикалық иерархия екінші ретті сандар теориясына жатады, мұнда жиындар және функциялармен квантация жасауға жол ашылады. Бізді алдымен бұл иерархияның бірінші деңгейі қатты қызықтырады, жеке жағдайда $\mathbb{N}$ үшін екінші ретті жалғыз квантормен анықталатын Π_1^1 қатынастар класы қызықтырады. Бірінші ретті индукциямен $\mathbb{N}$ -де анықталатын қатынастар класы, бізді қызықтыратын класс деп Клиннің тамаша теоремасы тұжырымдайды. Осы және келесі лекцияда Π_1^1 және Δ_1^1 кластарының есептеуінің сипаттамасын көрсетеміз және Клин теоремасының дәлелдемесінің нобайын жасаймыз.

Π_1^1 -ді анықтау

Универсалды екінші ретті теориялы сандық формуламен анықталатын, $\mathbb{N}$-дегі барлық қатынастар класы Π_1^1 класы болып табылады. Бұл жерде $f:\mathbb{N}\to\mathbb{N}$ функцияларда универсал квантация ($\forall$) қолданылатындықтан, «универсалды екінші ретті» деп айтылды. Бірінші ретті квантификация шектеусіз. Әртүрлі түрлендірулер ережесін пайдаланып, соның ішінде жұптап біріктіру және сколемдеу[1] де бар, әрбір осындай формула

$$\forall f\ \exists y\ \varphi\left(\overline{x},y,f\right), \tag{39.1}$$

түрінде жазылады деп ұйғаруға болады, мұндағы φ квантордар тәуелсіз (141-аралас жаттығу). Бұл формулалар n-арлы қатынасты

$$\left\{\overline{a}\in\mathbb{N}^n \mid \forall f\ \exists y\ \varphi\left(\overline{a},y,f\right)\right\}$$

анықтайды.

Индуктивті анықтау және IND программалау тілі

Дәстүрлі түрде, бірінші ретті индуктивті қатынас структурада бірінші ретті формуламен анықталған, монотонды бейнелеудің ең кіші нүктелері арқылы анықталады. Мысалы, жиындағы R бинарлық қатынастың R^* транзитивті тұйықталуы, монотонды бейненің

$$X \ \mapsto\ \left\{(a,c)\mid a=c\vee\left(\exists b(a,b)\in R\wedge(b,c)\in X\right)\right\} \tag{39.2}$$

ең кіші нүктелері болады (А лекцияны қараңыз). Бірінші ретті индуктивті анықтаудың теориясы өте жақсы қойылған; мысал үшін [87]-ні қараңыз.

Осы тақырыпқа деген біздің ұстаным есептеуге негізделген. Біз IND программалау тілін көрсетеміз және оны индуктивті және гипер қарапайым бейне және рекурсивті ординалдарды анықтауда пайдаланамыз. Төменде тұжырымдалатындай біздің ұстаным дәстүрлі ұстанымға эквивалентті болып шығады. Дегенмен, IND программасымен «есептеу» бейнесі қатаң есептелмейді. IND программа Харел және Козен [53] еңбектерінде анықталған ([54]-ті де қараңыз).

[1] $\exists x:\mathbb{N}\ \to\forall f:\mathbb{N}\to\mathbb{N}\ \varphi\left(f,x\right)\ \mapsto\ \forall \mathsf{g}:\mathbb{N}\to\left(\mathbb{N}\to\mathbb{N}\right)\ \exists x:\mathbb{N}\ \varphi\left(\mathsf{g}(x),x\right).$

IND программа белгіленген шектеулі ұйғарымдар тізбесінен тұрады. Әрбір тұжырым келесі үш форманың тек біреуін қабылдайды:

- Меншіктеу: $\ell:\ x := \exists$ $\ell:\ y := \forall$
- Шартты тексеру: $\ell:$ егер $R(t_1,\dots,t_n)$ болса, онда $goto\ \ell'$
- Тоқтау тұжырымы: $\ell:$ қабылдау-ашық $\ell:$ қабылдау-жабық

Программа семантикасы Тьюринг алтернативті машинасының семантикасына өте ұқсас, ал айырмашылық тармақталудың шексіз болуында ғана. Меншіктеу амалы саналымды санды ішкі процестерді туындатады, оның әрқайсысы айнымалыға $\mathbb{N}$-нен әртүрлі элемент меншіктейді. Егер тұжырым $x := \exists$ болса, онда тармақталу экзистенциалды болады; егер тұжырым $y := \forall$ болса, онда тармақталу универсиалды болады. Шартты ауысу атомдық формуланы $R(t_1,\dots,t_n)$ тексереді және егер ол ақиқат болса, онда көрсетілген белгіге (метка) ауысады. Қабылдау-ашық және қабылдау-жабық командалар тоқтатады және логикалық мәнді кері қарай атасына дейін жібереді.

Есептеу алтернативті Тюринг машинасындағыдай жүргізіледі. Енгізу программаның алғашқы меншіктеуі болады. Орындау саналатын төмен жайыла орналасқан бұтақтың пайда болуының себепкері және логикалық мән қабылдау-ашық (1) немесе қабылдау-жабық (0) бұтақты жоғары қуалай кері бағытта беріледі. Және бұл орындалу, әрбір экзистенционалды түйінде логикалық $\wedge$ амалын, ал әрбір универсиалды түйінде $\wedge$ амалын есептеуге шақырады. Егер есептеу бұтақ түбірі бір уақыттан кейін логикалық мән 1-мен белгіленсе, онда біз енгізуге программада қабылдау-ашық деп айтамыз; ал егер осы енгізуде бір уақыттан кейін түбір логикалық мән 0-мен белгіленсе, онда осы енгізуге программада қабылдау-жабық деп айтамыз. Және де, егер программада осы енгізуге не қабылдау-ашық, не қабылдау-жабық болса, онда программа енгізуде тоқтайды дейміз. Барлық енгізуде тоқтайтын IND программаны толалды деп айтамыз.

Бұл түсініктер толығымен Тьюринг алтернативті машиналары сияқты, сондықтан формалдау процесіне назар аудармай кейбір көрсетпе мысалдарға орын береміз.

Біріншіден, жоғарыда келтірілген түсініктерді қолдана отырып, бірнеше пайдалы программалау конструкцияларын қалай симуляция жасауға болатынын көрсетеміз. Шартсыз ауысу

goto ℓ',

$$\text{if } x = x \text{ then goto } \ell$$

деген тұжырыммен симуляцияланады.

Бұдан күрделілеу шартты тармақтау кездессе, ол басқару ағынын манипуляциялау арқылы іске асырылады. Мысалы, келесі тұжырымды

$$\text{if } R(\overline{t}) \text{ then reject else } \ell$$

программа бөлігі арқылы симуляцияланады

$$\text{if } R(\overline{t}) \text{ then goto } \ell'$$

$$\text{goto } \ell$$

$$\ell': \quad \text{reject}.$$

Жай меншіктеу

$$x := y + 1$$

оймен табу және тексеру арқылы іске асырылады, оның симуляциясы

$$x := \exists$$

$$\text{if } x \neq y + 1 \text{ then reject}.$$

Бұл процесс шексіз көп ішкі процестерді туындатады, оның тек басқалардан өзгеше бір жері бар, ол жерде бірден қабылдау-жабық болады!

Бірінші ретті кез келген бейне цикл жоқ программамен анықталады. Мысалы,

$$\exists y\ \forall z\ \exists w\ x \leq y \wedge x + z \leq w$$

қатынасы орындалатын x натурал сандар жиыны

$$\begin{aligned}&y := \exists\\&z := \forall\\&w := \exists\\&\text{if } x > y \text{ then reject}\end{aligned}$$

```
if x + z ≤ w then accept
reject ,
```

программасымен анықталады. Кері жағдай да орындалады: цикл жоқ кез келген программа бірінші ретті қатынасты анықтайды.

Десек те, индуктивті анықталатын бірінші ретті емес қатынасты да IND арқылы анықтауға болады. Мысалы, қатынас R-дің рефлексивті транзитивті тұйықталуы R^*-ны төменде келтірілген программамен анықтауға болады. Ол программа өзінің енгізуін x, z айнымалылары арқылы алады және егер $(x,z) \in R^*$ болса, онда оларға қабылдау-ашық. Сонымен

```
ℓ :   if x = z then accept
        y := ∃
      if ¬R(x, y) then reject
        x := y
        goto ℓ .
```

Осы программаны (39.2) -мен салыстырыңыз.

Тағы бір мысал. 8 лекцияда баяндалған екі адам үшін идеалды (таза) ақпаратты ойын логикалық предикат MOVE-тан тұратынын еске түсірейік. Екі ойыншы алма-кезек ауысып жүріп отырады. Егер тақтаның ағымдағы конструкциясы x болса және жүріс I ойыншыда болса, онда ол ойыншы $\mathrm{MOVE}(x, y)$ болатын y-ті таңдайды; осыдан кейін II ойыншы $\mathrm{MOVE}(y, z)$ болатын z-ті таңдайды; тағы сол сияқты кете береді. Ойыншы матпен жеңеді; яғни қарсыласты жүріс таба алмайтын жағдайға түсіруі қажет. Сондықтан, егер $\forall z\ \neg\mathrm{MOVE}(y,z)$ болатын y табылса, онда ол жағдай мат деп аталады.

Тақтада берілген x жағдайы үшін жүріс кезегін алған ойыншы x жағдайында жедел жеңіске жете ала ма, соны білгіміз келеді. Ол үшін, 8-лекциядағы сияқты, рекурсивті теңдеудің

$$\mathrm{WIN}(x) \iff \exists y\ \left(\mathrm{MOVE}(x, y) \wedge\ \forall z\ \mathrm{MOVE}(y, z)\ \rightarrow\ \mathrm{WIN}(z)\right)$$

ең кіші түбірі WIN-ді іздеу арқылы мақсатқа жете аламыз. (Базалық вариант қамтитын жағдай, бірден мат қою арқылы жеңіске жету: егер y маттық жағдай болса, онда ішкі формула $\forall z\ \mathrm{MOVE}(y, z)\ \rightarrow\ \mathrm{WIN}(z)$ мәнсіз ақиқат болады). Осы рекурсивті теңдеудің ең кіші шешімі,

$$\tau(R) \overset{\text{def}}{\Leftrightarrow} \{x \mid \exists y \ \mathrm{MOVE}(x,y) \wedge \ \forall z \ \mathrm{MOVE}(y,z) \ \to \ R(z)\}$$

формуласымен анықталатын монотонды бейне τ-дың ең кіші қозғалыссыз нүктесі болады. Біз WIN(x)-ді IND программа арқылы келесі түрде түрлендіре аламыз:

$$\begin{array}{ll} \ell: & y := \exists \\ & \text{if } \neg\mathrm{MOVE}(x,y) \text{ then reject} \\ & x := \forall \\ & \text{if } \neg R(y,x) \text{ then accept} \\ & \text{goto } \ell. \end{array}$$

Соңғы мысал фундирлі қатынасты пайдаланады. Индукция және фундирлеу қатар жүріп отыратыны бізге белгілі (23-аралас жаттығу). Төменде келтірілген IND программада қатаң дербес реттілік $<$ фундирлі бола ма соны тексереді:

$$\begin{array}{ll} & x := \forall \\ \ell: & y := \exists \\ & \text{if } \neg(y < x) \text{ then accept} \\ & x := y \\ & \text{goto } \ell. \end{array}$$

Бірінші ретті оң формуламен анықталған монотонды бейненің ең кіші қозғалмайтын нүктесі арқылы өрнектелетін кез келген қасиетті, IND программа арқылы есептеуге болады. Бұл жерде біздің нені айтқымыз келгені айтылады. Сонымен тәуелсіз айнымалылы $\overline{x} = x_1,\ldots,x_n$ және тәуелсіз n-арлы бейне R-ден тәуелді $\varphi(\overline{x},R)$ бірінші ретті формула болсын. φ-дегі R-дің барлық енулері оң болсын деп ұйғарайық; яғни теріске шығару $\neg$ символы жұп рет кездеседі дейміз. Кез келген n-арлы қатынас B үшін анықтаймыз

$$\tau(B) \overset{\text{def}}{=} \{\overline{a} \mid \varphi(\overline{a},B)\}.$$

Яғни φ-ді τ жиындық операторының қатынасы деп түсінеміз, ол жиындар кортежы B-ны басқа жиындар кортежіне бейнелейді $\{\overline{a} \mid \varphi(\overline{a},B)\}$. Оң болсын деген ұйғарым жиындық оператор τ-дың монотондылығын көрсететінін дәлелдеуге болады, сондықтан А.9

теорамасы бойынша n-арлы бейнелі ең кіші қозғалмайтын нүкте F_φ табылады. Қалыптасып кеткен жол бірінші ретті индуктивті бейнені, осындай қозғалмайтын нүктенің проекциясы ретінде анықтайды; яғни,

$$\left\{\left(a_1,\ldots,a_m\right) \mid F_\varphi\left(a_1,\ldots,a_m,b_{m+1},\ldots,b_n\right)\right\},$$

формадағы қатынас, мұндағы $b_{m+1},\ldots,b_n$ структураның қозғалмайтын элементтері. Формула φ және элементтер $b_{m+1},\ldots,b_n$ берілсе, онда $b_{m+1},\ldots,b_n$-ді $x_{m+1},\ldots,x_n$ айнымалыларына меншіктейтін IND программаны құрастыруға болады. Осыдан кейін, формуланы жоғарыдан төмен қарай жіктеу арқылы программа F_φ-тің $x_1,\ldots,x_n$ қанағаттандыратындығын тексереді. Пропозициялық байланымдар үшін басқару ағынын және атомдар үшін шартты тексерулерді пайдалана отырып, программа экзистенциалды кванторларда экзистенциалды меншіктеулер жасайды және универсиалды кванторларда универсалды меншіктеулер жасайды. Және ол индуктивті айнымалы R пайда болысымен программаның басқы жағына қарай циклды кері жасайды. Жоғарыда қарастырылған мысалдар транзитивті тұйықталу үшін рефлексивті, ойын және фундирлілік осы процесті иллюстрация жасайды.

Керісінше, IND программасы бойынша есептелетін кез келген қатынас индуктивті болып келеді. Оған қатты әсер ететін IND программада қабылданған анықтамалар есептеу бұтағының индуктивті түрде анықталған белгілер жиынының қозғалмайтын нүктелерін өзінде ұстайды.

Индуктивті және гиперэлементар қатынастар

Жоғарыда қарастырылған IND программалардың көбі, $\mathbb{N}$-нен басқа да кез келген структура (құрылым) үшін де интерпретация жасасақ деген мағына береді. Кез келген структураның $\mathfrak{A}$ индуктивті қатынасын, $\mathfrak{A}$-да IND программа арқылы есептелінетін қатынас арқылы анықтаймыз. $\mathfrak{A}$-ның гиперэлементар қатынасы, $\mathfrak{A}$-да тоталды IND программамен есептелетін қатынаспен анықтаймыз; яғни барлық енгізуде тоқтайтын программалар қарастырылады. $\mathfrak{A}$-ның элементар қатынастары тек бірінші ретті қатынас арқылы анықталады. Барлық элементар қатынастар гиперэлементарлы болады, өйткені олар әрқашан тоқтайтын бейциклды программалар арқылы есептелінеді. $\mathbb{N}$-де гиперэлементарлы және элементарлы қатынастар сәйкес гиперарифметикалы және арифметикалы деп аталады. Арифметикалық болмайтын гиперарифметикалық жиынның мысалы ретінде, бірінші ретті сандар теориясын $\mathrm{Th}(\mathbb{N})$ келтіруге болады (142-аралас жаттығу).

Қатынас $\mathfrak{A}$-де гиперэлементарлы болуы үшін, Сонда тек сонда $\mathfrak{A}$-де анықталған қатынас индуктивті және ко-индуктивті болса, онда ол гиперэлементар болады: егер R-ді қабылдайтын IND программа болса және $\sim R$ қабылдайтын басқа IND программа бар болса, онда Тюринг машинасының сәйкес нәтижесіне сай, осы программаларды параллель қоса алатын жалпы программа құрастыруға болады.

40-лекция

Клин теоремасы

Бұл лекцияда $\mathbb{N}$ арифметикалық структурасына ғана назар аударатын боламыз. Қарастырылатын структурада гиперэлементар қатынастар кейде гиперарифметикалық қатынас деп те аталады.

Рекурсивті бұтақ, рекурсивті ординал және ω_1^{ck}

Егер ординал мен ω арасында биекция бар болса, онда ординал саналады деп айтамыз. $\omega \cdot 2$ және ω^2 ординалдары, ω-дан үлкен болса да әлі саналатын болады. Ең төменгі саналмайтын ординал ω_1 деп аталады.

Дәстүр бойынша, рекурсивті ординал жалғыз түрде анықталады, бұл жерде ординал мен ω арасында саналатын биекция бар болады, яғни, керекті ординалды кодтау мен есептелу түсінігі (Rogers [104] қараңыз) қажет болады. Ең төменгі бейрекурсивті ординал ω_1^{ck} деп аталады. Ол саналатын ординал, бірақ кез келген есептелетін функция үшін саналмайтын сияқты көрінеді.

Рекурсивті ординалдарды рекурсивті ω-бұтақтар деп аталатын индуктивті белгілер көмегімен анықтаймыз. ω-бұтақтар ω^*-ның бос емес префикс тұйық жиыны болады. Басқаша айтсақ, ұзындығы шектеулі натурал сандар жолынан тұратын жиын – T. Ол келесі шарттарға бағынады

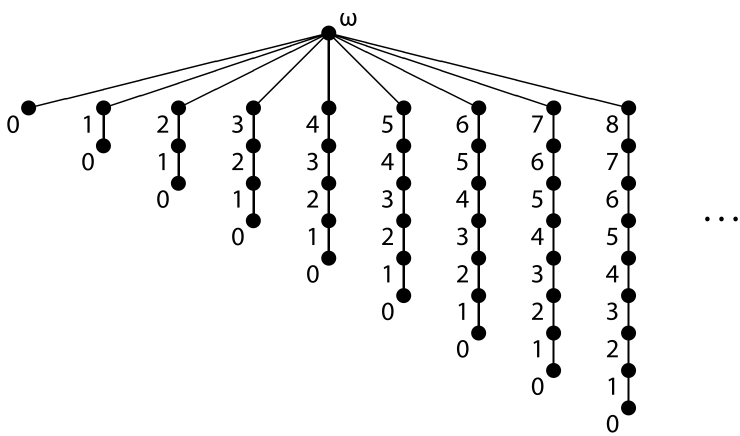
- $\epsilon \in T$, және
- егер $xy \in T$ болса, онда $x \in T$.

Сызықты реттелген префикс қатынаспен анықталған T-ның максималды ішкі жиыны T-дағы траектория деп аталады. Егер шексіз емес траекториялар бар болса, онда T бұтақ фундирленген деп аталады. T-ның басқа ешқандай элементінің префиксі болмайтын T-ның элементі жапырақ деп аталады. Егер T жиыны керекті түрде кодталған рекурсивті жиын болса, онда ω-бұтақты T рекурсивті деп аталады.

Фундирлі бұтақ T берілсін, белгі $o : T \to \mathrm{Ord}$ -ны келесі түрде

$$o(x) \overset{\mathrm{def}}{=} \sup_{\substack{n \in \omega \\ xn \in T}} (o(xn)+1)$$

индуктивті анықтаймыз. Сондықтан, егер x жапырақ болса, онда $o(x)=0$. Және егер x жапырақ болмаса, онда барлық $xn \in T$ үшін $o(x)$ алдымен $o(xn)$-пен анықталады, осыдан кейін барлық ординалдар мұрагерінен супремум алынады.



Мысалы, $n \geq 0$ және $m \leq n$ үшін ε және $(n, \underbrace{0, 0, \ldots, 0}_{m})$ түріндегі барлық тізбектен тұратын бұтақты қарастырамыз. Жапырақтар O-дің көмегі арқылы 0-мен белгіленеді, ал жапырақ үстіндегі келесі элемент 1-мен белгіленеді, тағы сол сияқты. Түбір ε ω-мен белгіленеді. Фундирленген бұтақ T үшін, T-ның түбіріне меншіктелген ординал $o(T)$ болсын. Әрбір $o(T)$ саналатын ординал және $\sup_T o(T) = \omega_1$.

Егер белгілі бір рекурсивті бұтақ T үшін ординал $o(T)$ болса, онда ол рекурсивті ординал болады. Рекурсивті ординалдар супремумы ω_1^{ck} болады.

Рекурсивті ординалдардың альтернативті анықтамасы ретінде барлық уақыт жиынындағы **IND** программаларды айтуға болады. Белгілі бір енгізудегі **IND** программалардың жұмыс уақыты, түбірді 1 немесе 0-мен белгілеуге кеткен уақыт. Бұл есептеу бұтағының қабылдау-ашық күйінің формалды анықтаудағы белгіні индуктивті анықтау, ол тұйықтайтын ординал болып табылады. Бұл рекурциялы бұтақты O белгісімен дефинациялауға ұқсас. ω_1^{ck} ординалы **IND** программаның барлық жұмыс уақытының супремумы.

Клин теоремасы

40.1-теорема (Клин [74]) $\mathbb{N}$ үшін *индуктивті қатынас және* Π_1^1 қатынас беттеседі, және гиперэлементарлы қатынас және Δ_1^1 қатынас та беттеседі.

Дәлелдеу схемасы. Алдымен кез келген индуктивті қатынас Π_1^1 болатынын көрсетеміз. Бұл бағыт, $\mathbb{N}$-нен басқа кез келген $\mathfrak{A}$ структура үшін де дұрыс болады. Қозғалыссыз нүктесі $F_\varphi \subseteq A^n$ болатын бірінші ретті оң формула $\varphi(\overline{x}, R)$ болсын, мұндағы A $\mathfrak{A}$-ның тасушысы. Біз F_φ-ді φ-ге қарасты барлық тұйық қатынастар қиылысуы ретінде жаза аламыз

$$F_\varphi(\overline{x}) \quad \Leftrightarrow \quad \forall R \ (\forall \overline{y} \ \varphi(\overline{y}, R) \to R(\overline{y})) \to R(\overline{x}).$$

Бұл формула Π_1^1.

Екінші жағынан $\mathbb{N}$-де кез келген Π_1^1 формуланы қарастырайық. Бұрын айтылғандай, формуланы манипуляциялаудың әртүрлі ережесін пайдаланып, жалпылықты сақтай отырып, формула

$$\forall f\ \exists(x)\ \varphi(x,f), \qquad (40.1)$$

түрінде жазылады дей аламыз, мұндағы φ-де кванторлар бар (141-аралас жаттығу).

Егер $f:\omega\to\omega$ функциясын, меншікті мәндердің $f(0), f(1), f(2),\ldots$,шексіз жолы ретінде қарастырсақ, онда функция f толық бұтақ ω^*-ның траекторияларымен бірдің бірге қатынасында болады. Бұдан басқа, кез келген x үшін, $\varphi(x,f)$-тің ақиқат болуы осы траекторияның шектеулі префиксімен анықталады. Ол префикс f-ке апаратын барлық аргументті және $\varphi(x,f)$ -ке кіретін барлық мүшелерді қамтиды. Ұзындығы n болатын шектеулі префикс f-ті $f\restriction n$ деп белгілейік. Біз $f\restriction n$-ді не ұзындығы n болатын натурал сандар жолы не дербес функция ретінде қарастыра аламыз. Мұндағы дербес функция f пен анықталу аймағы $\{0,1,\ldots\ ,n-1\}$ арқылы сәйкестенеді.

Егер $\varphi(x,f)$ ақиқат болатындай $f\restriction n$-де жеткілікті ақпарат болса, немесе кері жағдайда 0 болатын болса онда $\varphi'(n,x,f)$ формуласы $\varphi(x,f)$-пен бірдей аққиқат мәнге ие болатын функция болсын. Біз φ-ді пайдаланып, φ'-ті жеңіл тауып аламыз. Мысалы, егер $\varphi(x,f)$ қатынасы $x=f(f(x))$ болып және ол циклі жоқ IND программаға

if $x=f(f(x))$ then accept else reject,

эквивалентті болса, онда $\varphi'(n,x,f)$-ті бірінші ретті формула деп танимыз және ол циклі жоқ IND программаға

```
if x < n {              // f↾n(x) анықталған ба?
  y := f(x);            // егер солай болса, онда y оның мәні болсын
   if y < n {           // f↾n(f↾n(x)) анықталған ба?
     z := f(y);         // егер солай болса, онда z оның мәні болсын
     if x = z accept;   // x = f(f(x))-ті тексеру
   }
}
reject;
```

эквивалентті болады. Енді біз (40.1)-дің орнына

$$\forall f\ \exists x\ \varphi'(n,x,f\uparrow n) \qquad (40.2)$$

деп жаза аламыз. Назар аударайық, егер $\exists x\ \varphi'(n,x,f)$ болса, онда барлық $m \geq n$ үшін $\exists x\ \varphi'(m, x, f)$ болады. Бұл (40.2)-нің фундирлі шарт екендігін көрсетеді: егер шексіз бұтақ төбелерін $f \upharpoonright n$ ақиқаттық мәнмен $\exists x\ \varphi'(n, x, f)$ белгілейтін болсақ, онда ең соңында бұтақтың кез келген траекториясында біз 1 мәніне тап боламыз деп (40.2) айтады. 39-лекцияда байқалғандай фундирлілік индуктивті болады.

$\mathbb{N}$-де анықталған индуктивті және Π_1^1 қатынастар беттесетінін көрсеттік. Гиперарифметикалық қатынастар индуктивті және ко-индуктивті, ал Δ_1^1 қатынас Π_1^1 және Σ_1^1 болатындықтан, гиперарифметикалық және Δ_1^1 қатынастар да беттеседі.

Гиперэлементарлықтағы индуктивті экзистенциалдылық

$\mathbb{N}$-де Π_1^1-дің сипаттамалары **IND** программаның қабылдау-ашық жиындары сияқты және Δ_1^1-дің сипаттамалары жалпы **IND** программаның жиындары сияқты қарастырайық. Сонда, индуктивті және р.с. жиындар арасында және гиперэлементар және рекурсивті жиындар арасында күшті аналогия бар екендігі айқын байқалады.

Аналитикалық деңгейде Σ_1^0 сияқты класс Σ_1^1 болмай, Π_1^1 болатыны көзге оғаш көрінуі мүмкін. Оны (35.1)-де көрсетілген, р.с. жиынның сипаттамасына сәйкес келетін келесі нәтижемен түсіндіруге болады.

40.2-теорема Егер

$$A = \{x \mid \exists \alpha < \omega_1^{\mathrm{ck}}\ R(x, \alpha)\} \tag{40.3}$$

$$= \left\{x \mid \exists y\ \ y \text{ ординал мен } R(x, y) \text{-ты кодтайды}\right\}$$

шарттарына бағынатын элементар қатынас R бар болса, сонда тек сонда жиын $A \subseteq \mathbb{N}$ индуктивті болады.

Дәлелдеу нобайы. Егер R гиперэлементар болса, онда (40.3) үшін программадан $y := \exists$ тұжырымы шығатын **IND** программа құрастыра аламыз, ол y индексті Тюринг машинасында фундирлі рекурсивті бұтаққа қабылдау ашық және $R(x, y)$ болатынын параллельді түрде тексереді.

Керісінше, **IND** программа p арқылы қабылдау-ашық болған A индуктивті болсын, сонда A-ны экзистенциалды формула сипаттай алады, онда « рекурсивті ординал α табылып, α қадамда p тоқтайды және x-ке қабылдау ашық болады” деп айтамыз. Дәлірек айтсақ, «фундирленген рекурсивті бұтақ Т табылып, $o(T)$ қадам ішінде енгізу x-те p тоқтайды және қабылдау-ашық болады». Квантирлеу Тюринг машинасының индексіне жүргізіледі. Предикат «$o(T)$ қадам ішінде енгізу x-те p тоқтайды және қабылдау-ашық болады» деген тұжырым гиперэлементар екендігін дәлелдеу үшін, **IND** программа құрастырамыз. Ол программа p мен қоса $o(T)$ қадамда тоқтайтын q программасын да іске қосады (q экзистенциалды тармақталуды пайдаланып Т-ны тізбелеп шығады және қабылдау-жабық күй жасайды) және бірінші кезекте жасалатын амалдарды іске асырады.

41-лекция

Әділетті тоқтату және Харсел теоремасы

39 және 40-лекциялардан кейін, сізде ω_1^{ck} мен Π_1^1-нің компьютерлік ғылымға жақындығы шамалы екен деген көзқарас пайда болуы мүмкін. Олардың пайда болуына байланысты, сізге қолданысқа жарамды нақты мына мысалды келтірейік: параллель программалардың әділетті тоқтауының дәлелдемесі.

Әрбір қадамда тоқтауға жақындағандықты көрсету үшін тоқтаудың дәлелдемесі әдетте индукцияға негізделеді. Кәдімгі қарапайым тізбектелген программада, натурал сандармен ω жасалатын индукция әдетте жеткілікті.

Мысалы, берілген бүтін оң екі x, y сандарының ең үлкен ортақ бөлгішін (ЕҮОБ) есептейтін программаны қарастырайық. Программа есептеп біткенде ЕҮОБ айнымалы x-те сақталады.

$$
\begin{aligned}
&\text{while } (y \neq 0) \ \{ \\
&\quad z := x \mod y; \\
&\quad x := y; \\
&\quad y := z; \\
&\}
\end{aligned}
$$

Бұл программа кез келген теріс емес бүтін сандар x, y үшін түбінде аяқталады, өйткені циклдің әрбір итерациясы y-тің мәнін теріс болмауға және қатаң түрде кемітіп отыруға мәжбүрлейді, сондықтан тоқтау бағытындағы прогресс орындалады. Осыны дәлелдеп шығу үшін ω-дағы үйреншікті индукция жеткілікті.

Параллель программа үшін бұл біршама күрделі болып кетеді. Мұнда қатар жұмыс жасайтын бірнеше процесс керек болуы мүмкін. Бұл процестер ресурсқа таласуы әбден мүмкін, мысалы ортақ процессордағы есептеу уақыты немесе ортақ айнымалыға қабылдау-ашық болу мүмкіндігі. Қатарласа жасайтын программалар жұмысын модельдеуде, бейдетерминизм жиі кездеседі, өйткені біз бірнеше мүмкіндіктерді білгенімізбен, әр жағдайда қандай шарттылық шешілетінін, алдын ала дөп басып айта алмай қалуымыз мүмкін. Сондықтан есептеуді бұтақты ағашпен модельдеуге болады, оның әрбір траекториясы есептеу жүйесінің мүмкін жолы болып кетеді.

Өкінішке орай, бейдетерминизмнің үйреншікті семантикасында, қызықтырмайтын себептермен есептеудің кейбір жолдары тоқтаусыз кетуі мүмкін. Мысалы, келесі бейдетерминирлі программаны қарастырайық.

$$
\begin{aligned}
&x := 0; \\
&y := 0; \\
&\text{while } (x < 10 \ \vee \ y < 10)\{ \\
&\quad x := x + 1 \,||\, y := y + 1; \\
&\}
\end{aligned}
\tag{41.1}
$$

Мұндағы $||$ таңбасы “не p немесе q-ды жаса” дегенді білдіреді. Үйреншікті бейдетерминизм семантикасына сүйенсек, бұл программаның тоқтамайтын есептеуі бар; мысалда ол біреу, циклдің сол жақ тармағы әрқашан таңдалады. Дегенмен, сол жақты әрқашан таңдайтын кез келген жобалаушы әділетсіз болып саналады, өйткені оң жақ тармаққа жұмыс жасауға мүмкіндік бермеді, оң тармаққа шексіз көп мүмкіндік берсе де, ол жұмыс жасай алмас еді.

Сондықтан бейдетерминирлі шешетін агент таңдау нүктелерінде әділетті түрде таңдау жасайды деп ұйғарым жасауымызға болатын шығар. Бірақ бұл шара қалай іске асатынын білмеуіміз (немесе мазасызданбауымыз) де мүмкін. Біз жобалаушының кіршіксіз таза мінезіне сүйеніп, абстракциялаймыз және әділеттіліктің кейбір формалды қасиеттерін ұйғарамыз. Мысалы, егер $p \,||\, q$ формасының тұжырымы шексіз жиі қосылған (керек) болса, онда әрбір p және q атқарылуға шексіз жиі таңдалады, деп ұйғарым жасауымызға болады. Осы ұйғарым бойынша, жоғарыда келтірілген мысалдың барлық шексіз есептеу траекториялары әділетсіз болып шығады; басқаша айтсақ, барлық әділетті траекториялар тоқтатылған. Осындай ұйғарымдар корректілік қасиетті зерттеуге мүмкіндік береді, мысалы, енді тоқтату жобалаушының іске асыру жұмысынан тәуелсіз.

Әділетті тоқтату

Әділетті тоқтату проблемасы берілген параллель программа әділетті жобалаушының ұйғарымымен тоқтай ала ма, соны анықтау проблемасы. Интуитция бойынша, параллельді программа (41.1) әділетті тоқтайды деуге болады, өйткені тоқтатуды шақыратын кез келген амалдар тізбегін ешуақытта әділетті жобалаушы таңдамайды.

Әділеттілік шартты формалды түрде қасиеттің (ρ,σ) жұбы ретінде анықтаймыз, олар есептеудің ақиқат немесе өтірік күйімен анықталады. Біз ρ-ны ресурсқа сұратым жасайтын өрнек ретінде қарастырамыз. Сондықтан егер ρ мен байланысқан қандай да бір ресурс, қандай болғанына қарамай, дәл осы уақытта сұратылып жатса, онда s күйінің ақиқаттығы ρ болады ($s \models \rho$ деп жазылады). Біз σ-ны сұратыммен қанағаттандырылған шарттың өрнегі деп қарастырамыз.

Егер шексіз есептеу жолында, ρ шексіз жиі ақиқат болса, ал σ тек шектеулі сан рет ақиқат болса, онда әділеттік шарт (ρ,σ) -ға қатысты шексіз есептеу траекториясы әділетсіз деп аталады. Бұл арқылы, траектория бойында сұратым шексіз жиі жасалады, бірақ белгілі бір орыннан бастап ешқашан қанағаттандырылмайды деген идея модельденеді. Егер траектория әділеттілік шарттарының жиынтығының кез келгенінде әділетсіз болса, онда ол әділеттілік шарттарының жиынтығында әділетсіз деп аталады. (Шексіз автоматтар жайлы Рабиннің қабылдау шартына ұқсастығына назар аударыңыздар; 26-шы лекцияны қараңыз). Траектория әділетсіз болмаса, онда ол әділетті болады. Біздің анықтама бойынша, барлық шектеулі траектория әділетті болады.

Егер шексіз әділетті траектория жоқ болса, онда есептеу (бейдетерминирлі) әділетті тоқталады деп аталады; эквивалентті, егер барлық шексіз траекториялар әділетсіз болса. Интуитивті, егер есептеу бұтағы шексіз траекториялы және олар әділетсіз болса да қамти берсе, оған біз көп алаңдамаймыз, өйткені әділетті жобалаушы ол процестің шексіз жүре беруіне ешқашан жол бермейді.

Біздің (41.1) мысалдағы, while циклінде ℓ тұжырым болсын. ℓ соңғы рет орындалғанда оң немесе сол тармақта қабылдау-ашық болды ма соны білу үшін, ағымдағы күйде біршама бөгелеміз. Қасиет ρ “ℓ-ді орындағысы келеді” болсын және σ_0 (сәйкес σ_1) “ℓ соңғы рет орындалғанда сол (сәйкес оң) тармаққа қабылдау-ашық болған» қасиет болсын. Жұптар (ρ,σ_0) және (ρ,σ_1)-ның әділетті болу шарттарын қарастырамыз. Мұндағы, σ_0 мен σ_1-ді жеткілікті көп рет қанағаттандыратын кез келген есептеу жолдары, олар ең соңында цикл while-ден шығу шартын қанағаттандыратын болғандықтан, бұл шарттарға қатысты программа әділетті тоқтайды.

Әділетті тоқтату үшін дәлелдеу ережесі

Әділеттілік және әділетті тоқтатуға арналған оқулықтар өте көп ([44] және оның сілтемелерін қараңыз). Осы еңбектердің басым бөлігі, әртүрлі логикалық формалдаулар және әртүрлі әділеттілік ұйғарымдарында, дұрыстық және тоқтатуды қалыптастыру үшін дәлелдеу ережелерін алуға арналған. Негізгі түсінік пайдалы бағыт идеясы болып табылады, ол арқылы есептеу тоқтатуға қарай жылжытылады. Бұл түсінік соңында фундирлілікке апарып тірейді, бірақ ол тек бүтін параметрді азайту ғана емес. Ординалдағы трансфинитті функция ω-дан жоғары болатындығы өте қажет екендігі анықталды.

Харел теоремасы

Жағдайды 1986 жылы Харел [52] едәуір айқындай түсті. Ол шектеулі рекурсивті тармақты бұтақтардың әділетті тоқтатылуы саналатын рекурсивті тармақты бұтақтардың фундирлілігіне эквивалентті екендігін көрсетті. Ал саналатын рекурсивті тармақталған бұтақтардың фундирлілік болуының шешімі Π_1^1-болады (12-үй жұмысы, 1 (b) жаттығу) және фундирленген саналатын рекурсивті тармақты бұтақтардың ординалдарының супремумы ω_1^{ck} болады. Харел теоремасының қорытындысы: әділетті тоқтату проблемасы Π_1^1-толық болады және әділетті тоқтату дәлелдемесіне қатысатын ординалдар бүтіндей ω_1^{ck} сияқты үлкен болуы мүмкін.

Харел теоремасын, жалпы жағдай болмаса да теореманың негізгі идеясын беретін ерекше жағдай үшін дәлелдейміз. Бинарлы бұтақты префикс-тұйықталған бос емес ішкі жиын $\{0,1\}^*$ ретінде анықтаймыз. Дәл осылайша, ω-бұтақ префикс-тұйықталған бос емес ішкі жиын ω^*. Бұтақтың кез келген түрінде, префикс қатынаспен сызықты реттелген максималды ішкі жиын траектория болып табылады.

Бинарлы бұтақтар T үшін әділеттілік шартын $(\mathrm{true}, \mathrm{last}(0))$ қарастырамыз, мұнда $x \in \{0,1\}^*$ үшін $x \models \mathrm{last}(0)$, егер қандайда бір траектория y үшін $x=y0$ болса; яғни x-тық соңғы әрпі 0 болса. T-ның осы әділеттілік шартқа қатысты әділетсіз жолдары, $x1^\omega$ формадағы шексіз жолдардың шектеулі префикстерінің барлық жиындары болады, мұндағы $x \in \{0,1\}^*$. 0-ді "солға жүр" және 1-ді "оңға жүр" ретінде қарастырайық, сонда әділетсіз траектория оңға жүретін траектория болып шығады, бірақ шектеулі санды нүктеден тұрады. Егер солға шексіз жиі жүретін, шексіз траектория жоқ болса, онда бұтақ T әділетті тоқтайды.

Бинарлық бұтақтан ω-бұтаққа аударатын тиімді бейнені сипаттайық: егер тек егер бинарлы бұтақ әділетті тоқтаса, онда оған сәйкес ω-бұтақ фундирленген болады.

$x \in \{0,1\}^*$ үшін $\tau(x) = x_0 x_1 \cdots x_n$ -ді анықтаймыз, мұндағы x бір мәнді $1^{x_1}01^{x_2}0\cdots01^{x_n}$ түрінде жіктеледі. Мысалы,

$\tau(11010011111010011 1) = 2105103.$

Біз τ -ды индуктивті анықтай аламыз:

$$\tau(\varepsilon) \overset{\text{def}}{=} 0$$

$$\tau(x0) \overset{\text{def}}{=} \tau(x) \cdot 0$$

$$\tau(x1) \overset{\text{def}}{=} \text{lastinc}(\tau(x)),$$

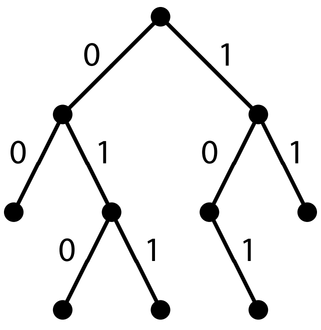
мұндағы $\text{lastinc}(xn) \overset{\text{def}}{=} x \cdot (n+1).$ ω^* -дағы бос траекториядан басқа жағдайда τ -дың бірдің бірге және суръективті болатынын жеңіл көрсетуге болады.

Енді, егер $T \subseteq \{0,1\}^*$ бинарлы бұтақ және

$$\tau(T) \overset{\text{def}}{=} \{\tau(x) \mid x \in T\} \cup \{\varepsilon\}$$

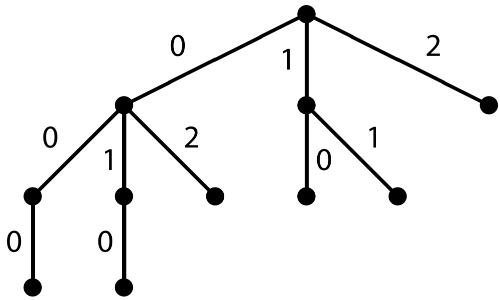
болсын. Сонда $\tau(T)$ бос емес және префикс-тұйықталған, сондықтан ω -бұтақ болады.

Мысалы, келесі бинарлы бұтақты $T \subseteq \{0,1\}^*$ қарастырайық.

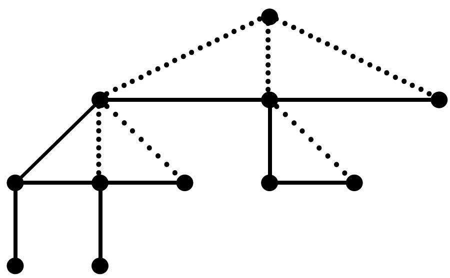


T -ның максималды элементтері 00, 010, 011, 101 және 11 болып табылады. Бұл жолдың әлі де аударылмаған префиксі 01, 10, 0, 1 және

ε. Максималды жолға τ-ды қолдансақ 000, 010, 02, 11 және 2 алынады, ал τ-ды басқа префикстерге аударсақ 01, 10, 00, 1 және 0 береді. Біз ε-ді де қосуымыз керек. Осының барлығы ω-бұтақты $\tau(T) \subseteq \omega^*$ береді.



T-дан бастап интуитивті түрде $\tau(T)$ конструкциясын қарастыру T-ның қабырғаларын басқаша бағыттау болып табылады. Бұл жағдайда T-да 0 мен белгіленгендер $\tau(T)$-ның төменгі сол жақ ең шеткі балаға дейін түседі және T-да 1-мен белгіленгендер $\tau(T)$-да оң жақтағы келесі бауырына барады. T-ның оң жақтағы ең шеткі омыртқасы $\tau(T)$-дағы түбірдің ұрпағы болып табылады.



Форма $\tau(T)$-ның ω-бұтақтары келесі қасиеттерге ие: егер $x \cdot (n+1) \in \tau(T)$ болса, онда $x \cdot (n) \in \tau(T)$. Егер ω-бұтақ осы қасиетке ие болса, онда ол бүтін деп аталады. Сонда әрбір $\tau(T)$ бүтін және белгілі бір бинарлы бұтақ T үшін әрбір бейтривиалды бүтін ω-бұтақ $\tau(T)$ болады.

Енді біз егер тек егер $\tau(T)$ фундирлі болса, онда T әділетті тоқтайды деп тұжырымдай аламыз. Егер тек егер $\tau(T)$-дың шексіз траекториясы бар болса, онда T-ның шексіз санды 0-ді шексіз траекториясы болады деген тұжырымды дәлелдегіміз келеді. Егер T-ның шексіз санды 0-ді шексіз траекториясы болса, онда бұл траектория $1^{x_0}01^{x_1}01^{x_2}0\cdots$ формадағы шексіз траекторияның шектеулі префикстер жиыны болады. Сонда шексіз $x_0x_1x_2\cdots$ жолдың барлық префикстері $\tau(T)$-ның мүшелері болады және бұл шексіз траектория. Керісінше, егер $\tau(T)$-тың шексіз траекториясы болса, онда ол $x_0x_1x_2\cdots$ формадағы шексіз жолдың шектеулі префикстерінің жиыны болуы қажет, сондықтан барлық n үшін $1^{x_0}01^{x_1}01^{x_2}0\cdots01^{x_n} \in T$ болады. Осындай жолдардың префикстерінің жиыны T-да жатады және шексіз әділетсіз траектория құрайды.

Біздің көрсеткеніміз:

41.1-теорема (Харел [52]) *Бейне τ бинарлы бұтақтар мен бейтривиалды бүтін ω-бұтақтан арасына бірдің бірге рекурсивті сәйкестігін құрайды, мұнда егер $\tau(T)$ фундирленген болса, сонда тек сонда бинарлы бұтақ T, әділеттілік шарт* $(\mathrm{true}, \mathrm{last}(0))$*-ге қарасты, әділетті тоқтатылған болады.*

41.2-салдар. Әділетті тоқтату Π_1^1-толық болады.

Дәлелдеу. 143-аралас жаттығу.

ЖАТТЫҒУЛАР

1-үй жұмысы

1. (a) Бейрегулярлы жиын қабылдайтын $O(n \log n)$ уақытпен шектелген жалғыз таспалы детерминирлі Тьюринг машинасын көрсетіңіз.
(b) $o(n \log n)$ уақытта жалғыз таспалы детерминирлі Тьюринг машинасында қабылдау-ашық болатын кез келген жиын регулярлы болатынын көрсетіңіз.

2. Санақшысы сызықты шектеулі *k-сана*қшылы автомат деп, тек оқуға арналған екіжақты бас тиекті енгізушісі бар бір таспалы ТМ және енгізу жолы 0 және n сандарының арасынан шықпайтын, тиянақты шектеулі санды бүтін мәнді санақшыларды айтамыз. Әрбір қадамда, әрбір санақшыны машина нөлге тексере алады. Осы ақпаратқа сүйеніп, өзінің ағымдағы күйінде және осы сәтте сканерленетін енгізу символын пайдаланып, ол санақшыларға бір санақшыны қосады немесе бір санақшыны алып тастай алады, ақпаратты оқу үшін бас тиекті оңға немесе солға жылжытады және жаңа күйге енеді.

(a) Осы машиналарға формалды анықтама беріңіз, оған қоса қабылдау-ашық түсінігіне анықтама беріңіз.

(b) Егер тек егер белгілі бір k үшін шектелген сызықты санақшылары бар k-санақшы детерминирлі (бейдетерминирлі) машинада қабылдау-ашық болса, онда ол жиын LOGSPACE (NLOGSPACE)-те жататынын көрсетіңіз.

3. Егер $P = NP$ болса, онда $NEXPTIME = EXPTIME$ болатынын дәлелдеңіз. (Көмек. Енгізуді артық $\#$ -мен толтырыңыз).

2-үй жұмысы

1. Келтірімділік қатынасы $\le_{\mathrm{m}}^{\log}$ транзитивті екенін дәлелдеңіз: егер $A \le_{\mathrm{m}}^{\log} B$ және $B \le_{\mathrm{m}}^{\log} C$ болса, онда $A \le_{\mathrm{m}}^{\log} C$ (Ескерту. Бұл бейтривиал емес! Аралық нәтижені толық көлемде жазып алатын, сізде жеткілікті орын жоқ).

2. Егер логикалық формула дизъюнкция операндыларының $\ell \vee \ell'$ түріндегі конъюнциясы болса, онда логикалық формула *2-конъюнктивті қалыпты түрде* ($2\mathrm{CNF}$) жазылған дейміз, мұндағы, ℓ мен ℓ' литералдар (логикалық айнымалылар немесе айнымалыны теріске шығару). $2\mathrm{CNF}$-де логикалық айнымалылардың орындалушылық проблемасын шешу $2\mathrm{SAT}$ деп белгіленеді. $\le_{\mathrm{m}}^{\log}$ амалында co-$NLOGSPACE$ үшін $2\mathrm{SAT}$ толық екенін дәлелдеңіз.

3. Берілген ақиқатты меншіктеуде берілген логикалық формуланың мәні детерминирлі logspace-те есептеліне алатынын дәлелдеңіз.

4. Келесі бейрегуляр жиынды қарастырамыз

$$B = \{\$b_k(0)\$b_k(1)\$b_k(2)\$\cdots\$b_k(2^k-1)\$ \mid k \ge 0\} \subseteq \{0,1,\$\}^*,$$

мұнда, $b_k(i)$ арқылы i-дің бинарлы k-битті түрде жазылуы белгіленген. Осы жиын $DSPACE(\log\log n)$-те жататынын көрсетіңіз. B үшін детерминирлі ТМ-ын бере салу жеткіліксіз екенін айтып өтейік, ол B-да әрбір қабылдау-ашық есептеу $O(\log\log n)$ кеңістік алады. Жарияланған анықтама бойынша, B-ның күрделілік класы $DSPACE(\log\log n)$-та жататынын дәлелдей алу үшін біз B үшін бейдетерминирлі ТМ-ды көрсете алуымыз қажет. Бұл машинада әрбір есептеу, қабылдау-ашық, қабылдау-жабық немесе цикл түзейтіні болсын $O(\log\log n)$ кеңістік алады.

3-үй жұмысы

1. (a) Теңгерілген жақшалар жолының жиыны контексті еркін грамматика

$$S \;\to\; (S) \mid SS \mid \varepsilon$$

көмегімен туындалған. Осы жиын $LOGSPACE$ -те жататынын дәлелдеңіз.

(b) Теңгерілген жақшалардың екі типінде

$$S \;\to\; (S) \mid [S] \mid SS \mid {}^{o}$$

қалай болады?

2. Жалпыланған география ойынының шартына төбелерді қайта пайдалануға болады деген өзгеріс енді деп ұйғарайық. Яғни I және II ойыншылар бағытталған жол бойынан қабырғаны кезектесіп таңдайды (қарапайым болуы шарт емес), олар берілген s төбеден ойынды бастап және бір-бірін келесі жүріс табылмайтын қапасқа апарғысы келеді. I ойыншының ұтатын стратегиясының бар болуын анықтаудың күрделілігі қандай?

3. Алтернативті шектеулі автомат (AFA)

$$M \;=\; (Q,\, \Sigma,\, \delta,\, F,\; \alpha),$$

бес кортежді болады, мұндағы, Q -күйлердің шектеулі жиыны, Σ шектеулі енгізу автоматы, $F : \mathrm{Q} \to \{0,1\}$ соңғы немесе қабылдау-ашық күйлер жиынының сипаттамалық теңдеуі, яғни

$$F(q) \;=\; \begin{cases} 1, & \text{if q is a final state} \\ 0, & \text{otherwise,} \end{cases}$$

ал δ ауысу функциясы болып табылады:

$$\delta : (Q \times \Sigma) \quad \to \quad ((Q \to \{0,1\}) \to \{0,1\}),$$

және α қабылдау-ашық күй:

$$\alpha : (Q \to \{0,1\}) \quad \to \quad \{0,1\}.$$

Интуитивті бұтақ жапырағына F-тің көмегімен 0 немесе 1 белгілері салынады және барлық $q \in Q$ және $a \in \Sigma$ үшін логикалық функция

$$\delta(q,a) : (Q \to \{0,1\}) \quad \to \quad \{0,1\}$$

күйлерде белгілеулер алады және күй q үшін жаңа белгіні есептейді; осы арқылы есептеу бұтағымен жоғары қарай, кері бағытта логикалық белгілер 0 немесе 1 беріледі.

Формалды, δ ауысу функциясы

$$\hat{\delta} : (Q \times \Sigma^*) \quad \to \quad ((Q \to \{0,1\}) \quad \to \quad \{0,1\}),$$

бейнені бір мәнді анықтайды. Бұл бейне төмендегі түрде индуктивті анықталған: $q \in Q$, $a \in \Sigma$ және $x \in \Sigma^*$ үшін

$$\hat{\delta}(q,\varepsilon)(u) \quad = \quad u(q)$$

$$\hat{\delta}(q,ax)(u) \quad = \quad \hat{\delta}(q,a)(\lambda \mathrm{p}.(\hat{\delta}(p,x)(u))).$$

Егер

$$a(\lambda p.(\hat{\delta}(p,x)(F))) \;=\; 1$$

болса, онда машинада $x \in \Sigma^*$-ке қабылдау-ашық деп айтылады.

Егер $A \subseteq \Sigma^*$ жиынның керісінде

$$A^R \;=\; \{a_n \cdots a_1 \mid a_1 \cdots a_n \in A\}$$

детерминирлі шектеулі автоматпен (DFA) 2^k-күйге қабылдау-ашық болса, сонда тек сонда A жиында k-күйге алтернативті шектеулі автоматпен қабылдау-ашық болады.

4-үй жұмысы

1. Савич теоремасының келесі жалпыламасын дәлелдеңіз.
$S(n) \geq \log n$ үшін

$$STA(S(n),*,A(n)) \;\subseteq\; DSPACE(A(n)S(n)+S(n)^2)$$

қатынасы орындалады. (Көмек. Q-дегі күй универсалды немесе экзистенциалды бола ма соны анықтайтын алтернативті машина спецификасындағы бейне $\mathrm{type}: Q \to \{\wedge, \vee\}$ болсын. Конфигурация үшін type-ті жеңіл түрде кеңейтеміз. Ол үшін келесі предикатты қарастырамыз: $R(\alpha,\beta,k)$ = «α конфигурациясынан β конфигурациясына дейін ұзындығы k-дан аспайтын есептеу траекториясы табылады, ол траекторияда барлық конфигурация γ қатынас $\mathrm{type}(\gamma)=\mathrm{type}(\alpha)$-ты қанағаттандырады, бірақ β-ны қанағаттандыра алмай қалуы мүмкін.» Савич аргументін пайдаланыңыз).

2. PH -ке ұқсатып, Σ_k^{PSPACE} және Π_k^{PSPACE} деңгейлі PSPACE-тегі иерархияның формалды анықтамасын беріңіз. PSPACE үшін осы иерархия күйрейтінін көрсетіңіз. (Көмек. 1-жаттығуды пайдаланыңыз).

3. Σ_k^P үшін 9-лекцияда анықталғандай H_k толық жиын:

$$H_k \;=\; \{M\$x\$^d \mid M \text{ машинасы } d\text{-дан көп емес уақытта } x\text{-ке}$$

$$\text{қабылдау-ашық болатын } \Sigma_k \text{ машина}\}.$$

Екілік жүйеде y-пен өрнектелген сан $\#(y)$ болсын. Келесі жиын

$$H_\omega \;=\; \{y\$z \mid z \in H_{\#(y)}\}$$

PSPACE үшін $\leq_m^{\log}$-толық болатынын дәлелдеңіз.

4. Уақыт кеңістігінің орнына H_k-ға ұқсас G_k жиындар класын анықтаңыз:

$$G_k = \{M\$x\$^d \mid M$$ машинасы d-дан көп емес уақытта x-ке

қабылдау-ашық болатын Σ_k машина$\}$.

Σ_k^{PSPACE} үшін G_k жиыны $\leq_{\mathrm{m}}^{\log}$-толық болатынын және жиын

$$G_\omega = \{y\$z \mid z \in G_{\#(y)}\}$$

экспоненциалды уақыт үшін толық болатынын көрсетіңіз. Осыны 2- жаттығумен қалай сәйкестендіруге болады?

5- үй жұмысы

1. Егер $NP = \text{co-}NP$ болса, онда NP үшін PH-тың күйрейтінін көрсетіңіз. Жалпыланған түр, егер $\Sigma_k^{\text{p}} = \Pi_k^{\text{p}}$ болса, онда Σ_k^{p} үшін PH-тың күйрейтінін көрсетіңіз.

2. Логикалық орындалушылық проблемасы үшін полиномиал өлшемді схемалар тізбегі $B_0, B_1, B_2, \ldots$ бар болады деп ұйғарайық. Яғни B_n -нің n енгізуі, бір шығаруы және $n^{O(1)}$ қақпасы болады және $|x| = n$ болатын логикалық формула ($\{0,1\}^*$-те кодталған) x берілген, егер тек егер x қанағаттандырылса, онда $B_n(x) = 1$ болады.

(a) Егер тізбек $B_0, B_1, \ldots$полиномиалды уақытты, біртекті (яғни 0^n-нен B_n полиномиалды уақытта өндірілсе) болса, онда $P = NP$ болатынын дәлелдеңіз.

(b) Егер тізбек полиномиалды уақытты, біртекті болмай қалған жағдай болса, онда Σ_3^{p} үшін PH күйрейтінін дәлелдеңіз.

3. 5-лекцияда және 1- үй жұмысында, 2-жаттығуда, біз k-санақшы автоматты қарастырдық, оның санағышының мәні енгізу ұзындығы n-нен аспайтын. Бұл шектеулер болмаған жағдайда, екі-санақшы автомат кез келген Тьюринг машинасы сияқты қуатты болатыны белгілі ([61, 76]-ны қараңыз). Бейдетерминирлі (шектелмеген) бір-санақшы автоматтың қамтылу проблемасы NLOGSPACE үшін толық болатынын дәлелдеңіз. (Ескерту. Дәлелдеудің ең қиын бөлігі циклды қадағалау. Шектеулі санақшы жағдайдан мұндағы айырмашылық, ұзындығы n болатын енгізулерде шексіз мүмкін конфигурациялар жиыны табылады).

6-үй жұмысы

1. Шексіз кеңістікті конструкциялық функцияның $S(n) \le O(\log\log n)$-де бар болатынын біз 3- лекцияда көрсеттік. Бұл жаттығуда функцияның өзі $\lceil \log\log n \rceil$ кеңістікті конструкциялы болмайтынын көрсетеміз.

(a) Кез келген кеңістікті конструкциялық функция $S(n) \le o(\log n)$-де шексіз көп n үшін, $S(n) = k$ қатынасы орындалатын k табылатынын дәлелдеңіз. Басқаша айтсақ,

$$\lim_{n \ge 0} \inf_{m \ge n} S(m) < \infty.$$

(Көмек. Егер $S(n)$ кеңістікті конструкция болса, онда ұзындығы n -ге тең болатын кез келген енгізуде өзінің жұмыс таспасында тура $S(n)$ кеңістік бөлетін машина бар болуы керек. Ол машина $S(n)$-нен артық кеңістік пайдаланбайды және тоқтайды. Күй конфигурациясы санын, жұмыс таспасының мәнін және жұмыс таспасының бас тиегінің орнын санаңыз (оқу бас тиегінің орны емес). Өте ұзақ енгізу 0^n арқылы машина сканерлегенде не болатынын қарастырыңыз).

(b) Функция $\lceil \log\log n \rceil$ кеңістікті конструкция болмайтын жағдайды қарастырып, (a)-дан тұжырымдама жасаңыз.

2. Шешім қабылдау проблемасының $LM = \Sigma^*$ тұжырымын дәлелдеңіз: берілген бейдетерминирлі шектеулі автомат (NFA) үшін *M* машина *PSPACE*-толық болады. (Көмек. Келесі есептеу тарихын пайдаланыңыз

$$\#\alpha_0\#\alpha_1\#\cdots\#\alpha_N\#,$$

мұндағы, әрбір $\alpha_i \in \Delta^*$ *PSPACE*-те жұмыс жасайтын белгілі бір ТМ N машинасының конфигурациясының кодталуы және *N*-де қабылданған ауысу ережесі бойынша α_i-дан α_{i+1} бір қадамда алынады).

3. Егер M детерминирлі болса, онда 2-есептің күрделілігі қандай? Дәлелдемені келтіріңіз.

7-үй жұмысы

1. Құрылымның $(\omega, \leq)$ бірінші ретті күрделілік теориясын анықтаңыз, мұндағы ω натурал сандар жиыны және $\leq$ ω-дағы бізге таныс сызықты реттілік.
2. Екі ойыншы Соня (әдебиетте *Spoiler* деген атпен де белгілі) және Дэвид (*Duplicator* деген атпен де белгілі) арасындағы G_n ойын Ehrenfeucht–Fraissé-ні қарастырайық. Әр ойыншыда n құмалақ болады, оның әрқайсысының түсі n әртүрлі түстің біреуі. Ойыншылар құмалақтарды $\mathcal{A}$ және $\mathcal{B}$ екі сызықты реттіліктердің элементерінің орнына кезектесе орналастырады. Әрбір раундта Соня өзінде қалған құмалақтардың біреуін $\mathcal{A}$ немесе $\mathcal{B}$-ның белгілі бір элементінің орнына қояды. Осыдан кейін Давид өзіндегі дәл сондай түсті құмалақты басқа құрылымының белгілі бір элементінің орнына қояды. n раунд өткеннен кейін, егер құмалақтардың $\mathcal{A}$-да орналасу реті, $\mathcal{B}$-дағы орналасу ретіндей болса (бұл реттілікті анықтауда құмалақтардың түстері маңызды), онда Дэвид жеңімпаз деп жарияланады. Кері жағдайда Соня жеңімпаз деп танылады.

(a) Егер $\mathcal{A}$ рационал сандар жиыны, ал $\mathcal{B}$ бүтін сандар жиыны болса, онда G_3-те Соняда мәжбүрлі жеңісі бар болатынын дәлелдеңіз.

(b) Егер $\mathcal{A}$ рационал сандар жыйыны, ал $\mathcal{B}$ нақты сандар жыйыны болса, онда кез келген n үшін Дэвид G_n-де мәжбүрлі жеңіске жете алатынын дәлелдеңіз.

(c) Егер тек егер барлық бірінші ретті тереңдігі n кванторлы сөйлемдерде $\mathcal{A}$ мен $\mathcal{B}$ сәйкестендірілсе, онда Дэвид G_n -де мәжбүрлі жеңіске жете алатынын дәлелдеңіз. (Формуланың кванторлы тереңдігі – белгілі бір символ пайда болып отыратын кванторлардың максималды саны. Мысалы, келесі формулада

$$\exists x\ ((\forall y\ y \leq x) \wedge (\exists z\ z \leq x))$$

кванторлық тереңдік екіге тең.)

8-үй жұмысы

1. $\{0,1\}^{\omega}$-да жолдар жиыны ретінде көрсетілген ω-ның шектеулі ішкі жиындарына, бейдетерминирлі Бучи автоматында қабылдау-ашық болатынын, ал детерминирлі Бучи автоматында қабылдау-жабық болатынын дәлелдеңіз. (Еске түсіру: егер $IO(\sigma) \cap F \neq \varnothing$ болса, онда Бучи қабылдау-ашықта $F \subseteq Q$, және жүгірту σ-да қабылдау-ашық болады.)
2. (a) Бүтін сандарды қосу (яғни, предикат "$x = y + z$") S1S -те анықталмайтынын дәлелдеңіз. (Көмек. Автоматтар теориясындағы помпа техникасын пайдаланыңыз. Қараңыз, [61, 4.1 бөлім] немесе [76, 11, 12 лекциялар].)
   (b) Екінші жағынан, шектеулі жыйын $A \subseteq \omega$ үшін, анықтаңыз

$$n(A) = \sum_{x \in A} 2^{x} \quad .$$

Және $\varphi(A,B,C)$ предикат

"A,B,C шектеулі жиындар және $n(A) = n(B) + n(C)$" берілсін.
Осыны S1S -де қалай өрнектеуге болатынын көрсетіңіз.

3. (a) $n \geq 1$ үшін жиын $B_n \subseteq \omega$

$B_n = \Big\{ x \mid$ егер $x = mn + k,\ 0 \leq k < n$ болса, онда $\left\lfloor \frac{m}{2^k} \right\rfloor$ тақ болады $\Big\}$

болсын. Басқаша айтсақ, кез келген $m \geq 0$-де $\{0,1\}^{\omega}$-ның B_n-де өрнектелген mn-ші, $mn+1$-ші, …, $mn+n$-1-ші биттері; алдымен төменгі ретті биттерден бастап бинарлы жүйеде $m \bmod 2^n$-мен өрнектеледі. Мысалы,

$$B_n = \underbrace{0000000}_{n}\underbrace{1000000}_{n}\underbrace{0100000}_{n}\underbrace{1100000}_{n}\underbrace{0010000}_{n}\ldots\ .$$

Енді $\varphi_n(x,y)$ предикат "$x \equiv y (\operatorname{mod} n)$" болсын және $\psi_n(B)$

предикат “ $B = B_n$ ” болсын. Сонда φ_1 және ψ_1 үшін S1S формула құрастырыңыз және индуктивті түрде егер φ_n және ψ_n берілсе, онда φ_{n2^n} және ψ_{n2^n} үшін қысқа S1S формуланы қалай алуға болатынын көрсетіңіз.

(b) S1S-тің элементар еместігін, (a)-ны пайдаланып қалай көрсетуге болатындығын бейформалды түрде түсіндіріңіз.

9-үй жұмысы

1. Егер бірінші ретті сандар теориясы тілінің сөйлемі φ берілсе (қосу мен көбейтуді пайдаланса болады) және бинарлы жүйеде $n \geq 2$ саны берілсе, онда модуль n бойынша бүтін сандар сақинасы $\mathbb{Z}_n$-де φ-дің орындалу күрделілігін анықтау қандай болады? Дәлелдемесін беріңіз.
2. Енгізу алфавиті $\{0,1\}$ болатын кез келген бейдетерминирлі Мюллер автоматы M үшін, жалғыз еркін айнымалы X-тен ғана тәуелді және

$$L(M) \;=\; \{A \subseteq \omega \mid \varphi_M(A)\}$$

қатынасы орындалатын, S1S-тың қысқа формуласы $\varphi_M(X)$ табылады.
(Егер $IO(\sigma) \in \mathfrak{F}$ болса, онда Мюллер қабылдауы $\mathfrak{F} \subseteq 2^Q$ және жүгіртпе σ-да қабылданатын еске саламыз).
3. Кем дегенде, екі әріпті шектеулі алфавит Σ болсын. Егер $c > 0$ тұрақты табылып,

$$|A \cap \Sigma^n| \quad \leq \quad n^c \text{ б.ж.д.}$$

орындалса, онда жиын $A \subseteq \Sigma^*$ сиретілген деп аталады.
Басқаша айтсақ, барлық бірақ шектеулі жиынды n үшін, A-ның ұзындығы n-ге тең элементтерінің саны полиноммен шектеулі. Сиретілген оракулды полиномиалды уақытта детерминирлі оракул машинамен есептелінетін жиындар класы P^{sparse}, біртекті болмаса да болатын полиномиалды өлшемді $B_0, B_1, \ldots$ схемалар табылатын жиындардың P^{sparse} дәл класы болатынын дәлелдеңіз.

10-үй жұмысы

1. Универсал функция U және s_n^m функцияны пайдаланып,

$$\varphi_{\mathrm{pair}(i,j)} = <\varphi_i, \varphi_j> \qquad \varphi_{\mathrm{const}(i)} = \kappa_i$$

қатынастары орындалатындай етіп жалпы рекурсиялы функциялар pair және const-ты құрастырыңыз.
Құрастырылатын конструкция 33-ші лекцияда келтірілген конструкция comp-қа ұқсас болуы қажет.
2. Рекурсия жайлы теоремада (33.1-теорема), кез келген жалпы рекурсиялы функция σ үшін $\varphi_{\sigma(i)} = \varphi_i$ қатынасы орындалатын индекс i табылатынын дәлелдеген болатынбыз. Осындай индекс i-ді σ-ның индексінен тиімді алуға болатынын дәлелдеңіз. Яғни, барлық j үшін φ_j ортақ

$$\varphi_{\varphi_j(\mathrm{fix}(j))} = \varphi_{\mathrm{fix}(j)}$$

болатын, жалпы рекурсиялы функция fix бар екенін дәлелдеңіз.
3. Жалпы рекурсиялы функция σ-ның шексіз көп қозғалыссыз нүктесі бар екенін дәлелдеңіз; одан басқа қозғалыссыз нүктелердің шексіз тізімі тиімді тізбеленетінін де дәлелдеңіз.

11-үй жұмысы

1. Ауысу амалы (') келесі түрде анықталады:

$$A' = K^A = \{x \mid \varphi_x^A(x)\downarrow\}.$$

Бұл тоқтау проблемасы A-ға релятивтелген. Анықтаймыз

$$A^{(0)} = A$$

$$A^{(n+1)} = (A^{(n)})'.$$

Σ_n^0, $n \geq 1$ үшін $\varnothing^{(n)}$ жиыны $\leq_m$-толық болатынын көрсетіңіз.

2. (a) Егер Σ_n^0, $n \geq 1$ үшін A жиыны $\leq_m$-толық болса, онда алдыңғы жаттығуда анықталған A' жиыны Σ_n^0-де жатпайтынын көрсетіңіз.
(b) Барлық $n \geq 1$ үшін енгізуге қатысты Σ_n^0 және Π_n^0 салыстыруға келмейтінін (a)-дан қорытыңыз.

3. Рекурсия жайлы теореманың келесі үш релятивтелген версиясын қарастырайық.

(a) Кез келген жалпы рекурсиялы функция $\sigma : \omega \to \omega$ үшін, $\varphi_n^A = \varphi_{\sigma(n)}^A$ болатын n табылады.

(b) A-да рекурсивті кез келген жалпы функция $\sigma^A : \omega \to \omega$ үшін $\varphi_n = \varphi_{\sigma^A(n)}$ болатын n табылады.

(c) A-да рекурсивті кез келген жалпы функция $\sigma^A : \omega \to \omega$ үшін $\varphi_n^A = \varphi_{\sigma^A(n)}^A$ болатын n табылады.

Осының екеуі ақиқат ал біреуі жалған. Қайсысы қайда жатады? Екі дәлелдеу және бір кері мысал келтіріңіз.

4. Егер кез келген төбелер жұбы (u, v) үшін u-дан v-ға дейін бағытталған жол табылса, онда бағытталған граф қатаң байланған деп аталатынын естеріңізге саламыз. Π_2^0 үшін төменде келтірілетін проблема $\leq_m$-толық болатынын көрсетіңіз: егер рекурсивті бинарлық қатынас $E \subseteq \omega^2$ берілсе, онда шексіз граф (ω, E) қатаң байланған бола ма?

12-үй жұмысы

1. (a) Натурал сандардағы әрбір IND программа үшін экзистенциалды меншіктеуі $y := \exists$ болмайтын, бірақ қарапайым меншіктеуі $y := e(\overline{y})$ болатын эквивалентті IND программа табылатынын дәлелдеңіз. (Көмек. Алдымен $\ell_i \vee \ell_j$ конструкцияны пайдаланып, саналатын экзистенциалды тармақталуды шектеулі тармақталуға түрлендіріңіз; осыдан кейін шектеулі экзистенциалды тармақтардан құтылыңыз). Осы тәсілді пайдаланып $y := \forall$ -ді неге жоғалта алмаймыз?

(b) Егер шексіз кемімелі тізбе (цеп) жоқ болса, онда бинарлы қатынас фундирленген деп аталатынын еске саламыз. Келесі проблеманың Π_1^1 -толық болатынын (а)-ны пайдаланып көрсетіңіз. Егер рекурсивті бинарлы қатынас $R \subseteq \omega^2$ берілсе, онда ол фундирленген бола ала ма?

2. Гёдель *μ-рекурсивті функцияны* қосымша программалау конструкциясы араласқан примитивті рекурсивті функция (91-аралас жаттығу) ретінде анықтайды, атап айтсақ шексіз минимизациялау: егер $f : \omega^2 \to \omega$ μ-рекурсивті функция болса, онда

$$g(x) \overset{def}{=} \mu y.(f(x,y)=0),$$

болады, мұндағы оң жақта орналасқан өрнек ең аз кез келген y-те $z \le y$ үшін $f(x,z)\downarrow$ болады дегенді білдіреді және егер сондай y табылса, онда $f(x,y)=0$ болады, ал басқа жағдайларда анықталмайды. Аксиомалы түрде (яғни, 33-ші лекцияда байыпты баяндалған универсал функцияның қасиеттеріне және s_n^m -ге негізделген конструкцияға сүйеніп), егер f дербес рекурциялы функция болса, онда g-да дербес рекурциялы функция болады, ал g-дің индексін f-тің индексінен тиімді алуға болатынын дәлелдеңіз. 111-ші аралас жаттығудағы шартты тексеруді $\varphi_{\mathrm{cond}(i,j)}$ дәлелдемесіз қолдануыңызға болады.

3. $DTIME(T(n)) = DSPACE(T(n))$ қатынасы орындалатын, $T(n)$ функциясы табылатынын дәлелдеңіз.

АРАЛАС ЖАТТЫҒУЛАР

Мұндағы [H] аннотациясы осы жаттығуға 361-беттегі Көмектер секциясында, көмек бар екенін білдіреді және [S] аннотациясы Шешімдер секциясында жаттығудың шешімі келтірілгенін білдіреді. Жұлдызшалар саны жуық түрде есептің қиындығын білдіреді.

1. 3.2-теореманы дәлелдеңіз.
2. $\{0^n1^n \mid n \geq 0\}$ жиыны $\Omega(\log n)$ кеңістік талап ететінін көрсетіңіз.
3. Симуляцияның келесі нәтижелерін дәлелдеңіз:

(a) Тұрақтылар $k > 1$ және $\varepsilon > 0$ үшін $T(n)$ уақыт аралығында жұмыс
жасайтын кез келген k-таспалы TM машина, $\varepsilon T(n) + O(n)$ уақыт аралығында жұмыс жасайтын k-таспалы TM машинамен симуляция жасалына алады.

(b) Тұрақты $\varepsilon > 0$ үшін $T(n)$ уақыт аралығында жұмыс жасайтын кез келген 1-таспалы TM машина, $\varepsilon T(n) + O(n^2)$ уақыт аралығында жұмыс жасайтын 1-таспалы TM машинамен симуляция жасалына алады.

(c) $T(n) \geq n$ және тұрақты $k > 1$ үшін $T(n)$ уақыт аралығында жұмыс жасайтын кез келген k-таспалы ТМ машина, $T(n)^2$ уақыт аралығында жұмыс жасайтын 1-таспалы ТМ машинамен симуляция жасалына алады.

*(d) $T(n) \geq n$ және тұрақты $k > 1$ үшін $T(n)$ уақыт аралығында жұмыс жасайтын кез келген k-таспалы ТМ машина, $T(n)\log T(n)$ уақыт аралығында жұмыс жасайтын 2-таспалы ТМ машинамен симуляция жасалына алады.

4. Кез келген тиянақты $t > s \geq 1$ үшін, 3-лекцияда көрсетілген толтыру техникасын пайдаланып $NSPACE(n^s) \subsetneqq NSPACE(n^t)$ болатынын дәлелдеңіз.

**5. 3-лекцияда көрсетілген толтыру техникасын пайдаланып 4-ші аралас жаттығудың жалпыланған нұсқасын дәлелдеңіз. Нақты аргументті функцияның нақты мәні $S_1, S_2 : \mathbb{R} \to \mathbb{R}$ болсын. Олар төменде келтірілген сәйкес конструктивтілік шарттарын қанағаттандырады:

(a) S_1 мен S_2 монотонды өспелі, яғни егер $m < n$ болса, онда $S_1(m) < S_1(n)$ және $S_2(m) < S_2(n)$; және

(b) $\varepsilon > 0$ табылып, $S_1(n)^{1+\varepsilon} \leq O(S_2(n))$ орындалады.

Сонда $NSPACE(S_1(n)) \underset{\neq}{\subset} NSPACE(S_2(n))$.

^{S}6. Кейде ұзындықтары бірдей әртүрлі енгізулерде ТМ машинаның уақыттық немесе кеңістіктік пайдалануын ажырата білу үшін, күрделілік талдауын жетілдірген тиімді болады. $G : \Sigma^* \to \mathbb{N}$ үшін анықтаймыз

$$DTIME(G(x)) \overset{def}{=} \{L(M) \mid M \text{ енгізу } x\text{-те } G(x)\text{-тан артық қадам қажет етпейтін детерминирлі ТМ машина}\}$$

$$DSPACE(G(x)) \overset{def}{=} \{L(M) \mid M \text{ енгізу } x\text{-те жұмыс таспаның } G(x)\text{-тан артық ұяшығын қажет етпейтін детерминирлі ТМ машина}\}.$$

$S:\mathbb{N}\to\mathbb{N}$ үшін, әдеттегідей тәсілмен $DTIME(S(n))$ және $DSPACE(S(n))$ анықталады (2-лекцияны қараңыз). Егер $T:\mathbb{N}\to\mathbb{N}$ монотонды және

$$DSPACE(n) \subseteq DTIME(T(n)$$

болса, онда $G(x)\geq|x|$ болатын кез келген $G:\Sigma^*\to\mathbb{N}$ үшін, конструктивті немесе басқаша болса, онда

$$DSPACE(G(x)) \subseteq DTIME(G(x)T(G(x))$$

орындалатынын дәлелдеңіз.

7. Ұзындығы n-ге тең енгізулерде $o(\log n)$-нен аспайтын символ жаза алатын кез келген ТМ машинада, регуляр жыйынға қабылдау-ашық болатынын көрсетіңіз.

8. a) Егер M машина $t(n)$-нен артық символ жаза алмаса және жұмыс жасау

қадамы $T(n)$-нен асып кетпесе, онда $T(n)$ $O(t(n)^2)$ болатынын дәлелдеңіз.

[H]b) $t(n)$-ді пайдаланатын $T(n)$-ге қойылған соңғы шектеу 1-таспалы ТМ машина үшін ең қолайлы болатынын дәлелдеңіз.

9. *k-бас тиекті шектеулі автомат* (k-FA) деп, тек оқуға арналған k енгізу бас тиектері бар 1-таспалы ТМ машинаны айтамыз, мұндағы тиектер оңға және солға жыжый алады, бірақ енгізу жолын тастап кете алмайды.

(a) Осы машиналардың формалды анықтамасын беріңіз, соның ішінде қабылдау-ашық анықтамасын да беріңіз.
(b) Егер тек егер белгілі бір k үшін детерминирлі (бейдетерминирлі) k-FA автоматта жиынға қабылдау-ашық болса, онда ол жиын LOGSPACE (NLOGSPACE)-те жататынын дәлелдеңіз.

10. Егер логикалық формула клоздар конъюкциясының $\ell_1 \vee \ell_2 \vee \ell_3$ формасында болса, онда ол логикалық формула *3-конъюнктивті қалыпты форма* (3CNF) деп аталады, мұндағы

ℓ_i литералдар (Логикалық айнымалылар немесе айнымалылардың теріске шығарылуы). 3CNF-орындалушылық $\leq_m^{\log}$ үшін *NP* толық болатынын дәлелдеңіз.

11. Сызықты-уақыт көптің-жалғызға келтірімділігіне қарасты *NSPACE* (*n*) үшін толық болатын жиынға мысал келтіріңіз. Келтірілген мысалға қатысты мынадай қорытынды жасаңыз: егер тек егер осы жиын *DSPACE* (*n*)-те жатса, онда *NSPACE* (*n*) = *DSPACE* (*n*) болады.

12. Егер $NSPACE\,(n) = DSPACE\,(n)$ болса, онда барлық $S(n) \geq n$ үшін $NSPACE\,(S(n)) = DSPACE\,(S(n))$ болатынын дәлелдеңіз.

13. Егер тек егер
$NSPACE(\log n) \cap \{a\}^* = DSPACE(\log n) \cap \{a\}^*$ болса, онда *NSPACE* (*n*) = *DSPACE* (*n*) болатынын көрсетіңіз.

$^{\text{H}}$14. $DSPACE\,(n) \neq P$ және $DSPACE\,(n) \neq NP$ болатынын дәлелдеңіз.

$^{\text{H}}$15. *NLOGSPACE* үшін келесі екі проблема $\leq_m^{\log}$ - толық болатынын дәлелдеңіз.

(a) Бейдетерминирлі шектеулі автомат берілген. Ол автоматта кез келген жолға қабылдау-ашық па?

(b) Бейдетерминирлі шектеулі автомат берілген. Ол автоматта шексіз көп жолға қабылдау-ашық бола ма?

16. Полиномиалды уақытпен шектелген Тюринг келтірілімділігі $\leq_T^p$ таңбасымен белгіленсін. Сонда

(a) $\leq_T^p$ транзитивті болатынын және

(b) $\leq_m^p$ келтірімділік $\leq_T^p$-ты дамытатынын дәлелдеңіз.

^{**S} 17. Жады магазин көмекші автомат (APDA) - ол жұмыс таспасына қосымша бір стек жалғанған TM. Әрбір қадамда ол бос стекті тексереді, егер ол бос болмаса, онда ең жоғарғы элементті оқиды. Жұмыс таспасында және енгізуде осы сәтте сканерлеу процесі жүріп жатқан символды да оқиды. Осы ақпарат және ағымдағы күй, негізінде ол шектеулі стек алфавитіндегі элементті әрі қарай жылжытуы немесе мүлде алып тастауы мүмкін, жұмыс таспасына символ жазуы мүмкін, енгізу және жұмыс бас тиектерін кез келген бағытта бір ұяшыққа жылжыта алады және жаңа күйге енеді. Ең жоғарғы элементті жоймайынша ол стек элементін оқи алмайды. $S(n)$ жұмыс кеңістікті детерминирлі және бейдетерминирлі APDA-да ғана $DTIME(2^{O(S(n))})$-да жататын жиынға қабылдау-ашық болатынын көрсетіңіз.

18. Егер барлық жолдар $x, y \in \Sigma^*$ үшін $h(xy) = h(x)h(y)$ орындалса, онда бейне $h : \Sigma^* \to \Gamma^*$ гомоморфизм деп аталады. Осы анықтамадан $h(\varepsilon) = \varepsilon$ болатыны шығады. Егер кез келген $a \in \Sigma$ үшін $h(a) \neq \varepsilon$ орындалса, онда гомоморфизм өшірілмейді деп аталады. Егер $A \in \mathcal{C}$ үшін кез келген өшірілмейтін гомоморфизм h-та $\{h(x) \mid x \in A\} \in \mathcal{C}$ орындалса, онда $\mathcal{C}$ жиындарының жиынтығы өшірілмейтін гомоморфизмдермен жабылған деп аталады.

(a) *NP* өшірілмейтін гомоморфизмдермен жабылғанын көрсетіңіз.

(b) Егер тек егер *P=NP* болса, онда *P* өшірілмейтін гомоморфизмдермен жабылған.

19. А қосымша лекцияда айтылғандай, толық дербес реттілік деп дербес $\leq$реттілікті Uжиынын айтады, ол жиынды құрастырудың негізгі заңдылығы: әрбір ішкі жиынның $A \subseteq U$ супремумы (ең кіші жоғарғы шекара) $\sup A$ табылуы қажет. Әрбір ішкі жиынның $A \subseteq U$ келесі қасиеттерге ие болатын

(a) Барлық $y \in A$ үшін $\inf A \leq y$ (A-ның төменгі шекарасы $\inf A$ болады).

(b) Егер барлық $y \in A$ үшін $x \leq y$ болса, онда $x \leq \inf A$ ($\inf A$ ең үлкен жоғарғы шекара), жалғыз инфинимумы (ең үлкен төменгі шекара) бар екенін көрсетіңіз.

**S 20. Егер тек егер жиындық оператор финитарлы болса, онда ол тізбелі-үзіліссіз болатынын дәлелдеңіз.

21. (a) Кез келген толық дербес реттілікте әрбір тізбелі-үзіліссіз оператор монотонды болатынын дәлелдеңіз.

[S] (b) Тізбелі-үзіліссіз болмайтын монотонды жиындық операторға мысал келтіріңіз.

22. Келесі шарттар орындалатын толық дербес реттілік U-ға мысал келтіріңіз: U-да монотонды оператор τ, және префикс нүктелер жиыны $A \subseteq U$ беріліп $\sup A$ префикс нүкте болмайды; сондықтан $\sup A \leq \inf PF_\tau(\sup A)$.

[S]23. Егер τ тізбелі-үзіліссіз болса, онда оның тұйықталу ординал ω-дан аспайтынын дәлелдеңіз, бірақ ол тұжырым барлық монотонды операторға жарамайды.

24. Индуктивтілік пен фундирлілік әрқашан қатар жүреді. Егер X жиынының шексіз төмендетілетін $x_0, x_1, \ldots$ тізбесінде, барлық $i \in \omega$ үшін $x_{i+1} < x_i$ болса, онда X жиынындағы бинарлық қатынас $<$ фундирлі болады. Мысалы, кез келген шектеулі жиындағы ішкі жиындардың қатаң $\subset$ реттілігі сияқты, ω-да қалыпты реттілік $<$ фундирлі болады. 2^ω -да қатаң ішкі жиындар фундирлі болмайды, өйткені $\omega \supset \omega - \{0\} \supset \omega - \{0,1\} \supset \cdots$.
Бинарлы қатынас $<$ үшін индукция принципінің тұжырымы: кез келген жиын $A \subseteq X$ үшін қатынас

$$(\forall x((\forall y\ y < x \to y \in A)) \quad \to \quad \forall x\ x \in A$$

орындалады.
Егер тек егер $<$ фундирленген болса, онда индукция приципі орындалатынын көрсетіңіз. Кез келген аксиоманы пайдалануыңызға болады (А лекцияны қараңыз).

25. (a) $S(n)$ кеңістікте жұмыс жасайтын алтернатив машина M-да, белгілі бір c үшін (M-нен ғана тәуелді, бірақ n-нен тәуелсіз) $c^{S(n)}$ уақыт аралығында есептеу бұтағының түбірлері 0 немесе 1-мен белгілетінін немесе ешқашан белгіленбейтінін дәлелдеңіз.

(б) (а)-ны пайдаланып $ASPACE(S(n)) = \bigcup_c DTIME(c^{S(n)})$ болатынын дәлелдеңіз.

26. Егер тек егер теріске шығару амалы жоқ алтернатив Тьюринг машинасы M-де x-ке қабылдау-ашық болса, онда x енгізуде есептеу бұтағында шектеулі қабылдау-ашық ішкі бұтақ табылатынын дәлелдеңіз. Яғни, есептеу бұтағы шектеулі T ішкі бұтағы табылып, ол бастапқы конфигурацияны келесі шарттармен қамтиды: әрбір $\vee$-конфигурацияның T-да кем дегенде бір мұрагері болады және $\wedge$-конфигурацияның барлық мұрагерлері T-да жатады.

[H]27. Егер сызықты кеңістік $DSPACE(n)$ үшін A жиын $\leq_{\mathrm{m}}^{\log}$-күрделі болса, онда $PSPACE$ үшін де A жиын $\leq_{\mathrm{m}}^{\log}$-күрделі болатынын дәлелдеңіз.

[S]28. $S(n) \geq logn$ және $T(n) \geq n$ болсын. Бейдетерминирлі $S(n)$ -кеңістікті және ТМ машинамен $T(n)$-уақытты шектеулі кез келген қабылдау-ашық жиынға, $S(n)logn\ T(n)$ кеңістікті детерминирлі ТМ-де қабылдау-ашық болуы мүмкін екенін дәлелдеңіз.

Басқаша айтсақ,

$$STA(S(n),T(n),\Sigma 1) \subseteq \mathrm{STA}(S(n)\log T(n),*,0).$$

Конструктивтілікті ұмытпаңыз.

[S]29. $1 \leq i \leq n$ болғанда берілген детерминирлі шектеулі автоматтар жиыны M_i үшін тұжырымның $\bigcap_{i=1}^{n} L(M_i) = \varnothing$ ақиқат екендігін анықтауды шешумен айналысатын проблема, $PSPACE$-толық болатынын көрсетіңіз.

[S]30. Криптожүйенің қауіпсіздігі біржақты функцияның бар болуына негізделеді. Осы проблеманың мақсатына жету үшін $f:\Sigma^* \to \Sigma^*$ бейненің ұзындығын сақтайтын біржақты функция полиномиалды уақытта есептелетін болсын және ол детерминирлі полиномиалды уақытта керіленбесін деп ұйғарамыз. Мұндағы керіленудің мағынасы мынада: y берілген кезде не $f(x) = y$ болатындай қандай да бір x-ті құрастырамыз немесе ондай қасиетпен x табылмайды деп айтамыз. Егер тек егер $P \neq NP$ болса, онда бір жақты функция табылатынын дәлелдеңіз.

[H]31. Төменде келтірілетін үш проблеманың біреуі co-*NLOGSPACE* үшін $\leq_{\mathrm{m}}^{\log}$-толықтылық, ал қалған екеуі *PSPACE* үшін $\leq_{\mathrm{m}}^{\log}$-толық болады. Ажыратыңыз, қайсысы қайсыған жатады? Дәлелдемені келтіріңіз.

(a) Егер регуляр өрнек α берілсе, онда $L(\alpha)=\varnothing$ ақиқат бола ала ма?

(b) Егер регуляр өрнек α берілсе, онда $L(\alpha)=\Sigma^*$ ақиқат бола ала ма?

(c) Егер екі регуляр өрнектер α мен β берілсе, онда $L(\alpha)=L(\beta)$ ақиқат бола ала ма?

32. [s](a) Егер тек егер полиномиалды уақытта есептелетін детерминирлі бинарлы предикат *R* бар болса және

$$A = \{x \mid \exists y \left|y\right| \leq \left|x\right|^c \wedge R(x,y)\}$$

орындалатын тұрақты c табылатын болса, онда *A* жиын *NP*-да жататынын дәлелдеңіз.

(b) 10.2-теореманың жалпыланған түрін дәлелдеңіз: егер тек егер полдиномиалды уақытта есептелетін детерминирлі (*k+1*)-арлы предикат бар болса және

$$A = \{x \mid \exists^{|x|^c} y_1 \forall^{|x|^c} y_2 \exists^{|x|^c} y_3 \cdots \mathrm{Q}^{|x|^c} y_k R(x, y_1, \ldots , y_k)\}.$$

орындалатын тұрақты c табылатын болса, онда A жиынның Σ_k^{p} -да жататынын дәлелдеңіз.

Шектеулі кванторлар $\exists^t$ және $\forall^t$ 10-лекцияның соңында анықталған.

33. $S(n) \geq \log n$ болсын. Сонда

$$\bigcup_k STA(S(n),*,\Sigma k) \cup \mathrm{STA}(S(n),*,\Pi k) = NSPACE(S(n))$$

орындалатынын дәлелдеңіз.

34. Σ_k^{p}-да оракулды полиномиалды уақытпен шектеулі, оракул детерминирлі машинада қабылдау-ашық $\Delta_{k+1}^{\mathrm{p}} = P^{\Sigma_k^{\mathrm{p}}}$ орындалатын жиындар үйірін анықтаңыз.

(a) $\Sigma_k^{\mathrm{p}} \cup \Pi_k^{\mathrm{p}} \subseteq \Delta_{k+1}^{\mathrm{p}} \subseteq \Sigma_{k+1}^{\mathrm{p}} \cap \Pi_{k+1}^{\mathrm{p}}$ орындалатынын көрсетіңіз.

(b) Тек бір ғана орындалатын меншіктеулі логикалық формулалар жиыны Δ_2^{p}-да жататынын көрсетіңіз.

(c) Δ_2^{p} үшін $\leq_{\mathrm{m}}^{\log}$-толық проблеманы қойып шығыңыз және оның толық екенін дәлелдеңіз.

35. Егер тек егер PH-тің $\leq_{\mathrm{m}}^{\log}$-толық жиыны болса, онда оның күйрейтінін дәлелдеңіз.

[H]36. *NLOGSPACE*-те тұрақты тереңдікті схемалар үшін тізбе мәндерінің проблемасы $\leq_{\mathrm{m}}^{\log}$-толық болатынын дәлелдеңіз.

37. Логикалық шешім диаграмасы (ЛШД) бір қайнарлы (источник) және екі тұғылды (сток) бағытталған ациклды граф; оның біреуі 0-мен, ал екіншісі 1-мен белгіленеді және белгілі бір логикалық айнымалы үшін барлық бейтұғырлы түйіндердің дәл екі шығатын қабырғасы бар, оның біреуі x деп белгіленсе, екіншісі $\overline{x}$ арқылы белгіленеді. σ ақиқаттықты меншіктеуде ЛШД -ның мәні тұғыл түйіндегі белгі болып табылады, ол қайнардан басталып тұғылмен аяқталатын уникалды σ-қосқыш жолда жатады, мұнда егер $\sigma(\ell)=1$ болса, онда ℓ литералды қабырға σ-қосқыш болады. ЛШД үшін күрделілік түрін анықтаңыз:

(a) берілген σ үшін мәнді анықтау?

(b) орындалушылық?

38. Дизъюнктивті нормалды формада n-ші схеманың жазылуының арқасында әрбір шешілушілік проблемасында өлшемі $O(n2^n)$ -нен аспайтын логикалық схемасының үйірі бар болады. Әрбір шешілушілік проблемасында табылатын логикалық схеманың өлшемі аспайды:

[H](a) $O(2^n)$-нен,

*(b) $O(2^n/n)$-нен.

**39. Тіксызықты лабиринт ([18]) деп шексіз шахмат тақтасының байланысты ішкі жиынын айтамыз. Яғни, ол байланысты бағытталмаған граф, оның төбелері реттелген бүтін сандар жұбын құрастырады және оның қабырғалары $((x, y),(x, y+1))$ немесе $((x, y),(x+1, y))$ формасында болады. Тіксызықты лабиринттер үшін MAZE проблемасы детерминирлі логарифмдік кеңістікте шешілетінін көрсетіңіз.

[H]40. Берілген бағытталған ациклді (V, E) графта топологиялық сорттауды табу үшін *NC* алгоритмді келтіріңіз. Яғни, *V*-ның $<$ толық реттілігін табыңыз, бұл реттілік егер *uEv* болса, онда $u < v$ болады мағынасында *E*-ні кеңейтеді.

[S]41. (a) Бірнеше шығу сымдары бар *NC* схемалар үйірінің көмегімен кез келген детерминирлі logspace түрлендіргішті симуляция жасауға болатынын көрсетіңіз.

(b) Мәндер проблемасының схемасының (CVP) *P* үшін $\leq_m^{\log}$-толық болатынын 6.1-теоремада көрсеткен болатынбыз. Осыдан және (a) бөліктен егер тек егер *P* = *NC* болса, онда $\mathrm{CVP} \in NC$ болады деген тұжырымды шығарыңыз.

[H]42. Ешқайсысы нөлге тең емес бүтін екі *m* және *n* сандарының ең үлкен ортақ бөлгішін (ЕҮОБ) есептейтін келесі Standard ML программаны қарастырайық.

```
fun Euclid(m : int, n : int) : int * int * int =
   if n = 0 then (1,0,m)
   else let
     val q = m div n
     val r = m mod n
     val (s,t,g) = Euclid(n,r)
   in
     (t,s - t * q,g)
   end
```

Бізге белгілі бүтін сандарды бөлу амалын пайдаланған *m*-ді *n*-ге бөлу процесі, мұндағы div бөліндіні, ал mod қалдықты есептейді. Программа өнімі (*s, t, g*) үштігі болатынын дәлелдеңіз, мұндағы g саны *m* мен *n*-нің ЕҮОБ-і, және $sm + tn = g$ орындалатын етіп алынған *s, t* бүтін сандар.

Н43. a және n бүтін оң сандар болсын. Келесі тұжырымдар эквивалентті екендігін дәлелдеңіз.

(i) модуль n-де a-ның реті табылады; яғни $a^m \equiv 1 \pmod{n}$ орындалатын m табылады.

(ii) n-ге қатысты a жай сан; яғни ЕҮОБ$(a,n)=1$.

(iii) n модельде a керіленеді; яғни $ab \equiv 1 \pmod{n}$ орындалатын b, $1 \le b \le n-1$ табылады.

44. 14.1-леммадағы BPP мен RP-ге ұқсастырып, IP мен PCP үшін келесі күшейту лемманы дәлелдеңіз. Егер $r(n)$ кездейсоқ битті және $q(n)$ сұратымды пайдаланатын L-дің IP (сәйкес PCP) протоколы бар болса, онда кез келген $\varepsilon > 0$ үшін $kr(n)$ кездейсоқ битті және $kq(n)$ сұратымды пайдаланатын L-дің IP (сәйкес PCP) протоколы бар болады және оның ε-мен шектелген қателік ықтималдығы болады, мұндағы k ол $O(-\log\varepsilon)$. PCP жағдайда, бұл мынаны білдіреді:

(i) Егер $x \in L$ болса, онда $\Pr\big((P,V)$-да x-ке қабылдау-ашық)=1,

(ii) Егер $x \notin L$ болса, онда кез келген P' үшін, $PR((P',V)$-да x-ке қабылдау-ашық$\big) \le \varepsilon$, және IP жағдай үшін:

(i) Егер $x \in L$ болса, онда $PR((P',V)$-да x-ке қабылдау-ашық$) \ge 1-\varepsilon$,

(ii) Егер $x \notin L$ болса, онда кез келген P' үшін, $PR((P',V)$-да x-ке қабылдау-ашық$\big) \le \varepsilon$.

45. Бұл жаттығуда 18.2 лемманың дәлелдемесін аяқтаймыз. Енді n айнымалылы $x_1, \dots, x_n$ және m клозды 3CNF логикалық формула B болсын, мұнда әрбір клоз әртүрлі айнымалы дәл үш литералды қамтиды. S_i мен S екеуі 18.2-лемма дәлелдемесіндегідей болсын. Сол дәлелдемеде қарастырылғандай, $\mathcal{E}(S) = 7m/8$. Сараң алгоритмнен алынған $x_1, \dots, x_n$-ге ақиқатты меншіктеу $a_1, \dots, a_n \in \{0,1\}$ болсын. Ақиқатты кездейсоқ меншіктеу $r_1, \dots, r_n$ үшін E_k оқиға

$$E_k \overset{\text{def}}{=} \bigwedge_{i=1}^{k} r_i = a_i$$

болсын. Сонда $\mathcal{E}\left(S \mid E_{k-1} \wedge r_k = a_k\right) \ge \mathcal{E}\left(S \mid E_{k-1} \wedge r_k = \overline{a}_k\right)$, өйткені a_k-ны сараң алгоритм арқылы таңдап алдық.

(a) $\mathcal{E}\left(S \mid E_k\right)$-ді тиімді таңдап алу жолын көрсетіңіз.
(b) Дәлелдеңіз

$$\mathcal{E}\left(S \mid E_{k-1}\right) = \mathcal{E}\left(S \mid E_{k-1} \wedge r_k = a_k\right) \cdot \Pr\left(r_k = a_k \mid E_{k-1}\right) + \mathcal{E}\left(S \mid E_{k-1} \wedge r_k = \overline{a}_k\right) \cdot \Pr\left(r_k = \overline{a}_k \mid E_{k-1}\right).$$

(c) (b)-дан, $\mathcal{E}\left(S \mid E_{k-1}\right) \le \mathcal{E}\left(S \mid E_k\right)$ болады деген қорытынды шығарыңыз.
(d) (c)-ны пайдаланып сараң меншіктеу $a_1, \ldots, a_n$, кем дегенде $7m/8$ клозды қанағаттандыратынын көрсетіңіз.

[H]46. Егер $P = NP$ болса, онда
(a) MAX-3SAT
(b) MAX-CLIQUE
проблемаларды полиномиалды уақыт дәлдігімен шешуге болатынын көрсетіңіз (18-лекцияны қараңыз).

47. 18.3-теореманың дәлелдемесін аяқтап шығыңыз, ол үшін сол дәлелдемедегі конструкциялар жайлы келесі тұжырымдарды пайдаланыңыз:
(a) Егер $a \neq b$ болса, онда (y, a) және (y, b) сәйкестенбейтінін дәлелдеңіз, сондықтан $((y, a), (y, b))$ графтың қабырғасы бола алмайды.
[H](b) Егер $x \in L$ болса, онда G-дың максималды кликасының өлшемі n^c болатынын дәлелдеңіз, осымен бірге, егер $x \notin L$ болса, онда G-дың максималды кликасының өлшемі αn^c-дан қатаң кем болатынын дәлелдеңіз.

48. Белгілі бір жай сан p үшін $\mathbb{F} = \mathbb{Z}_p$ болсын. Келесі шарттарды анықтайтын $PCP(n^3, 1)$ протоколды беріңіз: берілген оракулдер $f : \mathbb{F}^n \to \mathbb{F}$ және $f : \mathbb{F}^{n^3} \to \mathbb{F}$ үшін, белгілі бір $a \in \mathbb{F}^n$ үшін f функция $r \mapsto r \bullet a$ формадағы функцияға жуық па және h функция $t \mapsto t \bullet \left(a \otimes a \otimes a\right)$ формадағы функцияға жуық па (20-шы лекцияны қараңыз).

49. Сигнатурадағы шектеудің көмегімен бірінші ретті теория үшін тривиалды анықтамасын ойдан табыңыз. Сіздің анықтамаңыздағы тривиалдылықты пайдаланып дәлелдеңіз:

HS (a) Әрбір бейтривиалды теория PSPACE-күрделі болатынын дәлелдеңіз.

(b) Әрбір тривиалды теория полиномиалды уақытта шешіледі.

HS50. Егер ойын тақтасының әрбір позициасынан тек шектеулі көп мүмкін келесі жүрістер табылса, онда ойын Ehrenfeucht-Fraissé шектеулі деп аталады. Әрбір бірінші ретті теория шектеулі Ehrenfeucht-Fraissé ойынмен сипатталатынын дәлелдеңіз. Әрбір бірінші ретті теорияның шешілетінін осы неге көрсетпейді?

51. Детерминирлі Бучи автоматындағы қабылдау-ашық шексіз жолдар жиыны, бірігу және қиылысуға қатысты жабық болатынын дәлелдеңіз.

52. Бейдетерминирлі Бучи автоматы үшін күрделілік проблемасы қандай болады? Дәлелдеме келтіріңіз.

53. Бучи автоматының барлық жолға

(a) детерминирлі

*H(b) бейдетерминирлі

қабылдау-ашық болуының күрделілігі қандай? Дәлелдеме келтіріңіз.

54. Бучи автоматында қабылдау-ашық әрбір жиын AB^{ω} формадағы жиындардың шектеулі бірігуі болатынын дәлелдеңіз, мұндағы *A* мен *B* регуляр жиындар. Бұл жерде B^{ω} арқылы $w_0 w_1 w_2 \ldots$ формадағы шексіз сөз белгіленген, мұнда барлық $i \geq 0$ үшін $w_i \in B - \{\varepsilon\}$.

55. Жиындардың тең қуаттылығын, яғни предикат

$\varphi(A,B) \overset{def}{\Leftrightarrow} A$ және *B* элементтер саны тең болады,
S1S-те өрнектелмейтінін дәлелдеңіз.

56. Шексіз сөзді автомат үшін келесі екі қабылдау-ашық болатын шарттарды қарастырыңыз.

(a) *Streett шарты*: Рабиннің қабылдау-ашық шарты сияқты, шектеулі жұптар (G_i, R_i), $1 \leq i \leq k$ жиыны табылады, мұндағы

G_i мен R_i - Q-дың ішкі жиындары. Егер барлық i үшін $\mathrm{IO}(\sigma) \cap G_i \neq \varnothing$ қатынас $\mathrm{IO}(\sigma) \cap R_i \neq \varnothing$-ны меңзесе, онда жүгіртпе σ-да қабылдау-ашық.

(b) Жұптық шарт: Автомат күйі нөмірленген $\{0,1,\ldots,n-1\}$ деп ұйғарыңыз. Егер шексіз жиі байқалатын ең кіші нөмірленген күй жұп болса, онда жүгіртпе σ-да қабылдау-ашық.

Бейдетерминирлі автомат үшін осы екі қабылдау-ашық болу шарттары 25 және 26-лекцияларда қарастырылған басқа екі қабылдау-ашық шарттарға эквивалент болатынын көрсетіңіз.

57. [S](a) Барлық $z > 0$ үшін $(1-1/z)^z \leq e^{-1}$ болатынын және $z \to \infty$ ұмтылғанда $(1-1/z)^z$ тізбегінің шегі e^{-1}-ге ұмтылатынын көрсетіңіз, мұндағы $e = 2.718\ldots$ саны натурал лигарифмнің негізі болып табылады.

(b) Барлық $z > 0$ үшін $(1+1/z)^z \leq e$ болатынын және $z \to \infty$ ұмтылғанда $(1+1/z)^z$ тізбегінің шегі e-ге ұмтылатынын дәлелдеңіз.

58. $NP^A \neq \text{co-}NP^A$ орындалатындай рекурсивті оракул A-ны құрастырыңыз.

59. (Кай және Огихара [25]) Бұл жаттығуда Маханей (29.2-теорема) теоремасының алтернативті дәлелдемесін береміз. S жиын сиретілген NP-күрделі болсын. Ұзындығы n логикалық формула φ және φ-дің айнымалыларына ақиқатты меншіктеу $t \in \{0,1\}^n$ үшін t-да φ-ді бағалау кезінде алынған $\varphi(t)$ ақиқаттылық мәнін белгілесін. Ұзындығы бірдей ақиқатты меншіктеуде $\leq_{\mathrm{lex}}$ арқылы лексографикалық реттілік белгіленген болсын. Жиынды анықтаймыз

$$E \stackrel{def}{=} \{(\varphi, s) \mid \exists t \; s \leq_{\mathrm{lex}} t \wedge \varphi(t) = 1\}.$$

Егер $(\varphi, t) \in E$ және $s \leq_{\mathrm{lex}} t$ болса, онда $(\varphi, s) \in E$ болатынын байқаймыз. E үшін полиномиалды уақытты шешілімділік процедурасын белгілейміз.

(a) E-нің NP-толық болатынын көрсетіңіз. Сондықтан E үшін полиномиалды уақытты шешілімділік процедурасы $P = NP$-ны меңзейді.

(b) Уақытты шектеуі n^c болатын, E-ден S-ке полиномиалды уақытты көпмәнді келтіру σ болсын. $|S \cap \Sigma^{\le n^c}| \le n^d$ қатынасы орындалатындай n^d полиномы табылатынын дәлелдеңіз.

Ұзындығы n логикалық формула φ болсын. φ-дің айнымалыларына барлық ақиқаттықты меншіктеу жиыны $A_0 = \{0,1\}^n$ болсын. Біз өлшемі кемімелі ішкі жиындар тізбегін $A_0 \supseteq A_1 \supseteq A_2 \supseteq \cdots$ құрастырамыз, және келесі инвариантты сақтаймыз: егер φ орындалса, онда A_i қанағатандырарлық меншіктеуді қамтиды.

Бұл жерде A_i-ден A_{i+1}-ді қалай алуға болатынын көрсетеміз. $A_i = \{t_0,\dots,t_{m-1}\} \subseteq \{0,1\}^n$ болады деп ұйғарайық, мұнда $t_0 \le_{\text{lex}} \cdots \le_{\text{lex}} t_{m-1}$ және $m \ge n^d + 1$ (мұндағы d (b) бөліктегі d-ның дәл өзі). Енді

$$J \stackrel{def}{=} \left\{ \lfloor km/(n^d+1) \rfloor \mid 0 \le k \le n^d \right\} \subseteq \{0,1,2,\dots,m-1\}$$

болсын. Бұл A_i элементтерінің n^d+1 индексті бірқалыпты үлестірімінің жиыны.

(c) Барлық k үшін $\lfloor km/(n^d+1) \rfloor < \lfloor (k+1)m/(n^d+1) \rfloor$ орындалатынын көрсетіңіз, сондықтан J әртүрлі n^d+1 элементті қамтиды.

Әрбір $j \in J$ үшін $\sigma(\varphi,t_j)$-ды есептейміз. Егер мұнда $i<j$ және $\sigma(\varphi,t_i)=\sigma(\varphi,t_j)$ болатын $i,j \in J$ табылса, онда A_i-дегі интервалды $\{t_j \mid i \le k < j\}$ жоямыз. Егер, екінші жағынан, $j \in J$ үшін барлық $\sigma(\varphi,t_j)$ әртүрлі болса, онда A_i-дегі соңғы интервалды $\{t_k \mid k \le n^d m/(n^d+1)\}$ жоямыз.

(d) Барлық жағдайда инвариант сақталатынын аргументтеңіз.

(e) Барлық жағдайда A_i-ден $|A_i|/(n^d+1)$-ден аз емес элемент жоямыз. 57(a) есепті пайдаланып көрсетіңіз: келесі саннан көп емес қадамнан $\left(n^d+1\right)\left(n\ln 2 - \ln\left(n^d+1\right)\right) = O\left(n^{d+1}\right)$ кейін бізде жеткілікті аз жиын қалады, сондықтан қалған ақиқатқа меншіктеулерде бірден φ-ді бағалай аламыз.

(f) Осы конструкцияда өндірілген A_i жиынның, жалпылап айтсақ, өлшемі экспоненциалды болады. Дегенмен, олар тиімді өрнектеліп, жоғарыда сипатталған процедура полиномиалды уақытта орындалуы мүмкін. $|A_i|$ мен бейне $j \mapsto t_j$-ді тиімді есептеуге мүмкіндік беретін, A_i жиынының өрнегін келтіріңіз.

60. (a) Еске түсірейік, егер квадратты матрицаның барлық бағаны сызықты тәуелсіз болса, онда квадратты матрица сингулярлы емес болатын. $\mathbb{GF}_q$-да тура $\Pi_{i=0}^{n-1}(q^n - q^i)$ сингулярлы емес $n \times n$ матрицалар табылатынын дәлелдеңіз.

*H(b) $\mathbb{GF}_q$-да барлық сингулярлы емес матрицалары тең ықтималды болатын, кездейсоқ $n \times n$ сингулярлы емес матрицаны қалай тиімді құрастыруға болатынын көрсетіңіз.

61. Егер n өлшемді векторлық кеңістік V-ның ішкі кеңістігі E болса, онда $\dim E$ және $\operatorname{codim} E$ арқылы сәйкес E-нің өлшемі мен ко-өлшемі (n минус өлшем) белгіленеді. $\operatorname{codim} E \cap F \leq \operatorname{codim} E + \operatorname{codim} F$ орындалатынын көрсетіңіз.

62. Лемма F.1-де тағайындалған 3/4 төменгі шек барлық n үшін сығылған болатынын дәлелдеңіз.

63. (Пападимитриу және Зачос [93]) $\oplus P$ (“P-ның жұптығы”) класы полиномиалды уақытты бейдетерминирлі *TM* M машина табылатын A жиынының класы ретінде анықталады, мұнда егер тек егер енгізу x-те M-нің қабылдау-ашық есептеу жолдарының саны тақ болса, онда $x \in A$ болады. Эквивалентті, M машинаны алтернативті машина ретінде қарастыра аламыз, ол тармақталу түйіндерде мұрагерлердің белгілерінің қосындасын mod–2-де есептейді. G-лекцияда $\oplus P$-ның басқаша сипаттамасы берілген. $\oplus P^{\oplus P} = \oplus P$ болатынын дәлелдеңіз.

64. $\#P$-ның келесі екі сипаттамасы эквивалентті болатынын көрсетіңіз.

(a) Егер тек егер функция $f:\{0,1\}^* \to \mathbb{N}$ $\#P$-да жатса, онда барлық x үшін $A \in P$ және $k \geq 0$ табылып, $f(x) = |W(n^k, A, x)|$ орындалады.

(b) Егер тек егер функция $f:\{0,1\}^* \to \mathbb{N}$ $\#P$-да жатса, онда полиномиалды уақытты бейдетерминирлі TM M табылып, енгізу x-те M-нің қабылдау-ашық есептеу жолдарының саны $f(x)$ -ке тең болады.

65. Тод теоремасының (Теорема G.1) дәлелдемесінен, біз білетін

дүниелер, ол $h^m(z)$ полином келесі қасиеттерді қанағаттандырады:

$$z \text{ тақ } \Rightarrow h^m(z) \equiv -1 (\operatorname{mod} 2^{2^m})$$
$$z \text{ жұп } \Rightarrow h^m(z) \equiv 0 (\operatorname{mod} 2^{2^m}),$$

мұндағы

$$h(z) \stackrel{\text{def}}{=} 4z^3 + 3z^4$$

$$h^0(z) \stackrel{\text{def}}{=} z$$

$$h^{m+1}(z) \stackrel{\text{def}}{=} h(h^m(z)).$$

Шынымен, осылай болатынын көрсетіңіз.

66. Лемма G.2-нің келесі клоздарын дәлелдеңіз. Егер $\mathcal{C}$ $\leq_{\mathrm{T}}^{\mathrm{p}}$-мен төменнен жабылса, онда

[S](ii) $\Pi^{\log} \cdot \oplus \cdot \mathcal{C} \subseteq \oplus \cdot \mathcal{C}$;
*[HS](iii) $\oplus \cdot BP \cdot \mathcal{C} \subseteq BP \cdot \oplus \cdot \mathcal{C}$;
[H](iv) $BP \cdot BP \cdot \mathcal{C} \subseteq BP \cdot \mathcal{C}$;
[H](v) $\oplus \cdot \oplus \cdot \mathcal{C} \subseteq \oplus \cdot \mathcal{C}$;
[H](vi) $BP \cdot \mathcal{C}$ и $\oplus \cdot \mathcal{C}$ $\leq_{\mathrm{T}}^{\mathrm{p}}$-мен төменнен жабылған.

67. Барлық полиномиалды уақытта саналатын функциялардың класы $\#P$ болсын. Бұл кластардың формалды анықтамасы G лекцияда беріледі. Келесі нүктелі амалдарда $\#P$ тұйық болатынын дәлелдеңіз.

(a) Қосу: егер, $f, \mathsf{g} \in \#P$ болса, онда $f + \mathsf{g} = \text{ё}x.f(x) + \mathsf{g}(x) \in \#P$.
(b) Көбейту: егер, $f, \mathsf{g} \in \#P$ болса, онда $f \cdot \mathsf{g} = \text{ё}x.f(x)\mathsf{g}(x) \in \#P$
*(c) Егер f элемент $\#P$-да жатса, онда барлық тұрақты d үшін $\text{ё}x.f(x)^{|x|^d}$ болады.
*[S](d) Енгізу $x \in \{0,1\}^*$-те z детерминант емес z-пен полином беретін және оң бүтін мәнді коэффициенттермен $\mathbb{N} \to \mathbb{N}$ полиномиалды функцияны өрнектеген функция $\mathsf{g} : \{0,1\}^* \to \mathbb{N}[z]$ болсын. Ұйғарамыз, $|x|$-те $\mathsf{g}(x)$-тің дәрежесі полиномнан көп болмайды деген мағынада g полиномиалды уақытта есептеледі. $\mathsf{g}(x)$-тің коэффициенттері полиномиалды көп биттен аспайтын екілік жүйеде өрнектелуі мүмкін және z^i-дің коэффициенті x пен i-ден полиномиал уақытта алынуы мүмкін. Егер f элемент $\#P$-да жатса, онда $\text{ё}x.\mathsf{g}(x)(f(x))$-те сонда жататынын көрсетіңіз.

68. (Пападимитриу [92]). Кейбір есептеулер енгізу x-те нәтиже $f(x)$ -ті өндіреді, ол ең жаман деген жағдайда $|x|$ -тен әлдеқайда көп. Осындай проблемалар үшін $|x|+f(x)$-те полиномиалды болатын алгоритм қолда болғаны дұрыс, онымен берілген x пен y-ті пайдаланып, $f(x)=y$ ақиқат па соны шешуге болады. Осындай проблемалар класы нәтижелі полиномиалды уақыт деп аталады.

[H](a) Берілген регулярлы өрнектен, минималды күйлі эквивалентті детерминирлі шектеулі автомат өндіретін функцияны қарастырыңыз. Егер тек егер бұл функция нәтижелі полиномиалды уақытта жатса, онда $P = PSPACE$ болатынын дәлелдеңіз.

(b) Нәтижелі полиномиалды уақытта эквивалентті детерминирлі автомат өндіруге болмайтынын дәлелдеңіз, ол автомат $P = PSPACE$ болғанға дейін минималды күйлі автоматқа қарағанда полиномиалды көп күйге ие болады.

69. (a) Қатынасты дәлелдеңіз

$$(z \wedge u) \vee (\overline{z} \wedge v) \quad \equiv \quad (z \rightarrow u) \wedge (\overline{z} \rightarrow v) \quad .$$

(b) Басқа да мүмкін айнымалылармен бірге жалғыз оң енетін логикалық айнымалы x-ке тәуелді логикалық функция $\psi(x)$ болсын. Мұндағы оң дегеніміз жұп теріске шығару шеңберінде айтылған. Көрсетіңіз

$$\psi(x_0 \vee x_1) \; = \; \psi(x_0) \vee \psi(x_1)$$

$$\psi(x_0 \wedge x_1) \; = \; \psi(x_0) \wedge \psi(x_1)$$

орындалатынын, мұнда x_0 мен x_1 $\psi(x)$ -те кездеспейді.

[HS](c) Кез келген Σ_k оракул машина, барлық оракул сұратымды есептеудің соңында жасайтын оракул машинамен тиімді симуляция жасайтынын көрсетіңіз.

[H]70. 69(a) жаттығудағы теңдеудің сол жағындағы (немесе оң жақтағы эквивалент формула) логикалық формула $\mathrm{excl}(z,u,v)$ арқылы белгіленген болсын. Логикалық x айнымалылы және басқа да мүмкін айнымалылы кез келген логикалық формула $\varphi(x)$ болсын. Қатынасты дәлелдеңіз

$$\varphi(\mathrm{excl}(z,u,v)) \quad \equiv \quad \mathrm{excl}(z,\varphi(u),\varphi(v))$$

71. Логикалық формула φ-дің $M \leq \varphi$ орындалатын $\leq$-максималды M-нің термі минтерм деп аталады. Егер φ CNF формула болса, онда егер тек егер φ-дің M минтермі болса, онда φ-дің әрбір клозынан жоқ дегенде бір литералды M қамтиды және ешқандай M-нің жақындайтын ішкі термінде бұл қасиет болмайды. Осы тұжырымды дәлелдеңіз.

72. φ логикалық формула болсын. Барлық φ минтермдерінің дизъюнкциясы φ-ге эквивалентті DNF болатынын көрсетіңіз.

73. Егер F-ті жабатын G_i өзара болдырмайтын (тәуелсіз) болса, онда

$$\begin{aligned}\Pr(E \mid F) &= \sum_i \Pr(E \mid G_i \wedge F) \cdot \Pr(G_i \mid F) \\ &\leq \max_i \Pr(E \mid G_i \wedge F)\end{aligned}$$

орындалатынын дәлелдеңіз.

74. Тұжырымды дәлелдеңіз
$$\Pr(E \wedge F \mid G) = \Pr(E \mid F \wedge G) \cdot \Pr(F \mid G) \quad .$$

75. Егер $E_0 \wedge F_0$ мен $E_1 \wedge F_1$ тәуелсіз және F_0 мен F_1 тәуелсіз болса, онда
$\Pr(E_0 \wedge E_1 \mid F_0 \wedge F_1) = \Pr(E_0 \mid F_0) \cdot \Pr(E_1 \mid F_1)$ орындалатынын дәлелдеңіз.

76. (a) Егер G оқиға E және $E \wedge F$ екеуінен де тәуелсіз болса, онда
$$\Pr(E \mid F \wedge G) = \Pr(E \mid F)$$
болатынын дәлелдеңіз.

(b) Келесі тұжырымның қате екендігін көрсететін контр-мысал келтіріңіз. Егер G оқиға E және F екеуінен де тәуелсіз болса, онда
$$\Pr(E \mid F \wedge G) = \Pr(E \mid F)$$
болады.

77. (a) Тұжырымды дәлелдеңіз

$$\Pr(E \mid F) \leq \Pr(E) \quad \Leftrightarrow \quad \Pr(F \mid E) \leq \Pr(F)\,.$$

(b) Біршама жалпыланған тұжырымды дәлелдеңіз:

$$\Pr(E \mid F \wedge G) \leq \Pr(E \mid F) \quad \Leftrightarrow \quad \Pr(F \mid E \wedge G) \leq \Pr(F \mid G)\,.$$

78. Егер $M \leq \psi \leq \varphi$ болса және φ-дің минтермі M болса, онда ψ -дің де минтермі M болатынын дәлелдеңіз.

79. H.2-лемма (iii) және (iv)-ті дәлелдеңіз: егер ρ дербес баға және $\rho(W) = W$ орындалатын W айнымалылар жиыны, сонда

(a) $\rho(\varphi)^{-W} = \rho(\varphi^{-W})$
(b) егер тек егер $\rho(\varphi) = 1$ болса, онда $\rho(\varphi^{-W}) = 1$ болады.

[S]80. Анықтамалар және нотациялар үшін Лекция H-ты қараңыз. X айнымалыларда φ CNF формула болсын және $\rho : X \to X \cup \{0,1\}$ дербес бағалау болсын. ρ-ның көмегімен тағайындалмаған айнымалылар жиыны Y болсын және Y-тегі терм N болсын. Егер $\rho(\varphi)$-дің минтермі N болса, онда $N = \rho(M)$ орындалатын φ-дың минтермі M табылады. Осы тұжырымды дәлелдеңіз.

[S]81. Анықтама және нотацияны қайталау үшін H-лекцияны қараңыз. Кездейсоқ дербес ρ бағаны қарастырамыз, ол әрбір айнымалыға $\frac{1}{2}(1-\rho)$ ықтималдықпен тәуелсіз түрде 0 немесе 1-ді тағайындайды, немесе p ықтималдықпен тағайындаусыз қалдырады. φ және ψ үшін $s \geq 0$ болғанда, келесі шарттар орындалатындай етіп

$$\begin{aligned}&\Pr(\exists M \in \mathrm{m}(\rho(\varphi)) \ \ |M| \ \geq \ s) \\ &\quad < \ \Pr(\exists M \in \mathrm{m}(\rho(\varphi)) \ |M| \ \geq \ s \ \mid \ \rho(\psi) = 1)\end{aligned}$$

CNF формулалар келтіріңіз. Сондықтан шартты формуланың 1-ге эквивалентті болып қалыптасуы, үлкен минтерм бар болуы мүмкіндігін бұл формула көп төмендетпейді.

82. (a) Нақты мәнді кездейсоқ айнымалылар X_i болсын, мұндағы $1 \le i \le n$. Олардың көбейтінділерінің күтілетін мәні күтілетін мәндердің көбейтіндісіне тең болатынын көрсетіңіз.

(b) Тәуелсіз деген ұйғарым жоқ болса (a) бөлік қате болып қалатынын көрсетіңіз.

83. Марков шектеуін (I.1) дәлелдеңіз: орта мәні $\mu = \mathcal{E}X$ болатын теріс емес кездейсоқ шама X үшін

$$\Pr(X \ge k) \quad \le \quad \mu/k$$

орындалады.

84. Чебышев шектеуін (I.2) дәлелдеңіз: орта мәні $\mu = \mathcal{E}X$ және стандартты ауытқуы $\sigma = \sqrt{\mathcal{E}((X-\mu)^2)}$ болатын кездейсоқ шама X үшін, кез келген $\delta \ge 1$ -де

$$\Pr(|X - \mu| \;\ge\; \delta\sigma) \quad \le \quad \delta^{-2}$$

орындалады.

85. Жетісті болу ықтималдығы p болатын Бернуллидің n тәжірибесінің стандартты ауытқуы $\sqrt{np(1-p)}$ болатынын дәлелдеңіз.

86. (I.2) Чебышев шектеуі (a) және (I.7) Чернов шектеуі (b) екеуін пайдаланып, тәжірибенің жетісті болу ықтималдығы p болатын Бернуллидің n тәжірибесінде, тәжірибенің жетісті болу саны орта мәннің жартысынан кіші болатын оқиғаның ықтималдығын табыңыз.

87. Чернов шектеулерінің (I.3), (I.4) және (I.6) алтернативті формалары өзара эквивалентті екенін дәлелдеңіз.

88. (I.7)- (I.9) Чернов шектеулерін төменгі құйрық үшін дәлелдеңіз.

89. (I.10) Чернов шектеуін Пуассон тәжірибесінің қосындысы үшін орта мән μ болғанда

$$\Pr\left(X < (1-\delta)\mu\right) \; < \; e^{-\delta^2\mu/2}$$

болатынын дәлелдеңіз, мұндағы $0 \le \delta \le 1$.

90. $\mathbb{N}$ ауқымында төмендегі конструкцияларды қамтитын $x, y, \ldots$ айнымалылы қарапайым программалау тілін қарастырамыз.

(i) Қарапайым меншіктеу:	$x := 0 \quad x := y+1 \quad x := y$
(ii) Тізбекті композиция:	*p ; q*
(iii) Шартты тексеру:	if $x < y$ then *p* else *q*
(iv) *For* цикл:	for y do *p*
(v) *While* цикл:	while $x < y$ do *p*

Мұндағы (iii) және (v)-те $<$ қатынасты $>$, $\ge$, $\le$, $=$, немесе $\ne$ қатынастарының кез келген біреуімен алмастыруға болады.
Осы конструкциямен индуктивті құрастырылған программаны *while* программа деп атайды. Позиция (v)-тегі while конструкциясыз құрастырылған програма *for* программа деп атайды.
for-дың семантикасы мынадай: циклге енер алдында айнымалы y бағаланады және цикл *p* осынша рет орындалады. Цикл ішінде y-ке жасалған ешқандай меншіктеу циклдің орындалу санын өзгерте алмайды және циклдің орындалуы y-тің мәнін өзгерте алмайды (айқын меншіктеу жағдайынан басқа).
$\mathbb{N}$-дегі әрбір while программа, тек бірден аспайтын while-ды қамтитын жалғыз программаға эквивалентті болатынын дәлелдеңіз. Программа *for* циклден қанша керек болса, сонша қамти алады және сізге қосымша локалды айнымалыларды жариялауға мүмкіндік берілген, тек олар эквиваленттілік анықтамасында болмауы қажет.

91. Гёдел примитивті рекурсивті функцияларды теориялы сандық функцияның $\mathbb{N}^k \to \mathbb{N}$ ең кіші класы $\mathcal{P}$ ретінде анықтады, ол тұрақты нөлдік функцияны $zero(\,) = 0$, реттелген функцияны $s(x) = x+1$, және $\pi_k^n(x_1, \ldots, x_n) = x_k$ арқылы берілген проекция функцияларын $\pi_k^n : \mathbb{N}^n \to \mathbb{N}$ қамтиды, мұндағы $1 \le k \le n$. Және ол келесі амалдарда тұйықталған.

(a) Композиция:

Егер $f:\mathbb{N}^k \to \mathbb{N}$ және $g_1,\dots,g_k:\mathbb{N}^n \to \mathbb{N}$ $\mathcal{P}$-де жатса, онда функция $f\circ(g_1,\dots,g_k):\mathbb{N}^n \to \mathbb{N}$ болады, ол $\overline{x}=x_1,\dots,x_n$ енгізуде келесі қатынасты береді

$$(f\circ(g_1,\dots,g_k))(\overline{x}) \overset{def}{=} f(g_1(\overline{x}),\dots,g_k(\overline{x})) \quad .$$

(b) Примитивті рекурсивтілік:

Егер $1\le i\le k$ болып $h_i:\mathbb{N}^{n-1}\to\mathbb{N}$ және $g_i:\mathbb{N}^{n+k}\to\mathbb{N}$ $\mathcal{P}$ -де жатса, онда функция $f_i:\mathbb{N}^n\to\mathbb{N}$, $1\le i\le k$ болады, ал ол өзіндік индукция көмегімен анықталады:

$$f_i(0,\overline{x}) \overset{\text{def}}{=} h_i(\overline{x})$$

$$f_i(x+1,\overline{x}) \overset{\text{def}}{=} g_i(x,\overline{x},f_1,(x,\overline{x}),\dots,f_k(x,\overline{x})),$$

мұндағы $\overline{x}=x_2,\dots,x_n$.

Теориялы сандық функцияның ең кіші класы ретінде $\mathcal{C}$ класын анықтаңыз, ол тұрақты нөлдік функцияны, реттелген функцияны, проекция функциясын қамтиды және келесі амалдарда тұйықталған.

(a) Композиция:

Егер $f:\mathbb{N}^m\to\mathbb{N}^n$ және $g:\mathbb{N}^n\to\mathbb{N}^k$ $\mathcal{C}$-де жатса, онда функцияда $g\circ f:\mathbb{N}^m\to\mathbb{N}^n$ жатады, ал ол

$$(g\circ f)(\overline{x}) \overset{def}{=} g(f(\overline{x}))\cdot$$

қатынаспен анықталады. Жоғарыда қарастырылған жағдайдан айырмашылығы тек Гёдел композициясында ғана, соған назар аударыңыз.

(b) Кортеж:

Егер $f_1,\dots,f_n:\mathbb{N}^m\to\mathbb{N}^n$ $\mathcal{C}$-де жатса, онда функция $(f_1,\dots,f_n):\mathbb{N}^m\to\mathbb{N}^n$ болады, ал ол

$$(f_1,\dots,f_n)(\overline{x}) \overset{def}{=} (f_1(\overline{x}),\dots,f_n(\overline{x}))$$

қатынаспен анықталады.

(c) Итерацияланған композиция:
Егер $g:\mathbb{N}^n \to \mathbb{N}^n \in \mathcal{C}$ болса, онда функция $f:\mathbb{N}^{n+1} \to \mathbb{N}^n$ болады, ал ол

$$f(x,\overline{y}) \overset{def}{=} g^x(\overline{y})$$

қатынаспен анықталады және ол $\mathcal{C}$-да жатады, мұндағы g^n-ді g-ді өзімен өзін n рет композиция жасау арқылы алады:

$$g^0(\overline{y}) \overset{\text{def}}{=} \overline{y}$$

$$g^{n+1}(\overline{y}) \overset{\text{def}}{=} g(g^n(\overline{y})).$$

Жалғыз нәтижелі функция үшін $\mathcal{P}$ мен $\mathcal{C}$ беттесетінін көрсетіңіз.

92. (Meyer және Ritchie [85]) 90-шы жаттығуда қарастырылған $\mathbb{N}$-дегі *for* программа, примитивті рекурсивті функцияның дәл өзін есептейтінін көрсетіңіз. 91-аралас жаттығудан кез келген эквивалент дефинацияны (анықтаманы) пайдаланыңыз.

HS93. (Meyer и Ritchie [85]) Егер тек егер функция примитивті рекурсивті болса, онда ол примитивті рекурсивті уақыт шектеуінде *while* программамен есептелінетінін көрсетіңіз.

**H94. Аккерман функциясы индуктивті былай анықталады:

$$A(0,n) \overset{\text{def}}{=} n+1$$

$$A(m+1,0) \overset{\text{def}}{=} A(m,1)$$

$$A(m+1,n+1) \overset{\text{def}}{=} A(m,A(m+1,n)).$$

Сондықтан,

$$\begin{aligned} A(1,n) &\overset{\text{def}}{=} n+2 \\ A(2,n) &\overset{\text{def}}{=} 2n+3 \\ A(3,n) &\overset{\text{def}}{=} 2^{n+3}-3 \\ A(4,n) &\overset{\text{def}}{=} \underbrace{2^{2^{\cdot^{\cdot^{\cdot^{2}}}}}}_{n+3}-3 \,. \end{aligned}$$

Кез келген примитивті функцияға қарағанда ё$x.A(x,2)$ асимтотикалы жылдам өсетінін дәлелдеңіз. 91 және 92-ші аралас жаттығуларда келтірілген примитивті рекурциялы функцияның үш түрлі эквивалентті сипаттамасының кез келгенін пайдаланыңыз.

95. C немесе командалық жол UNIX -тен басқа кез келген программалау тілінде өзін-өзі теретін бағдарлама жазыңыз. Сіздің программаңыз нөлдік болмауы қажет, кез келген I/O файлды іске қоса алмайды және синтаксистік түрі дұрыс болуы қажет. 34-лекцияда C және командалық жол UNIX -те жазылған программалар келтірілген.

H96. Пост сәйкестік проблемасы (PCP) шешілмейтінін дәлелдеңіз (25-ші лекцияны қараңыз).

97. Кез келген Тюринг машинасы барлық жерде дерлік полиномиалдыдан көп уақыт жұмсайтындай етіп рекурсивті жиын A-ны құрастырыңыз. Басқаша айтсақ, $L(M)=A$ орындалатын кез келген M және кез келген тұрақты k үшін, барлық бірақ шектеулі көп енгізу x үшін, M машина $|x|^k$ қадамнан артық жұмыс жасайды. $A \in EXPTIME$ қатынасын ала аласыз ба?

HS98. Минималды индекстердің барлық жиыны

$$\{x \mid \forall y < x \;\; \varphi \neq \varphi_x\}$$

иммунды болатынын көрсетіңіз.

H99. Әрбір шексіз р.с. жиынның шексіз рекурсивті ішкі жиыны бар болатынын көрсетіңіз.

100. (a) Төмендегі шарттар орындалатын Тюринг машинасының индексі үшін A р.с. жиын болмайтынын көрсетіңіз:

- егер $i \in A$ болса, онда M_i жалпы болады; және
- әрбір рекурсивті жиынның A-да индексі бар.

*Н(b) Төмендегі шарттар орындалатын Тюринг машинасының индексі үшін A р.с. жиын болатынын көрсетіңіз

- егер $i \in A$ болса, онда $L(M_i)$ рекурсивті; және
- әрбір рекурсивті жиынның A-да индексі бар.

101. (a) Рекурсивтік теориясын пайдаланып өзінің индексін есептейтін программаны жеңіл табуға болады: $\varphi_{\sigma(n)}(x) = n$ орындалатын етіп жалпы σ-ны құрастырамыз, осыдан кейін қозғалмайтын нүктені таңдаймыз. Бір-бірінің индекстерін есептейтін, әртүрлі екі программа табылатынын дәлелдеңіз. Яғни, $m, n \in \omega, m \neq n$ табылып, $\varphi_n(x) = m$ және $\varphi_m(x) = n$ орындалатынын көрсетіңіз.

(b) Бір-бірін печаттайтын әртүрлі екі бағдарламаны кез келген программалау тілінде жазыңыз.

102. (a) φ және ψ Гёдел нөмірлеуі болсын. Мұнда индекстер $m, n \in \omega$ табылып, $\varphi_n(x) = m$ және $\psi_m(x) = n$ орындалатынын көрсетіңіз.

(b) Бірін-бірі шығаратын екі бағдарламаны өзіңіз жақсы көретін екі программалау тілінде жазыңыз.

*103. Полиномиалды уақытта есептеуге келмейтін рекурсивті жиын A болсын. Мұнда шексіз рекурсивті ішкі $B \subseteq A$ жиын табылып, ол ішкі жиында A-ны есептейтін кез келген ТМ B-да полиномиалды уақыттан көп уақыт б.ж.д. жұмсайды. Осыны көрсетіңіз.

104. (a) Егер $P = NP$ болса және егер айқын уақыттық шектеу n^k-де бейдетерминирлі полиномиалды уақытты машина берілсе, онда эквивалентті детерминирлі полиномиалды уақытты машинаны тиімді (яғни, жалпы рекурсивті функция көмегімен) табуға болатынын көрсетіңіз.

(b) Егер уақыттық шектеу n^k белгісіз болса да, осы тұжырым ақиқат болатынын көрсетіңіз.

105. Келесі проблемалар үдету жайлы теорема дәлелдемесіне қатысты айтылады (32.2-теорема).

(a) Берілген машина M_i-де, M_i машина *A* жағдай немесе *B* жағдайға түсе ме, шешілмегенін көрсетіңіз.

(b) $m(i)$-дің мәнін тиімді алуға болмайтынын көрсетіңіз.

**(c) *k*-да $f^{n-k}(2)$ уақыт жұмыс жасайтын *A* машинаны тиімді алуға болмайтынын көрсетіңіз.

106. Сызықты уақытты жиынның *P* бейтривиалды қасиеті болсын, ол қасиет барлық регуляр жиында ақиқат болады. *P*-ның шешілмейтінін көрсетіңіз.

*[HS]107. Бұл изоморфизмнің жалпы теориясы, ол екі теорияны да қамтиды: Роджерс изоморфизмі (34.3-теорема, 108-аралас жаттығу) және ерекше жағдай ретінде Майхилл изоморфизмі (109-шы аралас жаттығу). Мұнда ω-дағы бинарлық қатынастарда $\circ$ арқылы реляциялық композиция және $^{-1}$ арқылы реверсивті (кері) оператор белгіленсін. Яғни, $R, S \subseteq \omega \times \omega$ үшін, анықтаңыз:

$$R \circ S \overset{\text{def}}{=} \{(u,\omega) \mid \exists \upsilon\ (u,\upsilon) \in R, (\upsilon,\omega) \in S\}$$

$$R^{-1} \overset{\text{def}}{=} \{(u,\upsilon) \mid (\upsilon,u) \in R\}.$$

Функция $f : \omega \to \omega$ үшін анықтаңыз:

$$\text{graph}\, f \overset{\text{def}}{=} \{(x, f(x)) \mid x \in \omega\} \quad .$$

$R \circ R^{-1} \circ R \subseteq R$ қатынасы орындалатын ω-дағы бинарлы қатынас *R* болсын. Және $\text{graph}\, f \subseteq R$ және $\text{graph g} \subseteq R^{-1}$ қатынастары орындалатын бірдің-бірге жалпы рекурсивті функциялдар $f, \text{g} : \omega \to \omega$ болсын. $\text{graph}\ h \subseteq R$ қатынасы орындалатын бірдің-бірге және съюрективті жалпы рекурсивті функция $h : \omega \to \omega$ табылатынын көрсетіңіз.

[H]108. Мұнда Роджерс изоморфизмі теоремасының (34.3-теорема) алтернативті дәлелдемесін алу үшін, біз 107-ші аралас жаттығуды пайдаланамыз. Барлық i үшін $\varphi_i = \psi_{\sigma(i)}$ және $\psi_i = \varphi_{\tau(i)}$ қатынастары орындалатын $\sigma, \tau : \omega \to \omega$ табылады деп ұйғарамыз. Барлық i үшін $\varphi_i = \psi_{\rho(i)}$ қатынасы орындалатын бірдің-бірге және сьюрективті жалпы рекурсивті функция $h : \omega \to \omega$ табылатынын көрсетіңіз.

[H]109. Кантор-Шрёдер-Берштейн теоремасы былай дейді: егер бірдің бірге функциялары $A \to B$ және $B \to A$ бар болса, онда A мен B -ның қуаттары бірдей болады. Майхиллдың изоморфизм жайлы теоремасы деген атпен белгілі, Майхиллға ([104]-ті қараңыз) таңылған тиімді нұсқа бар. Бірдің-бірге келтіруі арқылы бір-біріне келтірілетін кез келген екі жиын, рекурсивті изоморфты болатынын көрсетіңіз. Басқаша айтсақ, $A, B \subseteq \omega$ болсын және барлық $x \in \omega$ үшін $x \in A \Leftrightarrow f(x) \in B$ және $x \in B \Leftrightarrow g(x) \in A$ қатынастары орындалатын, бірдің-бірге жалпы рекурсивті функциялар $f, g : \omega \to \omega$ болсын. Барлық $x \in \omega$ үшін $x \in A \Leftrightarrow h(x) \in B$ қатынасы орындалатын бірдің-бірге (биективті) және сьюрективті жалпы рекурсивті функция $h : \omega \to \omega$ табылатынын көрсетіңіз.

110. *[S] (a) A және $\sim A$ екеуі де шексіз болатын, бірақ олардың ешқайсысы полиномиалды уақытта есептелетін шексіз ішкі жиынды қамтымайтын, рекурсивті A жиынды көрсетіңіз.

*(b) TM -ның р.с. тізімі табылатын рекурсивті жиындар үйірі $\mathcal{L}$ болсын, олар тек $\mathcal{L}$-да жататын жиындарда әрқашан тоқтайды және бәріне қабылдау-ашық. A және $\sim A$ екеуі де шексіз болатын, бірақ A немесе $\sim A$ -ның ешқандай ішкі жиыны $\mathcal{L}$ -де жатпайтын, рекурсивті A жиынды көрсетіңіз.

111. Жалқау тексеруді

$$\varphi_{\mathrm{cond}(i,j)}(x, y) = \begin{cases} \varphi_i(y), \text{ егер } & x = 0, \\ \varphi_j(y), \text{егер } & x \neq 0. \end{cases}$$

құрастыру үшін, s_n^m-ге негізделген конструкцияны пайдаланыңыз және шартты жалқау тексеруді ұйымдастыруға арналған, 33-лекцияда баяндалған универсал функцияның қасиеттерін пайдаланыңыз. Шартты тексеру жалқау дегенді мына мағынада түсінеміз: егер $x \neq 0$ болса, онда ол $\varphi_i(y)$-ті бағалауға тырыспауы керек, ал егер $x = 0$ болса, онда $\varphi_j(y)$-ті бағалауға тырыспауы керек.

112. 37.8-лемманың келесі жалпылануын дәлелдеңіз. Егер $A \leq_{\mathrm{m}} B$ және A өнімді болса, онда B-да өнімді болады.

[H]113. A және $\sim A$ екеуі де өнімді болатын A жиынды көрсетіңіз.

114. Егер $A \subseteq C$ және $B \subseteq \sim C$ қатынастары орындалатындай рекурсивті C жиын табылса, онда қиылыспайтын A және B жиындары рекурсивті ажыратылады (сеперабелді) деп аталады. Егер осындай C жиын табылмаса, онда A және B жиындары рекурсивті ажыратылмайды деп аталады.

(a) Қиылыспайтын ко-р.с. жиынның кез келген жұбы рекурсивті ажыратылмаған болатынын дәлелдеңіз.

*[H](b) Рекурсивті ажыратылмайтын р.с. жиынның жұбын құрастырыңыз.

[H]115. Егер $\varphi_x = \varphi_y$ болса, онда $x \equiv y$ болады деп анықтаймыз. $\equiv$-эквиваленттіліктің өзгешеленетін кластарының кез келген жұбы рекурсивті ажыратылмайтынын көрсетіңіз. (114-ші аралас жаттығуды қараңыз).

[H]116. Күрделіктің абстракциялық мерасы Φ болсын (Лекция *J*-ді қараңыз). Φ_i -дің дербес рекурциялы функция болатынын дәлелдеңіз, және *i*-ден Φ_i -дің индексін тиімді алу мүмкіндігі бар екенін де көрсетіңіз. Яғни, $\varphi_{\sigma(i)} = \Phi_i$ қатынасы орындалатындай жалпы рекурциялы функция σ табылады.

117. Күрделіктің абстракциялық мерасы Φ болсын. Кез келген жалпы рекурциялы функция g үшін, *f*-тің барлық индексі *i*-де $\varphi_{\sigma(i)} = \Phi_i$ б.ж.д. орындалатын, 0,1-мәнді жалпы рекурциялы функция *f* табылатынын дәлелдеңіз.

118. [H](a) Күрделіктің абстракциялық мерасы Φ болсын. Барлық индексі i-де $\Phi_i(n) \le f(n, \varphi_i(n))$ б.ж.д. орындалатын, жалпы рекурциялы функция f табылмайтынын дәлелдеңіз; яғни, рекурсивті функцияның күрделілігі олардың мәнімен бірқалыпты рекурсивті шектелмейді.

(b) Екінші жағынан, рекурсивті функцияның мәндері олардың күрделілігімен бірқалыпты рекурсивті шектелгенін көрсетіңі; яғни, барлық индексі i-де $\varphi_i(n) \le g(n, \Phi_i(n))$ б.ж.д. орындалатын, жалпы рекурциялы функция g табылады.

119. (a) Егер тек егер әйтеуір бір $\varphi_i(n)\downarrow$ болғанда $\Phi_i(n) \le f(n)$ қатынасы орындалатын жалпы рекурциялы функция f табылса, онда φ_i -дің анықталу аймағы рекурсивті болатынын дәлелдеңіз.

(b) (a)-дан қорытынды жасаңыз: шексіз қабылдау-ашық енгізулерде кез келген ТМ қабылдау-ашық рекурсивті р.с. жиын, кез келген жалпы ТМ-мен салыстырғанда көп кеңістік пайдалануы қажет.

120. Күрделіктің абстракциялы өлшемі үшін үзіліс теореманы дәлелдеңіз (J.1-теорема).

[H]121. (Блюм [17]) Келесі бәсеңдету теоремасын дәлелдеңіз. Барлық жалпы рекурциялы функциялар f және g үшін, f-тің индексі i табылып, кез келген n үшін $\Phi_i(n) > g(n)$ қатынасы орындалады.

122. Кез келген күрделіліктің абстракциялы екі өлшемі бірқалыпты бір-бірімен рекурсивті шектелгенін көрсетіңіз. Формалды, Φ және Ψ күрделіктің мералары болсын. Барлық индексі i -де $\Psi_i(n) \le f(n, \Phi_i(n))$ б.ж.д. және $\Phi_i(n) \le f(n, \Psi_i(n))$ б.ж.д. қатынастары орындалатындай етіп, f функциясын көрсетіңіз.

123. (a) (Комбинациялайтын лемма) Күрделіктің абстракциялық өлшемі Φ болсын. Екі айнымалылы жалпы рекурсивті оператор c болсын, ол операторда кез келген i, j үшін егер $\varphi_i(n)\downarrow$ және $\varphi_j(n)\downarrow$ болса, онда $\varphi_{c(i,j)}(n)\downarrow$ болады. Барлық i, j үшін

$$\Phi_{c(i,j)}(n) \le h(n, \Phi_i(n), \Phi_j(n)) \text{ б.ж.д.}$$

орындалатын жалпы рекурсивті функция h табылатынын көрсетіңіз.

(b) Жеке жағдайда, жалпы рекурсивті оператор c болсын , мұнда барлық i үшін егер $\varphi_i(n)\downarrow$ болса, онда $\varphi_{c(i)}(n)\downarrow$ болады. Барлық i үшін

$$\Phi_{c(i)}(n) \le h(n, \Phi_i(n))$$

орындалатын жалпы рекурсивті функция h табылатынын көрсетіңіз.

[H]124. Күрделіктің абстракциялық өлшемі Φ болсын. Егер f-тегі индекс көмегімен қандай да бір $\mathcal{C}_f^{\Phi}$ күрделілік класы берілсе, онда бірқалыпты және тиімді түрде күрделіліктің үлкен қатаң класын табуға болатынын көрсетіңіз. Яғни, толық рекурсивті функция σ табылып, $\mathcal{C}_{\varphi_{\sigma(i)}}^{\Phi}$ класы $\mathcal{C}_{\varphi_i}^{\Phi}$ класын қатаң қамтиды.

125. Егер барлық i үшін $\varphi_{\sigma(i)} = \Phi_i$ және $\Phi_{\sigma(i)}(n) \le \Phi_i(n)$ б.ж.д. орындалатындай жалпы рекурсивті функция σ табылса, онда күрделіліктің абстракциялық өлшемі Φ толық әділетті деп аталады. Басқаша айтсақ, берілген дербес рекурсивті функция φ_i -дің күрделігі үшін, біз индексті тиімді таба аламыз және енді функцияның өзін есептеумен салыстырғанда күрделікті есептеу аса қиындық тудырмайды.

(a) Тьюринг машинасының кеңістікті күрделілігі толық әділетті болатынына көз жеткізіңіз.

(b) Күрделіктің абстракциялық өлшемінің барлығы толық әділетті бола бермейтінін көрсетіңіз.

*[S]126. Біріктіру теоремасының (J.3-теорема) тұжырымындағы f_k функция, монотондық шартты қанағатандару үшін постулантталып еді, ол монотондық шартта барлық i және n үшін $f_i(n) \le f_{i+1}(n)$ орындалатын. Осы шарт болмаса теорема орындалмауы мүмкін екендігін көрсетіңіз.

127. Егер тек жиында жататын функцияларды және бәрін көрсететін р.с. индекстер жиыны табылса, онда жалпы рекурциялы функциялар жиыны рекурсивті саналады (р.с.) деп аталады. Мысалы, күрделілік класы P р.с. болады, өйткені біз оны полиномиалды уақытағы сағатты TM -ның р.с. тізімі арқылы өрнектей аламыз.

(a) Күрделіктің абстракциялық мерасы Φ болсын. Барлық дерлік жерде тұрақты болатын функциялардың бәрін қамтитын Φ -күрделілік класы, р.с. болатынын көрсетіңіз.

(b) Күрделілік өлшемі Φ шектеулі модификациялар астында инвариантты деп ұйғарайық; яғни егер $f(n) = g(n)$ б.ж.д. болса, онда $f \in \mathcal{C}_t^{\Phi}$ болады, егер тек егер $g \in \mathcal{C}_t^{\Phi}$ орындалатын болса. Барлық Φ -күрделі кластар р.с. болатынын көрсетіңіз.

*Н(c) Мұнда $\mathcal{C}_f^{\Phi}$ р.с. бола алмайтын, Φ өлшемі және жалпы рекурсивті функция f табылатынын көрсетіңіз.

128. 35.1-ші теореманы дәлелдеңіз.

129. $\Sigma_n^0 \cup \Pi_n^0 \neq \Delta_{n+1}^0$ орындалатынын дәлелдеңіз (35-ші лекцияны қараңыз).

130. Толық болудың келесі нәтижелерін дәлелдеңіз (анықтамаларды 35-ші лекциядан қараңыз).

(a) Σ_1^0 үшін HP $\leq_m$ -толық болады.

(b) Π_1^0 үшін EMPTY $\leq_m$ -толық болады.

(c) Π_2^0 үшін TOTAL $\leq_m$ -толық болады.

131. Тюринг машинасы және дербес рекурсивті функция сәйкес M_i және φ_i -мен белгіленсін. Жиындар

(a) $\mathrm{ALL} \stackrel{\mathrm{def}}{=} \{i \mid L(M_i) = \Sigma^*\}$

S(b) $\mathrm{EQUAL} \stackrel{\mathrm{def}}{=} \{(i, j) \mid \varphi_i = \varphi_j\}$

Π_2^0 үшін $\leq_m$ -толық болатынын көрсетіңіз.

**Н132. Барлық регуляр жиынды қамтитын, рекурсивті жиындар үйірінің кез келгені $\mathcal{L}$ болсын, мұнда әрқашан тоқтайтын, бәріне қабылдау-ашық және жиын тек $\mathcal{L}$ -де болатын TM -ның р.с. тізімі табылады. Σ_3^0 үшін $\{i \mid L(M_i) \in \mathcal{L}\}$ $\leq_m$ -толық болатынын көрсетіңіз.

[H]133. Σ_3^0 үшін келесі шешілу проблемаларының $\leq_m$-толық болатынын дәлелдеңіз.

(a) Егер Тюринг машинасы M берілсе, онда $L(M)$ регу-ляр жиын бола ала ма?

(b) Егер Тюринг машинасы M берілсе, онда $L(M)$ контекстен тәуелсіз тіл бола ала ма?

(c) Егер Тюринг машинасы M берілсе, онда $L(M)$ рекурсивті жиын бола ала ма?

134. [S](a) Пеано арифметикасы PA деп белгіленсін (немесе сандар теориясындағы сіздердің сүйікті дәлелдеу жүйеңіз). Полиномиалды уақытта жұмыс жасайтын, PA-да дәлелдеуге келмейтін полиномиалды уақытты машина табыла ма?

*[S](b) Егер полиномиалды уақытта жұмыс жасайтын машина берілсе, онда n^k формадағы шектеуді әрқашан тиімді есептеп алуға бола ма?

*[H]135. Мұнда Пеано арифметикасында жалпы дәлелдеуге келмейтін, жалпы есептелетін функция $f : \mathbb{N} \to \mathbb{N}$ табылатынын дәлелдеңіз.

*[S]136. Келесі тұжырымды формалды түрде жазыңыз және дәлелдеңіз. “ Сандар теориясының кез келген формалды дедуктивті жүйесінде ешқандай алгоритм соңына дейін толық корректі дәлелдей алмайтын шешілу проблемасы табылады.”

[H]137. Егер бірдің-бірге σ келтірімділік арқылы $A \leq_m B$ болса, онда $A \leq_1 B$ деп жазамыз. $\leq_1$ үшін $K = \{x \mid M_x(x)\downarrow\}$ жиын Σ_1^0 - толық болатынын көрсетіңіз.

[H]138. (a) 130-аралас жаттығуда Π_2^0 үшін

$$\mathrm{TOTAL} \stackrel{\mathrm{def}}{=} \{M \mid \forall x\ \exists k\ M(x)\downarrow^k\}$$

$\leq_m$ -толық болатынын көрсеткен едік. Ал келесі жиын жайлы не айтамыз:

$$\mathrm{WAYTOTAL} \stackrel{\mathrm{def}}{=} \left\{M \mid \exists k\ \forall x\ M(x)\downarrow^k\right\}\ ?$$

*(b) Егер $M \in \mathrm{WAYTOTAL}$ берілген болса, онда k шектеуді әрқашан тиімді есептеп алуға бола ма?

**[HS]139. Төмендегі мәселелердің біреуі Σ_2^0-толық, ал екіншісі Π_1^1-толық. Қайсысы қайда жатады? Сіз жасаған таңдаудың дұрыстығына сенімді дәлел келтіріңіз.

(a) Егер бейдетерминирлі Тюринг машинасы M және M-нің күйі q берілсе, онда есептеу траекториясы бойында M машина q күйге шектеулі жиі көше алатындай, енгізу ϵ-де M-нің есептеу траекториясы бар ма?

(b) Егер бейдетерминирлі Тюринг машинасы M және M-нің күйі q берілсе, онда әрбір есептеу траекториясында енгізу ϵ-де M машина q күйге шектеулі жиі көше ала ма?

140. (a) $LOGSPACE$-тегі келесі проблеманы дәлелдеңіз. Егер шектеулі бинарлы қатынас берілсе, ол транзитивті бола ма?

[S](b) Π_1^0 үшін келесі проблема $\leq_m$-толық болатынын дәлелдеңіз. Егер R-де жұптар жиынына қабылдау-ашық Тьюринг машинасының көмегімен бинарлы қатынас $R \subseteq \omega^2$ берілсін десек, онда R транзитивті бола ма?

141. $\mathbb{N}$-де кез келген Π_1^1 формуланы (39.1)-ші формаға жұптар пайда болуын және сколоминазация ережесін

$$\forall x:\mathbb{N}\ \exists f:\mathbb{N}\to\mathbb{N}\ \varphi(f,x) \quad \mapsto \quad \exists g:\mathbb{N}\to(\mathbb{N}\to\mathbb{N})\ \forall x:\mathbb{N}\ \varphi(g(x),x).$$

пайдаланып келтіруге болатынына көз жеткізіңіз.

142. $\mathbb{N}$-де бірінші ретті сандар теориясы гиперэлементар болатынын, бірақ элементар болмайтынын көрсетіңіз (39.1-ші лекцияны қараңыз). Мұнда бірінші ретті сандар теориясы мына жиынға қатысты

$\mathrm{Th}(\mathbb{N}) \stackrel{\mathrm{def}}{=} \{\varphi \mid \varphi$ бірінші ретті сандар теориясы тілінің сөйлемі және $\mathbb{N} \models \varphi\}$.

143. 41.2-салдарды дәлелдеңіз.

**144. ([77]) 39 және 40-лекцияларда аргументтелгендей, $\mathbb{N}$ -де IND программада Π_1^1 жиындағы дәлдікпен қабылдау-ашық. Кез келген саналатын структурада, IND программада Π_1^1 жиындағы дәлдікпен қабылдау-ашық болатынын көрсетіңіз. Деректердің абстракциялы структурасы сөздік болады, ол деректер элементін кілтпен ассоциацияландырады және кілтпен ізделініп табылатындай етіп жасайды.

ҮЙ ЖҰМЫСТАРЫНЫҢ ШЕШІМІ

1- үй жұмысының шешімі

1. (a) $O(n\log n)$ уақытта қабылдау-ашық болатын мүмкін бейрегуляр жиынның біреуі

$$\{a^{2^n} \mid n \geq 0\}.$$

Әрбір екінші элементті өшіріп және қалған *a*-ның саны жұп бола ма соған тексеріп, енгізуді қайта-қайта сканерлейміз. Осы процесс жалғыз *a* қалғанша жүргізіледі.

(b) Бұл дәлелдеуде, қиылысатын тізбектер тек күйлер тізбегі болатыны ескеріледі. Қабылдау-ашық күйге енбес бұрын *M* оңға тірелгенше жылжиды деп ұйғарайық. *M*-нің күйлерінің саны *q* болсын. 1.4-лемманың дәлелдемесіндегі аргумент аналогын пайдаланамыз, егер *L*(*M*) бейрегулярлы болса, онда қабылдау-ашық енгізулерде *M*-мен туындалатын, қиылысатын тізбектер ұзындығына тиянақты шектеулі шектеу болмайды. Сонда әрбір $k > 0$ үшін, *L*(*M*) жолы табылады, бұл жол үшін ұзындығы $\geq k$ болатын, қиылысатын тізбектер генерацияланады. Осындай жолдардың ұзындығы ең қысқасы x_k болсын және $n = |x_k|$ деп алайық. 1.3-теореманың дәлелдемесі сияқты, енгізу x_k-да *M* машина *n/2*-ден аз емес әртүрлі қиылысатын тізбектер генерациялауы қажет. Кері жағдайда ішкі жол x_k-ны жойып жіберіп, ең ұзын қиылысатын тізбекті генерациялайтын қысқа жол алуымызға болар еді, бірақ ол x_k-ның минималдығына қайшы келеді. Лексикографикалық реттілікпен анықталатын, *n/2* мүмкін өте қысқа қиылысатын тізбектер жиыны S_0 болсын. S_0 жиыны *m-1*-ге дейінгі ұзындықты қиылысатын тізбектердің барлығын қамтуы қажет, мұндағы, *m*

$$\sum_{i=1}^{m} q^i \geq \frac{n}{2}$$

орындалатын ең аз сан.

x_k-ның жұмыс істеу уақыты *S*-тегі қыйылысатын тізбектердің ұзындықтарының қосындысымен төменнен шектеледі, өйткені әрбір қиылысатын тізбектің әрбір элементін генерациялау бір қадамды қажет етеді; сондықтан:

Ал $T(n) \quad \geq \quad \sum_{c \in S} |c| \quad \geq \quad \sum_{c \in S_0} |c| \quad \geq \quad \sum_{i=1}^{m-1} iq^i.$

$$T(n) \quad \geq \quad \Omega(n \log n)$$

шектеуі осы теңсіздіктерден және белгілі бір арифметикадан алынады.

2. (a) (Бейдетерминирлі) k-негізгі шектеулі автоматтың (k-FA) 7 картежі бар, олар

$$M \;=\; (Q, \Sigma, \vdash, \dashv, \delta, s, f),$$

мұндағы

- Q – шектеулі жиын (күйлер);
- Σ – шектеулі жиын (енгізу алфавиті);
- $\vdash, \dashv$ таңбалары Σ-да жатпайды (сәйкес, оң және сол түкпір белгі);
- $\delta \subseteq (Q \times (\Sigma \cup \{\vdash, \dashv\})^k) \times (Q \times \{-1, 0, +1\}^k)$ (ауысу қатынасы),
- $s \in Q$ (бастапқы күй) және
- $f \in Q$ (қабылдау-ашық күй).

Сонымен, егер

$$((p, a_1, \ldots, a_k), (q, d_1, \ldots, d_k)) \;\in\; \delta \tag{1}$$

болса, онда егер $a_i = \vdash$ болса, онда $d_i \neq -1$ болады, ал егер $a_i = \dashv$ болса, онда $d_i \neq +1$ болады. Егер δ бір мәнді болса, онда M машинасы детерминирлі болады.

Бейформалды, (1)-дің мағынасы: егер машина $1 \leq i \leq k$, өзінің i-ші бас тиегі астындағы a_i символын сканерлеп p күйде тұрса, онда ол $1 \le i \le k$, i-ші бас тиекті d_i бағытына жылжыта алады және q күйіне енеді. Бас тиек енгізу сыртына шығып кетпеуі үшін $\vdash$ және $\dashv$ таңбаларын пайдаланып, шарттар қойып отырмыз.

$x \in \Sigma^*$ болсын, $x = x_1 x_2 \ldots x_n$ делік, мұндағы $x_i \in \Sigma$, $1 \le i \le n$. $x_0 = \vdash$ және $x_{n+1} = \dashv$ болсын. Енгізу x-тегі M-нің конфигурациясы деп

$$Q \times \{0, \ldots, n+1\}^k$$

элементін айтамыз.

Бейформалды, ағымдағы күй мен k бас тиектің позициясын конфигурация анықтайды. Егер α мен β конфигурация болса, онда:

$$\alpha \xrightarrow{1} \beta$$

деп жазамыз. Және егер

$$\alpha = (p,\ i_1,\ i_2, \ldots, i_k)$$
$$\beta = (q,\ i_1 + d_1,\ i_2 + d_2, \ldots, i_k + d_k)$$

және

$$((p, x_{i_1}, x_{i_2}, \ldots, x_{i_k}), (q, d_1, d_2, \ldots, d_k)) \in \delta$$

орындалса, онда α-дан β конфигурация бір қадамда алынады деп айтамыз.

Мұндағы $\xrightarrow{1}$ қатынасының рефлексті транзитивті тұйықталуы $\xrightarrow{*}$ таңбасымен белгіленеді. Енгізу x-тегі M-нің бастапқы және қабылдау-ашық конструкциялары сәйкес

$$(s, 0, 0, \ldots, 0) \qquad (f, n+1, n+1, \ldots, n+1)$$

конфигурациялар болады. Егер

$$(s, 0, 0, \ldots, 0) \xrightarrow{*} (f, n+1, n+1, \ldots, n+1)$$

болса, онда машинада енгізу x *-ке қабылдау-ашық* деп айтамыз.

(b) $(\Leftarrow)$ k-ға M берілген, кеңістікті шектелген $O(\log n)$ Тюринг N машинасын құрастырамыз, ол M-ді келесі түрде симуляция жасайды. N-нің жұмыс таспасы k соқпаққа бөлінеді, оның әрқайсысы $-(n+1)$ және $n+1$ арасында, аралықтың шеткі мәндерін қоса есептегенде, $O(\log n)$ битті бинарлы санды сақтайды. Бұл сандар, N-нің оқуға арналған бас тиегінің орналасуына қарай, M-нің симулация жасалынған k бас тиегінің орналасуын жазуда қолданылады. M-нің күйі N-нің соңғы бақылауында есте сақталады. M-нің қозғалысын симуляция жасау үшін, ағымдағы уақытта M-нің әрбір бас тиегі қандай символды сканирлеп жатыр, соны N білуі қажет. Сол жақ түпкірден бастау алған N, өз бас тиегін келесі оң жақтағы бір ұяшықты оқу мақсатында бас тиекті бірнеше мәрте оңға жылжытады және k санағыштың әрқайсысын кемітеді. Санақшы 0-ді сақтаған әрбір жағдайда білетініміз, осы санақшыға сәйкес келетін M-нің симуляция жасалатын бас тиегі, ағымдағы уақытта N-де сканерлейтін, енгізу таспасының ұяшығын сканерлейді. N өзінің енгізу таспасынан символды оқиды және оны соңғы басқаруда есте сақтайды. N-нің бас тиегі таспаның оң жақ түкпіріне жеткенде, ол M-нің k симуляцияланатын бас тиектерінің барлық символдарын көріп шықты. Ол кезегінде санағыштарды дайындайды және M-нің ауысу қатынасына сай M-нің симуляция жасалатын күйін өзгертеді, ол N-нің соңғы бақылауында есте сақталады. Осыдан кейін, ол бас тиекті кері қарай солға жылжытады, осы жолда k санағыштарды арттырып отырады және симуляцияның келесі қадамы жасалады.

$(\Rightarrow)$ Жұмыс лентасының $\{0,1\}$ алфавитінде N машинасы кеңістікті шектелген $O(\log n)$ ТМ болсын. Біз тек оқуға арналған екі жақты енгізу бас тиекті және әрқайсысы тек теріс емес бүтін санды сақтай алатын шектеулі жиынды санағышты M машинасымен N машинасын қалай симуляция жасауға болатынын көрсетеміз. Әрбір қадамда, M машина кез келген санақшыдан бірді алады немесе бірді қосады және 0-ге тексереді. Белгілі бір k үшін k-FA-ны пайдаланып, ондай машинаны жеңіл симуляция жасауға болады, бұл жерде санағыштың мәні ешқашан n-нен аспайды деп шарт қоя отырып, санағыш үшін бас тиектің орналасуын пайдаланамыз.

Алдымен, келесі санағыш амалдарын қалай іске асыруға болатынын көрсетеміз:

• Санағыш мәнін көшіру;
• Санағыш мәнін екі есе арттыру;
• Санағыш мәнін екіге бөлу;
• Санағыш мәні жұп болатынын тексеру;
• Бір санағыштың мәніне екінші санығыш мәнін қосу немесе алу.

Санағыш *c*-ның мәнін көшіру үшін екі кездейсоқ *d* және *e* санағыштарын нөлге айналдырамыз, содан кейін *c*-ны бірнеше есе азайтамыз және бір мезетте *d* мен *e*-ны арттырамыз. *c*-ны екі есе арттыру үшін, алдымен кездейсоқ санақшы *d*-ны нөлге айналдыру қажет және *c*-ны бірнеше есе азайту, ал *d*-ны екі есе арттыру қажет. Екіге бөлу де осы тәсілмен жасалады. Жұптыққа тексеру үшін санағышты екіге бөлеміз де, бір қалды ма соны тексереміз. Жалғау үшін бір санағышты арттырамыз, ал екіншісін кемітеміз. Алу үшін екеуін де азайтамыз.

Кез келген конфигурация *N* үшін, *N*-нің жұмыс таспасының мәнін $c\log n$ -битті бинарлы сан ретінде қарастыруға болады. Осы санды әрқайсысында $\log n$ бит болатын етіп, *c* блокқа бөлеміз, және осы $\log n$ битті сандарды *M*-нің *c* санағышына жазамыз. *M*-нің басқа санағышы *N*-нің жұмыс таспасының бас тиегінің орнын сақтау үшін пайдаланылады: жұмыс таспасындағы блок *N*-мен сканерленіп болғанан кейін, *M*-нің соңғы басқаруында есте сақталады және *i*-дің орны блокта 2^i санағыштағы сияқты көрсетілген. *N*-нің енгізу бас тиегінің орны *M*-нің енгізу бас тиегінің орны арқылы дәл өрнектеледі. *M*-нің басқа шектеулі санағыштары уақытша есте сақтау үшін қолданылады. *N*-нің күйі *M* -нің күйінде көрсетілген.

N-нің жүрісін симуляция жасау үшін ағымдағы уақытта *N*-мен екі таспада сканерленетін символдарды *M* машина білуі қажет. Ол енгізу таспасынан символдарды тікелей оқи алады. Жұмыс таспасы үшін ол санның *i*-ші битін анықтауды және өзгеруді жасай алуы қажет, бұл *i* белгілі бір санағыш *c*-да сақталады, мұндағы, 2^i басқа *d* санағышта есте сақталатын сан. Айтқанды іске асыру үшін алдымен, *c* және *d*-ның көшірмесін жасап алу қажет, яғни олардың мәнін жоғалтып алмауымыз керек. *d*-да 1 алынғанша, *c* мен *d*-ны бірнеше мәрте теңдей екі бөлікке бөлеміз, осыдан кейін ғана *c* жұп па соны тексереміз. Бұл *c*-ның *i*-ші битін анықтайды. Бастапқы *c*-ның мәнін пайдаланып, бастапқы *d*-ның мәнін қосу немесе алу арқылы, *c* -ны өзгерте аламыз.

3. $P = NP$ деп ұйғарайық. $L \in NEXPTIME$ болсын, 2^{n^c} уақытпен шектелген бейдетерминирлі Тьюринг M машинасында қабылдау-ашық деп айталық. Және

$$\hat{L} = \{x\#^k \mid x \in L \qquad k = 2^{|x|^c}\}$$

болсын. Сонда $\hat{L} \in NP$, ал енгізу $x\#^k$-де #-ның саны $2^{|x|^c}$ болатынына сенімді болу үшін, біз #-ның қанша дана екенін санауымызға болады, осыдан кейін оларды жойып жібереміз де, *M*-ді іске қосамыз. Ұйғарым $P = NP$-ға сай, $\hat{L} \in P$ болады, полиномиалды уақытты детерминирлі N машинасында қабылдау-ашық деп айтамыз. Сонда $L \in NEXPTIME$, өйткені енгізу *x*-те #-ның санын $2^{|x|^c}$-ге дейін арттыруымызға болады. N-ді іске қосамыз.

2-үй жұмысының шешімі

1. Егер f функциясы logspace түрлендіруі арқылы есептелінсе, онда $|f(x)|$-ті $|x|$-тің полиномы ретінде жазуға болады, өйткені түрлендіргіш конфигурацияны қайталамас бұрын полиномиалды уақыттан артық жұмыс жасай алмайды. M және N-нің logspace түрлендірулері арқылы сәйкес f және g функциялары есептелінеді делік. $g(f(x))$-ді есептеу үшін енгізу $f(x)$-те N-нің есептеуін симуляция жасаймыз. Жол $f(x)$ алдын ала есептелінбейді, бірақ сұратымда (запроста) символ артынан символ жіберу арқылы N-ге жеткізіледі. N машина өз енгізуінің i-ші символын оқығысы келген әрбір жағдайда, нөлден бастау алатын x енгізуінде M-ді симуляция жасайтын программаға i саны жеткізіліп беріледі. Ол барлық шығатын символдарды оқиды және оқылған i-ге дейінгі символдарды ескермейді, ал i-ші символды шақырушы программаға қайтарады. M және N-нің жұмыс таспасын сақтайтын жұмыс таспасынан жеткілікті кеңістік бөлінуі қажет және $|f(x)|$-ке дейін санай алатын санақшы да керек болады. Осылар жеткілікті. Терең ақпарат алам десеңіз [63, 13.1- және 13.2-леммаларды] қараңыз.

2. Алдымен, MAZE проблемасының келтірімділігінің әсерінен алынатын 2SAT-тің $\text{co}-NLOGSPACE$-күрделі болатындығын дәлелдеп аламыз. Бұл 2CNF орындалмаушылығындағы комплементарлылық проблемасының $NLOGSPACE$-күрделігі деген атпен белгілі дүние. MAZE проблемасының үлгісі $G=(V,E,s,t)$ арқылы берілген, оны V-ны логикалық айнымалылар жиыны деп қабылдаймыз және 2CNF формуланы

$$\varphi G \quad = \quad s \wedge \big(\bigwedge_{(u,v)\in E} (u \to v)\big) \wedge \neg t$$

қарастыратын боламыз.

Егер G-да s тен t-ға дейін жол бар болса, онда осы жолдың қабырғаларына сәйкестендірілген φ_G клоздар $s \to t$ меңзейді, сондықтан φ_G дегеніміз $s \wedge (s \to t) \wedge \neg t$ болып шыға келеді, ал ол орындалмайды. Керісінше, егер s -тен t -ға дейін жол жоқ болса, онда s -тен жете алатын барлық төбеге 1 санын меншіктейміз, ал басқа айнымалыларға 0 меншіктейміз. Бұл меншіктеу φ_G -ді қанағаттандырады, ал s -ке 1 меншіктелетіндіктен, t -ға 0 меншіктеледі. Және u -ға 1 меншіктелетін, ал v -ға 0 меншіктелетін ешқандай клоз $u \to v$ табылмайды. Сондықтан $\text{co} - NLOGSPACE$ үшін 2SAT күрделі болып табылады.

Енді 2SAT $\text{co} - NLOGSPACE$ -да жататынын дәлелдейміз немесе эквивалентті, 2CNF орындалмаушылығы $NLOGSPACE$ -те жататынын дәлелдесе де болады. 2CNF -тің $\mathcal{B}$ формуласы берілген, $\mathcal{B}$ -ның клоздарында екіден көп литерал болмайды және кез келген (u) түріндегі клозды $(u \vee u)$ клозымен алмастыру арқылы, жалпылықты сақтай отырып екі литерал бар деп тұжырым жасайық. Енді әрбір екі литералды $(u \vee v)$ клозды ішкі текст жұбы

$$(\neg u \to v) \quad \text{және} \quad (\neg v \to u) \tag{2}$$

ретінде қарастырамыз. Төбелері әрбір литералға арналған және бағытталған қабырғалары ішкі текст (2)-ге сәйкестендірілген, $G = (V, E)$ бағытталған графын тұрғызамыз. Келесі тұжырымды қиналмай-ақ дәлелдеуге болады (қараңыз, мысалы, [75, 119-бет]): сонда тек сонда $\mathcal{B}$ орындалмайды, егер комплементарлы екі литералды u, $\neg u$ қамтитын цикл G табылатын болса. Соңғы шартты $NLOGSPACE$ -те тексеруге болады, оны былай іске асыруға болады: u -ды таңдап аламыз, осыдан кейін таңдаймыз және циклді бақылаймыз, мұндағы мақсат – оның цикл екендігіне көз жеткізу және u мен $\neg u$ -ларды қамтитынын тексеру. Бұл жерде тек logspace талап етіледі ол u -ды сақтау үшін керек, біздің орнымыз циклде және бастапқы нүкте де циклде (соңы!).

3. Формуланы рекурсивті келесі түрде бағалаңыз. Түбірден бастаймыз. Ішкі формуланы $\varphi \wedge \psi$ бағалау үшін алдымен, φ -ді бағалаймыз; егер мән 1-ге тең болса, онда бүкіл өрнектің мәні ψ -дің мәні болады, кері жағдайда ол 0-ге тең болады да ψ -ды бағалаудың қажеті болмай қалады. Дуалды, $\varphi \vee \psi$ -ді бағалау

үшін алдымен φ-ді бағалаймыз; егер мән 0-ге тең болса, онда бүкіл өрнектің мәні ψ -дің мәні болады, олай болмаса, ол 1-ге тең, сондықтан ψ -ді бағалаудың қажеті жоқ. $\neg\varphi$-ді бағалау үшін φ -ді бағалаймыз және нәтижені теріске шығарамыз. Бұтақты аралап шығу үшін бізге тек шектеулі саусақ қажет болады, сондықтан оны logspace-те жасауға болады.

4. Айталық,

$$\begin{array}{lcl} B & = & \{\#b_k(0)\#b_k(1)\#b_k(2)\#\cdots\#b_k(2^k-1)\# \mid k \geq 0\} \\ B_j & = & \{\#u_0\#u_1\#\cdots\#u_{m2^j-1}\# \mid m \geq 0,\ b_j(i) \equiv u_i \pmod{2^j}, \\ & & \qquad\qquad\qquad\qquad 0 \leq i \leq m2^j\text{-}1\} \\ F_k & = & \#0^k(\#((0+1)^k - 0^k - 1^k))^*\#1^k\#, \end{array}$$

болсын, мұндағы, $b_j(i)$ арқылы $i \bmod 2^j$-нің бинарлы j-битті түрі белгіленген. B_j-дағы жолдар ұзындығы j-ден кем емес жолдар тізбегі $u \in (0+1)^*$-дан құралады. Олар u-дың тізбектегі жолының төменгі ретті j биті, белгілі бір m үшін 0-ден $m2^j-1$ -дейін өзгеретін сандар тізбегі $\bmod 2^j$ түрінде өрнектелетіндей етіп, $\#$ мен бөлінген. F_k-ның жолдары $\#$-мен ажыратылған ұзындығы k тізбектелген жолдан тұрады. Оның бірінші жолы 0^k, ал соңғы жолы 1^k және аралық жолдардың ешқайсысы не 0^k немесе 1^k болмайды. Келесі қамтуларға назар аударыңыз:

$$B_0 \supseteq B_1 \supseteq B_2 \supseteq \ldots \supseteq B_k,$$

мұнда

$$B = B_k \cap F_k = \bigcap_{j=0}^{k} B_j \cap F_k.$$

Берілген жол x, жиын B-да жата ма соны білу үшін, орналасу ретіне қарай B_0, B_1, ... , B_k, F_k жиындарында x жата ма соны тексереміз. Біз бұл амалды көп кеңістікті пайдаланбайтын етіп атқарамыз, егер енгізу B-да жатпаса да осы принцип сақталады. B_j жиынын $\log j$ кеңістігінде тануға болады. Тізбектелген жолдар u және v үшін, u және v-ның сәйкес төменгі ретті j биттерін салыстыра отырып, j биттер тізбектелген бүтін сан $\bmod 2^j$ түрінде ме соны тексереміз. Жақын орналасқан $\#$ битінен оң жақ шетке дейінгі қашықтықты есептеу үшін $\log j$ кеңістік керек. Оған қоса, u -дың бірінші жолы $0 \mod 2^j$ -де, ал соңғы жолы -1 $\mod 2^j$ болатынына көз жеткізуіміз қажет. $j = 0, 1, 2, \ldots ,$ үшін таспаның $\log j$ ұяшығын қалдырамыз және B_j -ға жататындыққа тексереміз. B_j -дің барлық жолдарының ұзындығы $j2^j$ -ден кем емес, сондықтан егер тексеру сәтті шықса, онда біз тек $\log j \le \log \log n$ кеңістікті пайдаландық. Егер белгілі бір уақытта тест сәтсіз біткен j -ге тап бола қалсақ, онда сөзсіз қабылдау-жабық; бірақ осы жағдайдың өзінде $x \in B_{j-1}$ болғандықтан, біз тек

$$\log j \quad \le \quad 1 + \log(j-1) \quad \le \quad 1 + \log \log n$$

кеңістікті пайдаландық. Егер барлық тексеру $x \in B_j$-те сәтті өтсе, онда біз $\log k$ кеңістік бөліп қалдырдық деген сөз. Ол кеңістік F_k -да жатушылықты тексеруге жеткілікті болады.

3-үй жұмысының шешімі

1. (a) x жолындағы $\#L(x)$ (сәйкес $\#R(x)$) сол жақша саны (сәйкес, оң жақша) болсын. Индукцияны пайдаланып келесі тұжырымды дәлелдеуге болады: егер тек егер

(i) $\#L(x) = \#R(x)$, және
(ii) x-тын әрбір префиксі y үшін , $\#L(x) \geq \#R(x)$,

болса, онда x теңгеріледі. Сондықтан $\#L(y) - \#R(x)$-ты санай отырып, солдан оңға қарай сканерлей аламыз.

(b) Егер x жолы жақша түріне тәуелсіз түрде (a) бөліктегі (i) және (ii) шарттарын қанағаттандырса және әрбір сәйкестендірілетін жұп бірдей тип болса, сонда тек сонда екі типті x жолы теңгеріледі. Санамалау арқылы сәйкестенетін жұптарды табуға болады. Егер тек егер жақша түрінен тәуелсіз түрде y (i) және (ii)-ді қанағаттандырса, онда $x[y]z$ жолындағы жақша сәйкестікті көрсетеді.

2. Ойынның бұл версиясы *ALOGSPACE* = *P* үшін толық болады. Өйткені төбелер қайтадан пайдалануы мүмкін, тақта позициясы ағымдағы уақытта тек қолданысқа түсетін төбелерден және жүріс кезегі кімде екен айтып отыратын биттерден тұрады. Алдынғы нұсқадан айырмашылық мынада, мұнда қай төбелер ойналып кетті, соны есте сақтауға міндетті емеспіз. Тақтаның қазіргі позициясын есте сақтау үшін, тек logspace орын қажет етеді, сондықтан бірінші ойыншы мәжбүрлі жеңіске жете ме соны алтернативті logspace машина анықтай алады, соның әсерінен проблема *ALOGSPACE* = *P*-да жатады.
Енді *P* үшін проблема күрделі болатынын көрсету үшін мәндер схемасының мәселесін осы мәселеге келтіреміз. Егер CVP-ның үлгісі берілсе, онда алдымен, оны ешқандай теріске шығаруы жоқ және қатаң кезектесуі жоқ үлгіге айналдырамыз. Теріске шығарудан құтылу үшін схеманың көлеңкесін жасаймыз:

Оригинал	Көлеңке
$c_i := c_j \wedge c_k$	$c'_i := c'_j \vee c'_k$
$c_i := c_j \vee c_k$	$c'_i := c'_j \wedge c'_k$
$c_i := 0$	$c'_i := 1$
$c_i := 1$	$c'_i := 0$
$c_i := \neg c_j$	$c'_i := \neg c'_j$

Осыдан кейін барлық $c_i := \neg c_j$-ды $c_i := c'_j$-ға және $c'_i := \neg c'_j$-ны $c'_i := c_j$-ге ауыстырамыз. Кезектесуді қатаң жасау үшін бір мезгілде әрбір $c_i := c_j \wedge c_k$ тұжырымды екі тұжырыммен $c_i := d \vee d$ және $d := c_j \wedge c_k$ алмастырамыз, мұндағы, d жаңа айнымалы және әрбір $c_i := c_j \vee c_k$ тұжырымды үш тұжырыммен алмастырамыз $c_i := d \vee e$ $d := c_j \wedge c_j$, $e := c_k \wedge c_k$, мұндағы, d және e жаңадан енгізілген айнымалылар. Бұл амал айнымалылар санын үш еседен аспайтындай арттырады.

Енді география ойынын жасайтын боламыз. Назар аударыңыз: егер ойыншы қарсыласты қапасқа қамаса, онда ол ұтады (төбенің жарты дәрежелі нәтижесі 0). Біз төбелері $\{c_i \mid 0 \le i \le n\} \cup \{\perp\}$ болатын граф өндіреміз, оның бағытталған қабырғалары (c_i, c_j) және (c_i, c_k) әрбір тұжырымға $c_i := c_j \wedge c_k$ немесе $c_i := c_j \vee c_k$ арналған, ал $(c_i, \perp)$ қабырғалары, әрбір тұжырым $c_i := 1$-ге арналған. Сонымен, тұжырым $c_i := 0$ схемада пайда болатын $\perp$ және c_i қапасты өрнектейтін болды. Бастапқы орын, ең үлкен индексті айнымалы – c_n. Егер схеманың мәні 1-ге тең болса, сонда тек сонда бірінші ойыншы мәжбүрлі жеңіске жетеді.

3. Ақпарат үшін шектеулі автоматтар жайлы [63] немесе [76]-ны қараңыз.

Алтернативті шектеулі автоматтар сияқты детерминирлі автоматты, F-ті соңғы күйлер жиыны емес соңғы күйлер жиынының сипаттама функциясы ретінде қарастыру техникалық ыңғайлы саналады. Яғни, $F : Q \to \{0,1\}$, өйткені:

$$F(q) = \begin{cases} 1, & \text{if } q \text{ is a final state} \\ 0, & \text{otherwise.} \end{cases}$$

AFA -дан DFA -ны құрастыру үшін, жиын

$$A = (Q_A, \Sigma, \delta_A, F_A, \alpha_A)$$

AFA -мен берілген болсын, және $|Q_A| = k$ деп алайық. $Q_A \to \{0,1\}$ қатынастағы барлық функциялар жиыны Q_D болсын. DFA -ны анықтаймыз:

$$D = (Q_D, \Sigma, \delta_D, F_D, s_D),$$

мұндағы,

$$\delta_D(u,a)(q) = \delta_A(q,a)(u) \tag{3}$$

$$F_D = \alpha_A \tag{4}$$

$$s_D = F_A \ . \tag{5}$$

AFA -дан DFA -ны құрастыру үшін, жиын

$$D = (Q_D, \Sigma, \delta_D, F_D, s_D)$$

DFA -мен берілген және $|Q_D| = k$ болсын. Q_A өлшемі $\lceil \log k \rceil$ болатын кез келген жиын болсын және Q_D-ның әрбір элементін ерекше функция $Q_A \to \{0,1\}$ -мен танимыз. AFA -ны анықтаймыз:

$$A = (Q_A, \Sigma, \delta_A, F_A, \alpha_A),$$

мұндағы δ_A, F_A және α_A параметрлері (3)-(5) орындалатын етіп анықталады. Ал $u \notin Q_D$ үшін $\delta_A(q,a)(u)$ -ны еркін анықтаймыз. Екі мәліметте де, $|x|$-те индукцияны пайдаланып, кез келген $q \in Q_A$, $u \in Q_D$, және $x \in \Sigma^*$ үшін

$$\hat{\delta}_D(u,x)(q) = \hat{\delta}_A(q, \mathrm{rev}\, x)(u)$$

болатынын көрсетуге болады, мұндағы, $\hat{\delta}_D : Q_D \times \Sigma^* \to Q_D$ қатынасы δ_D -ның көп баспалдақты версиясы. Ол енгізу жолының ұзындығынан индукциямен алынады:

$$\hat{\delta}_D(u,\varepsilon) \stackrel{\text{def}}{=} u$$

$$\hat{\delta}_D(u,xa) \stackrel{\text{def}}{=} \delta_D(\hat{\delta}_D(u,x),a).$$

Сондықтан:

$$\begin{aligned}
x \in L(D) &\Leftrightarrow F_D(\hat{\delta}_D(s_D,x)) = 1 \\
&\Leftrightarrow \alpha_A(\hat{\delta}_D(F_D,x)) = 1 \\
&\Leftrightarrow \alpha_A(\lambda q.(\hat{\delta}_D(F_A,x)(q))) = 1 \\
&\Leftrightarrow \alpha_A(\lambda q.(\hat{\delta}_A(q,revx)(F_A))) = 1 \\
&\Leftrightarrow \mathrm{rev}\, x \in L(\mathrm{A}).
\end{aligned}$$

4-үй жұмысының шешімі

1. Алдымен, $A(n)$ және $S(n)$ кеңістіктерді конструктивті деп ұйғарайық. $A(n)$-кезектескен шектеулі, $S(n)$-кеңістікті шектелген машина M болсын. Ұзындығы n енгізулер үшін M-нің конфигурациялар жиыны C_n болсын. Тек M-нен тәуелді, тиянақты c тұрақтысы табылып, $|C_n| \le c^{S(n)}$ қатынасы орындалады.
Конфигурация универсалды немесе экзистенциалды бола ма соны $\mathrm{type}: C_n \to \{\wedge, \vee\}$ анықтайды дейік. Қабылдау-ашық конфигурация мұрагері жоқ универсал конфигурацияға, ал қабылдау-жабық конфигурация мұрагері жоқ экзистенциалды конфигурацияға жатады. Егер $\alpha, \beta \in C_n$ болып, α-дан β-ға дейінгі ұзындығы k-дан аспайтын есептеу траекториясы табылса, онда:

$$R(\alpha, \beta, k)$$

деп жазамыз, мұнда β-дан басқа, траектория бойындағы барлық γ конфигурациялар $\mathrm{type}(\gamma) = \mathrm{type}(\alpha)$ және $\mathrm{type}(\beta) \ne \mathrm{type}(\alpha)$ шарттарын қанағаттандыруы қажет. $\alpha, \beta \in C_n$ үшін, предикат

$$R\left(\alpha, \beta, k^{S(n)}\right)$$

бейдетерминирлі $S(n)$ кеңістігінде шешіледі, сондықтан ол Савич теоремасына сай детерминирлі кеңістік $S(n)^2$ -та шешіледі.
Сонда бастапқы экзистенциалды конфигурация α_0-де, ұзындығы n енгізу x үшін егер тек егер

$$\begin{array}{l}
\exists \alpha_1\ R(\alpha_0, \alpha_1, c^{S(n)}) \wedge \\
\quad \forall \alpha_2\ R(\alpha_1, \alpha_2, c^{S(n)}) \to \\
\quad\quad \exists \alpha_3\ R(\alpha_2, \alpha_3, c^{S(n)}) \wedge \\
\quad\quad\quad \forall \alpha_4\ R(\alpha_3, \alpha_4, c^{S(n)}) \to \\
\quad\quad\quad\quad \cdots \\
\quad\quad\quad\quad\quad Q\alpha_{A(n)}\ R(\alpha_{A(n)-1}, \alpha_{A(n)}, c^{S(n)})
\end{array}$$

болса, онда M-де қабылдау-ашық. Оны логикалық мәнді рекурсивті процедура $S(\alpha)$ арқылы тексеруге болады, ол процедура былай жұмыс жасайды. Егер α экзистенциалды болса, онда ол лексиграфикалық түрде барлық β-ны түгендеп шығып, $R(\alpha,\beta,c^{S(n)})$ және $S(\beta)$ болатын β бар ма соны тексереді. Ол Савич алгоритмін пайдаланып, біріншіні тексереді, егер оған қол жеткізсе, онда ол соңғыны рекурсивті шақыру арқылы тексереді. Осы тәсілмен, егер α универсал болса, онда ол барлық β-ны түгендеп шығады; осы процесте егер $R(\alpha,\beta,c^{S(n)})$ болса, онда $S(\beta)$ болады.

Әрбір процедураны рекурсивті нақтылауда (конкретизация) рекурсивті шақыру арқылы ағымдағы конфигурация α-ны сақтау үшін $S(n)$ кеңістік қажет болады, ал рекурсия тереңдігі $A(n)$, сондықтан осы мақсат үшін барлығы $A(n)S(n)$ кеңістік қажет болады. Осыдан басқа, Савичтің R процедурасын есептеу үшін $S(n)^2$ кеңістік керек болады, ол қайтадан қолданылуы да мүмкін. Осының барлығы жалпы кеңістіктік шектеу $A(n)S(n)+S(n)^2$-ты береді.

$A(n)$ және $S(n)$ конструктивті кеңістік болмаған жағдайда, біз итеративті түрде A және S-тің барлық мәнін

$$AS+S^2=1,2,\ldots\ .$$

арқылы сынақтан өткіземіз.

2. $PSPACE$-те,

$$\begin{aligned}\Sigma_k^{PSPACE} &= STA(n^{O(1)},*,\Sigma k)\\ \Pi_k^{PSPACE} &= STA(n^{O(1)},*,\Pi k),\end{aligned}$$

қатынастарын орнату арқылы иерархияны анықтаймыз. Осыдан кейін, алдынғы жаттығуға сүйенсек:

$$\begin{aligned}\Sigma_k^{PSPACE} &= STA(n^{O(1)},*,\Sigma k)\\ &= \bigcup_{c>0} STA(n^c,*,\Sigma k)\\ &\subseteq \bigcup_{c>0} DSPACE(kn^c+n^{2c})\\ &\subseteq \Sigma_0^{PSPACE}.\end{aligned}$$

3. $H_\omega = \{y\$z \mid z \in G_{\#(y)}\}$ жиыны *PSPACE*-де жатады, өйткені енгізу $y\$M\$x\$^d$-де универсалды алтернатив машина *x*-те M -ді симуляция жасауы мүмкін, ол симуляцияның әрбір қадамында бір $\$$-ты белгілейді және әрбір кезектесуде бинарлы сан *y*-ті азайтып отырады. M -нің әрбір қадамы және симуляцияға керек M -нің ұзындығы *d*-да полиномиалды уақыттан көп уақыт талап етпейді және қадам саны *d*-дан көп емес. Егер не уақыт шектеуінен немесе кезектесу саны шектеуінен асып кетсек, онда машинаның симуляция жасау процесіне қабылдау-жабық.
PSPACE үшін H_ω күрделі болатынын дәлелдеу үшін *APTIME*-дегі кез келген жиынды H_ω-ға келтірсек жеткілікті болады. n^c уақытта жұмыс жасайтын кез келген алтернатив машина M болсын. Сонда бейне:

$$x \quad \mapsto \quad y\$M\$x\$^{|x|^c}$$

$L(m)$-нен H_ω-ға дейінгі $\leq_{\mathrm{m}}^{\log}$ келтірімділікті бейнелейді, мұндағы, $\#(y) = |x|^c$.

4. Жиын

$$G_k = \left\{ M\$x\$^d \mid M \text{ - да } \vee \text{-дан бастап, } d \text{ кеңістікте және } k \text{ кезектесуде } x \text{ -ке қабылдау ашық} \right\}$$

$STA(n^2,*,\Sigma k)$ -да жатады. Енгізу $M\$x\$^d$-де, *x* енгізуі үшін *M*-ді симуляция жасаймыз. Ал симуляцияланатын машинаны таспада көрсету үшін таспаның әрбір символы $|M|$-нен артық кеңістік талап етпейді (келесі шарт міндетті: таспа символы машина сипаттамасында дәл көрсетіледі) және *M*-нің лентасын көрсету үшін симуляциялау $d \cdot |M|$-дан артық кеңістік талап етпейді. Σ_k^{PSPACE} үшін жиын G_k -да күрделі болады: n^c кеңістігінде жұмыс жасайтын М машина кез келген Σ_k машина болсын. Сонда бейне

$$x \quad \mapsto \quad M\$x\$^{|x|^c}$$

$L(m)$-нен G_k -ға дейінгі $\leq_{\mathrm{m}}^{\log}$ келтірімділікті құрайды.

Енді:

$$
\begin{aligned}
G_\omega &= \{y\$z \mid z \in G_{\#(y)}\} \\
&= \Big\{ y\$M\$x\$^d \mid M \text{ -да } \vee \text{ -дан бастап, } d\text{-дан көп емес кеңістікте және } \#(y) \text{ кезектесуде } x\text{-ке қабылдау ашық} \Big\}.
\end{aligned}
$$

Жоғарыда қарастырылған проблемалардың біреуі секілді симмуляция көмегі арқасында G_ω жиыны *APSPACE* = *EXPTIME*-те жатады. G_ω-дың *APSPACE* үшін күрделі болатынын көрсету үшін: *M* машина n^c кеңістікті шектелген ATM машина болсын. Және тек *M* машинасынан тәуелді *e* тұрақты саны табылып, ұзындығы *n* болатын енгізулердегі *M*-нің әртүрлі конфигурациялары e^{n^c} -мен шектелген болсын. Сонда бейне:

$$x \mapsto y\$M\$x\$^{|x|^c},$$

$L(m)$-нен G_ω-ға дейінгі $\leq_{\mathrm{m}}^{\log}$ келтірімділікті құрайды, мұндағы, $\#(y) = e^{|x|^c}$.

5-үй жұмысының шешімі

1. $\Pi_k^{\mathrm{p}} \subseteq \Sigma_k^{\mathrm{p}}$ болады деп ұйғарайық. 10.2-теоремаға сай, белгілі бір c үшін кез келген жиын $A \in \Sigma_{k+1}^{\mathrm{p}}$ -ны

$$A = \{x \mid \exists y \mid y \mid \leq |x|^c \wedge R(x,y)\}$$

түрінде жазуға болады, мұндағы, R Π_k^{p} предикат. Қабылданған ұйғарымға сүйенсек, екінші жағынан R Σ_k^{p}-предикат болады, сондықтан $\exists y \mid y \mid \leq |x|^c \wedge R(x,y)$ орындалады әрі $A \in \Sigma_k^{\mathrm{p}}$ болады. Мұнда күйреу артынан k арқылы индукция кетеді.

2. (a) Бұл жерде қабылданған ұйғарымдарда $\mathrm{SAT} \in P$ қатынасы дұрыс болатынын дәлелдейік. Ұзындығы n енгізу үшін x-те схема B_n-ны генерация жасаймыз және $B_n(x)$-ді бағалаймыз. Қабылданған полиномиалды уақыт біртектілігі жайлы ұйғарымға сай схема генерациясы P-да жасалуы мүмкін және бағалау CVP-ның қарапайым үлгісі, сондықтан ол P-да жатады. Бұл схема мәнінің проблемасы болып табылады, ол 6.1-теоремаға сай P-толық болады.

(b) 1-жаттығуды пайдаланып, $\Pi_3^{\mathrm{p}} \subseteq \Sigma_3^{\mathrm{p}}$ болатынын көрсетсек жеткілікті болады. $A \in \Pi_3^{\mathrm{p}}$ болсын және $L\left(M^{\mathrm{SAT}}\right) = A$ орындалғандай, n^k-уақытпен шектелген Π_2^{p} оракул машина M болсын. A-ға қабылдау-ашық Σ_3^{p} машинаны келесі түрде тұрғызамыз. Сонымен, $m = n^k$ болсын. Ұзындығы n болатын енгізу x-те Σ_3^{p} есептеу көмегімен B_m-ді тұрғызамыз: схеманы табу, содан кейін ол схема коррект бола ма соны тексеру қажет. Егер $|y| = m$ орындалатын y табылып, логикалық формуланың барлық кодтауында

$$B_m(y) = 1 \quad \Leftrightarrow \quad y \in \mathrm{SAT}. \tag{6}$$

қатынасы орындалса, онда схема коррекці болады.

Схема $\vee$-тармақ арқылы танылады, сонда ұзындығы m болатын формула y-тің барлық жиыны $\wedge$-тармақ көмегімен

генерацияланады және соңында, (6)-шы шартты Δ_2^{p} -де тексеруге болады ($\Delta_2^p = P^{NP} \subseteq \Sigma_2^p \cap \Pi_2^p$ болатынын еске саламыз).
Енді Σ_3^{p}-те $x \in A$ болатынын тексеру үшін, біз жоғарыда жазылғандай $\vee$-тармақты пайдаланып, алдымен, B_m схемасын тануымыз қажет (кез келген корректі схема жарай береді), осыдан кейін екі шартты тексереміз:
(i) B_m корректі болады және
(ii) Оракул сұратымға жауап беру үшін B_m-ды пайдаланады, M -де x -ке қабылдау-ашық.
Осы екі фактор бірмезгілде Π_2^{p} -де тексеріледі.

3. Алдымен, MAZE мәселесін бейдетерминирлі бір санақшылы автоматтың қамтылу проблемасына келтіреміз. Бұл жай, қамтылу проблемасы $NLOGSPACE$ -те күрделі болатынын көрсетеді. Егер екі ерекше төбелер *s* және *t*-дан тұратын графтан құралған, MAZE проблемасының үлгісі берілсе, онда оны төбе аттары унитарлы нотацияда (яғни *n* саны 0^n түрінде көрсетілген) жазылған бүтін сандар болатын үлгіге айналдырамыз. Бұл logspace-те жасалуы мүмкін. Осындай кодтауда бейдетерминирлі бір санақшылы автомат, төбе аттарын сақтауға арналған өзінің санақшыларын пайдаланып, *s*-тен *t*-ға дейінгі траекторияны (ізді) тани алады, осы төбеде автомат ағымдағы уақытта болады.
Енді біз қамтылу проблемасы $NLOGSPACE$ -те жататынын көрсете аламыз. Бұл проблеманың ең күрделі бөлігі; оның киындығы санақшының шексіз болуында жатыр. Дегенмен, егер қабылдау-ашық траектория бар болса, онда $O(n^3)$ -тан аспайтын бір ұзындық бар, мұндағы *n* – енгізудің өлшемі. Осы жағдай logspace ТМ-да, санақшыны екілік жүйеде жұмыс таспасында сақтай отырып, бір санақшылы автоматты симуляция жасауға мүмкіндік береді. Және симуляцияланған қадамдарды санап отырады және егер симуляция бөлінген уақытта аяқталмаса, онда ол тоқтайды.
M бір санақшылы бейдетерминирлі автомат болсын. Жалпылық ережесін сақтай отырып, егер *M*-де қабылдау-ашық болғысы келсе, онда ол қабылдау-ашық күйге көшер алдында өзінің санақшысын босатады деп ұйғарамыз. *M*-нің соңғы бақылауы күйінің саны *q* болсын. Ұзындығы *n*-ге тең енгізуде $m = q(n+2)$ түрлі күйлердің

мүмкін конфигурациясы және енгізу бас тиегінің осынша мүмкін орны бар. $|x|=n$ болатын енгізу x-те қабылдау-ашық болатын α есептеу траекториясы бар болады. Уақыттың t мезетіндегі санақшының мәні $c(t)$ болсын және t уақыттағы α -дағы жұп $q(t)$ (енгізу бас тиегінің күйі) болсын. Егер $s<u<t$ диапазонында барлық u үшін $s<t$, $c(s)=c(t)$ және $c(u)>c(t)$ орындалса, онда (s,t) уақыт жұбы беттесетін интервал деп аталады. Осыған дейін, санақшыда сақталған мәндердің ішіндегі ең максималдысы N болсын және $c(t_N)=N$ орындалатын максималды уақыт t_N болсын. $1\le i<N$ үшін, t_N уақытының алдындағы $c(s_i)=i$ орындалатын ең кеш уақыт s_i болсын және t_N -нен кейінгі $c(t_i)=i$ орындалатын ең ерте уақыт t_i болсын. Сонда:

$$s_1<s_1<\cdots<s_{N-1}<t_N<t_{N-1}<\cdots<t_2<t_1$$

және (s_i,t_i) беттесетін интервал болады, ал $1\le i<N$. Егер $N>m^2+1$ болса, онда $1\le i<j<N$ табылып, $q(s_i)=q(s_j)$ және $q(t_i)=q(t_j)$ орындалады. Сондай жағдайда, α -ның s_i мен s_j арасындағы және t_j мен t_i арасындағы бөлігін жою арқылы алынатын, бұрынғыдан қысқа қабылдау-ашық есептеу траекториясы бар.

Біз x үшін қабылдау-ашық есептеудің санақшысының минималды мәні m^2+1-ден асып мән қабылдамайтынын көрсеттік. $m(m^2+2)$ -ден аспайтын күйлер конфигурациясы сондай-ақ осы саннан аспайтын бас тиектің орны бар, және есептеуде қолданылатын санақшының да мәні осы шектен аспайды. Егер әлде бір есептеудің ұзындығы $m(m^2+2)$ -тан асып кететін болса, онда қайталанылатын конфигурациялардың болғаны, сондықтан қысқа түрдегі қабылдау-ашық есептеуді алу үшін, есептеудің біраз бөлігін алып тастауға болады. Сонымен, минимал ұзындықты есептеудің ұзындығы $m(m^2+2)=O(n^3)$ -ден аспайды.

6-үй жұмысының шешімі

(a) Егер $S(n)$ кеңістікті конструкция болса, онда ұзындығы n-ге тең болатын кез келген енгізуде өзінің жұмыс таспасында дәл $S(n)$ кеңістік бөлетін машина M бар болады, ол машина $S(n)$ -нен артық кеңістік пайдаланбайды және тоқтайды. Егер M -нің q күйі және жұмыс таспасының d символы болса және егер $S(n) \le o(\log n)$ болса, онда жұмыс таспаның мәнін және жұмыс таспаның бас тиегінің орнын сақтайтын күйлердің конфигурация саны $qS(n)d^{S(n)}$-нен аспайды, ол сан жеткілікті үлкен n үшін $n/2$ -ден кіші болады. Егер M машина әйтеуір бір кезде ортаға дейін өте ұзын 0^n енгізу жолын сканерлесе, онда ол ортаға жеткенге дейінгі уақытта, тізбеде (цикл) болуы қажет. Яғни сол жақ шеткі маркерге соңғы рет жолыққаннан кейін және ортаға дейін жеткенге дейін, және енгізу таспасының екі түрлі i және j, $i < j < n/2$ орналасуын сканерлеген кезде ол бұрынғы конфигурация c болуы қажет, және олардың ортасындағы сол жақ шеткі маркерді көрмеуі қажет. Сонымен, M машина i-ден кейін енгізетін жұмыс таспасындағы 0-ден басқа ештеңе көрмейтіндіктен, ол j-ден бастау алып, i-де өтіп шыққан конструкциялар тізбегін тағы да қайталап атқарады, осы процесс оң жақ шеткі маркер көрінгенше қайталана береді, ол маркер жұмыс түрін өзгертуі мүмкін. Егер 0-ұзындықты жолды $p_1 = j - i \le n/2$ -ге еселі етіп апарып қойсақ, онда машина өзгешелікті айта алмайды; ол оң жақ шеткі маркерге өзгеріссіз қолданыстағы конструкциямен жетеді. Егер машина оңнан солға қарай сканерлесе де, осы айтқанымыз дұрыс болады; кері бағытта машина ортаға жеткен уақытта ол тізбенің $p_2 \le n/2$ периодында болады, т.с.с. Сондықтан кез келген $m \ge n$ үшін ұзындығы $m!$ болатын 0-дер жолын енгізуге болады, ол осы тізбелердің мүмкін периодтары p_i -дің бәріне еселі болады және машина өзін басқаша сезіне алмайды; жеке жағдайда, ол өзінің жұмыс таспасынан артық кем кеңістік бөле алмайды. Бұл жағдайда, барлық $m \ge n$ үшін $S(n+m!) = S(n)$ орындалады деп айтылады.

(b) $\lim\ \inf \lceil \log\log n \rceil = \infty.$

2. Проблема $PSPACE$-те жататынын көрсету үшін M-де қабылдау-ашық емес жабық жолды табуымыз және осы жолға қабылдау-ашық емес екендігіне көз жеткізуіміз қажет. Керекті жолды табу үшін M-нің алғашқы күйіне құмалақ қоюдан бастаймыз, осыдан кейін енгізу жолын символдан символға жылжып тексереміз, құмалақ та M-нің күйінде жылжып отырады, осы арқылы осыған дейін тексерілген жолдарда бастапқы күйден бастап қол жететін барлық күйлер белгіленеді. Бізге табылған жолды есте сақтау мақсат емес, осы уақыттағы M машинасының күйі ғана керек. Егер әйтеуір бір уақытта қабылдау-ашық күйдегі құмалақ жоқ болатын жағдайға келсек, онда бізде қабылдау-ашық. Ал конфигурация құмалақ полиномиалды кеңістікте көрсетілетіндіктен, бұл бейдетерминирлі $PSPACE$ есептеу. Оны Савич теоремасының көмегі арқылы детерминирлі жасауға болады. (Ескерту. Келесі тұжырым дұрыс емес: егер NFA-де ұзындығы күй санына тең немесе аз барлық жолға қабылдау-ашық болса, онда барлық жолға қабылдау-ашық, әуелі бір әріпті алфавит болса да қабылдау-ашық. Мысалы, алғашқы күйі, жұпталған өзара жай $p_1, p_2, \ldots, p_n$ ұзындықты, n қиылыспайтын тізбеге баратын автоматты қарастырайық. Алғашғы күйден басталатын тізбесі ең ұзын жалғыз күйден басқасын қабылдау-ашық жасайық. Сонда барлығы $1 + p_1 + p_2 + \ldots + p_n$ күй бар, бірақ ең қысқа қабылдау-ашық емес жолдың ұзындығы $p_1 p_2 \cdots p_n$).

$PSPACE$ үшін проблема күрделі болатынын көрсету үшін кез келген детерминирлі n^k кеңістікті шектеулі TM-ды N машина деп алайық. Жалпылықты сақтай отырып, N-нің уникалды (жалғыз) қабылдау-ашық конфигурациясы бар деп ұйғарайық. Егер енгізу x берілсе, онда біз $O(n^k)$ күйге ие болатын бейдетерминирлі шектеулі автомат M-ді құрастырамыз, ол автоматта енгізу x-те N-нің қабылдау-ашық емес есептеу тарихының барлық жолына қабылдау-ашық. Сонда, егер тек егер N-де x-ке қабылдау ашық емес болса, онда $L(F) = \Sigma^*$ болады. (Осы арқылы, біз соңында $L(N)$-ді $\{M \mid L(M) \neq \Sigma^*\}$ -ға келтіреміз).

Ұзындығы n енгізу x-тегі N-нің қабылдау-ашық есептеу тарихы

$$\#\alpha_0\#\alpha_1\#\alpha_2\# \;\cdots\; \#\alpha_{m-1}\#\alpha_m\# \tag{7}$$

формасындағы жол болады, мұндағы, әрбір α_i белгілі бір шектеулі алфавит Δ -да анықталған ұзындығы n^k болатын жол. Ол енгізу *x*-те N конфигурациясын төмендегідей кодтайды:

(i) x -те α_0 N -нің алғашқы конфигурациясы болады;

(ii) x -те α_m N -нің қабылдау-ашық конфигурациясы және

(iii) N -дегі алмасу ережесі бойынша әрбір α_{i+1} α_i -ден алынады.

Егер жол қабылдау-ашық есептеу тарихы болмаса, онда не (7)-ші форма жоқ немесе үш шарттың (i), (ii), (iii) біреуі орындалмайды. Осылардың ішінен қайсысын тексеру қажет екенін NFA M бейдетерминирлі тауып отырады. Енгізу жолында (7)-ші форма жоқ екенін анықтау басқа тексерулерді қажетсінеді, олар: енгізу жолы регулярлы жиында $(\#\Delta^{n^k})^*\#$-да жатпайды және әртүрлі форматты басқаша қарапайым тексеру (конфигурацияда N -нің тек бір күйі болуы, әрбір конфигурация соңғы маркермен басталатыны және аяқталатыны және т.с.с). Осының бәріне M -нің $O(n^k)$ күйі қажет. Шарттар (i) немесе (ii)-ні тексеру, енгізу бастала ма немесе ұзындығы n^k болатын тиянақты анықталған жолмен аяқтала ма деген қарапайым тексерулерді шақырады. Бұл жолдар M -нің соңғы бағалауында кодталады. Сөз соңында, (iii)-ті тексеру үшін Кук-Левин теоремасы дәлелдемесінен еске түсірейік: α_i -дің $(j-1)$ -ші j -шы және $(j+1)$ -ші символдарымен және α_{i+1} -нің j -шы символымен байланысқан шектеулі жиынды локалды шарттар бар еді. Яғни егер тек егер барлық j, $i<j<n^k$ үшін осы локалды шарттар орындалса, онда (iii) шарт орындалады деп еді. Локалды шарттар N -нің сипаттамаларына ғана тәуелді. (iii)-тің бұзылатынын тексеру үшін M барлық енгізуді сканерлейді және белгілі бір уақытта қай жерде бұзылу жүріп жатқанын бейдетерминирлі анықтайды. Ол өзінің соңғы бақылауында келесі кезектегі үш символды есте сақтайды, егер келесі символ тривиал болмаса, онда ол келесі n^k символды өткізіп жібереді және қабылдау-ашық күйге енеді.

3. *NLOGSPACE* үшін проблема толық болады. Проблеманың *NLOGSPACE*-те жататынын көрсету үшін, қабылдау-ашық және қабылдау-жабық күйлермен алмасып алып, қабылдау-ашық жиын бос емес пе деп сұрасақ жеткілікті болады. Егер тек егер осы

айтқанымыз дұрыс болса, онда бастапқы күйден белгілі бір соңғы күйге апаратын жол табылады. Егер k соңғы күй табылса, онда олар *MAZE*-нің k үлгісі болып табылады.

Проблеманың *NLOGSPACE*-те күрделі болатынын көрсету үшін, біз *MAZE*-ні оған келтіреміз. Егер *MAZE*-дің үлгісі $G = (V, E, s, t)$ берілген болса, онда жалпылықты сақтай отырып, әрбір v төбеден шығатын жоқ дегенде бір қабырға бар деп санаймыз. Егер олай болмаса, онда v-дан s-ке қосымша қабырға тартамыз; оның s-тен t-ға бара алушылыққа ешқандай әсері жоқ. Кез келген төбеден шығатын максималды шығу дәрежесі m болсын және $\Sigma = \{0, 1, \ldots, m-1\}$ болсын делік. V-ның күйлерінен, енгізу алфавиті Σ, алғашқы күй s, ерекше соңғы күй t мен DFA M құрастырамыз. Және Σ элементтерінен алынған маркировкалы ауысулар: кез келген $a \in \Sigma$ және $v \in V$ үшін a меткалы v-ден шығатын тек жалғыз қабырға табылады деген жолмен құрылады (Қабырғадағы метка саны бірден көп болуы мүмкін). Енді, егер тек егер G-да s-тен t-ға дейін жол табылса, онда M детерминирлі және $L(M)$ бос емес болады.

7-үй жұмысының шешімі

1. Бұл PSPACE-толық проблемасы. Сондықтан әрбір бейтривиалды бірінші ретті теория PSPACE-күрделі болып табылады (49-шы аралас жаттығу), оның PSPACE-та жататындығын көрсету дәлелдеменің қызықты бөлігі болып табылады.
Нақты сандар k-кортежі $a_1,\dots,a_k$ және $b_1,\dots,b_k$ үшін $a_0=b_0=0$ болсын. Егер

$$a_{\pi(0)} \le a_{\pi(1)} \le \dots \le a_{\pi(k)}$$
$$b_{\pi(0)} \le b_{\pi(1)} \le \dots \le b_{\pi(k)}$$

орындалатындай алмастыру $\pi:\{0,1,\dots,k\}\to\{0,1,\dots,k\}$ бар болса (яғни, егер a-лар және b-лар бірдей реттілікпен кездессе) және барлық $0\le i\le k-1$ үшін,

$$\min\{2^m,\ a_{\pi(i+1)}-a_{\pi(i)}\} = \min\{2^m,\ b_{\pi(i+1)}-b_{\pi(i)}\},$$

болса, онда

$$a_1,\dots,a_k \equiv_k^m b_1,\dots,b_k$$

деп анықтаймыз.
Басқаша айтсақ, a-ның кез келген көршілес жұбы және оған сәйкес b-ның көршілес жұбы үшін, не олардың бір-бірінен арақашықтығы 2^m-нен кіші және тең, немесе екеуі де 2^m-нен кіші.

Лемма Егер

$$a_1,\dots,a_k \equiv_k^m b_1,\dots,b_k$$

болса, онда барлық a_{k+1} үшін

$$a_1,\dots,a_k,a_{k+1} \equiv_{k+1}^{m-1} b_1,\dots,b_k,b_{k+1}$$

орындалатын b_{k+1} табылады.

Дәлелдеу. Алмастыру π арқылы a-лар мен b-лардың реттілігі берілетін болсын. a_{k+1} кез келген болсын және $a_{\pi(i)} \le a_{k+1}$ орындалатын i ең үлкен сан деп ұйғарайық. Сондықтан не $i < k$ және $a_{\pi(i)}$ мен $a_{\pi(i+1)}$-дің арасына a_{k+1} орналасады немесе $i = k$ және $a_0, \dots, a_{k+1}$-дің максимумы a_{k+1} болады. Анықтаймыз:

$$b_{k+1} = \begin{cases} b_{\pi(\mathrm{i}+1)} - a_{\pi(\mathrm{i}+1)} + a_{k+1}, \\ \\ b_{\pi(\mathrm{i}+1)} + a_{k+1} - a_{\pi(\mathrm{i})}, \end{cases}$$

Жаңа алмастыруды $\rho : \{0,1,\dots,k+1\} \to \{0,1,\dots,k+1\}$,

$$\rho(j) = \begin{cases} \pi(j), & \mathrm{j} < i, \\ k+1, & \mathrm{j} = i, \\ \pi(j-1), & \mathrm{j} > i. \end{cases}$$

арқылы анықтаймыз. Сонда

$$a_{\rho(0)} \le a_{\rho(1)} \le \cdots \le a_{\rho(k+1)}$$

$$b_{\rho(0)} \le b_{\rho(1)} \le \cdots \le b_{\rho(k+1)}$$

және барлық $0 \le i \le k$ үшін,

$$\min\{2^{m-1},\ a_{\rho(i+1)} - a_{\rho(i)}\} = \min\{2^{m-1},\ b_{\rho(i+1)} - b_{\rho(i)}\},$$

сондықтан

$$a_1, \dots, a_k, a_{k+1} \equiv_{k+1}^{m-1} b_1, \dots, b_k, b_{k+1}.$$

Енді, егер тек егер барлық $0 \le i \le j \le k$ үшін $b_i \le b_j$ болса, онда $a_i \le a_j$ болады деген тұжырымды $a_1, \dots, a_k \equiv_k^0 b_1, \dots, b_k$ қатынасы меңзейді. Бұл арқылы, барлық атомдық формулада $a_1, \dots, a_k$ және $b_1, \dots, b_k$ сәйкестенетінін білеміз, сондықтан ол тұжырым барлық бейкванторлы формулада орындалады. Осыны тұғыр ретінде пайдалансақ, лемманы пайдаланатын индуктивті аргумент мынаны көрсетеді: егер $a_1,\dots,a_k \equiv_k^m b_1,\dots,b_k$ болса, онда

$$Q_{k+1}x_{k+1}\dots Q_{k+m}x_{k+m}\ \varphi(a_1,\dots,a_k,x_{k+1},\dots,x_{k+m})$$

болады, егер тек егер

$$Q_{k+1}x_{k+1}\dots Q_{k+m}x_{k+m}\ \varphi(b_1,\dots,b_k,x_{k+1},\dots,x_{k+m})$$

орындалатын болса.

Бұл аргумент 21-лекцияда берілген тығыз сызықты реттілік теориясына ұқсайды.

k-кортеждің $a_1,\dots,a_k$ эквиваленттік класын $\equiv_k^m$ алмастыру арқылы көрсетуге болады, ол a-лардың ретін береді және a-лардың екі көршілес жұбының ара қашықтығын да максимум 2^m-ге дейін береді. Бұл ақпарат полиномиалды кеңістікте өрнектелуі мүмкін. Егер ондай өрнектелу берілсе, онда жаңа a_{k+1} -ді қосу арқылы алынған $\equiv_{k+1}^{m-1}$-эквиваленттілік класының барлық мүмкін өрнектелуі полиномиалды уақытта тармақтала есептеумен туындалуы мүмкін. Бұл тығыз сызықты реттілік теориясы үшін 21-лекцияда келтірілгендей ұқсас кванторды жою үшін альтернатив алгоритм береді.

2. (a) 1-раунд: Соня $0 \in \mathcal{B}$ деп ойын бастайды; Дэвид белгілі бір $p \in \mathcal{A}$ деп ойнайды.

2-раунд: Соня $1 \in \mathcal{B}$ деп ойнайды; Дэвид белгілі бір $q < p$, $q \in \mathcal{A}$ деп ойнайды (егер Дэвид белгілі бір $q \le p$ деп ойнаса, онда бірден ұтылады).

3-раунд: Соня $(p+q)/2 \in \mathcal{A}$ деп ойнайды. Дэвид $\mathcal{B}$-да 0 және 1 арасындағы диапазонда ойнай алмайды, сондықтан Соня ұтып шығады. Назар аударыңыз: бір реттілік тығыз ал екіншісі тығыз емес болғандықтан, Соня ұтып шықты.

(b) Екі структура да соңғы нүктелері жоқ тығыз сызықты реттілік, сондықтан Соня қай жерде қалай ойнаса да, тәуелсіз түрде Дэвид үшін екінші структурадан құмалақ ретін сақтап отыратын жүрісті әрқашан табатын мүмкіндік бар.

(c) Жеңілдік үшін терiске шығару амалы тек атомдық формулаларда ғана қолданатындай етіп, формулаларды түрлендіреміз. Осы формадағы бірінші ретті әрбір формуланы эквивалентті формулаға түрлендіру үшін төмендегі ережелерді пайдаланса болады:

$$\neg(\varphi \vee \psi) \quad \Rightarrow \quad (\neg\varphi) \wedge (\neg\psi)$$

$$\neg(\varphi \wedge \psi) \quad \Rightarrow \quad (\neg\varphi) \vee (\neg\psi)$$

$$\neg(\exists x\ \varphi) \quad \Rightarrow \quad \forall x\ \neg\varphi$$

$$\neg(\forall x\ \varphi) \quad \Rightarrow \quad \exists x\ \neg\varphi$$

$$\neg\neg\varphi \quad \Rightarrow \quad \varphi.$$

Осы түрлендіруді $\neg\varphi$-ге қолданғанда алынған нәтиже φ' болсын. Бір бағытта

$$\mathcal{A} \vDash \varphi \quad \text{және} \quad \mathcal{B} \vDash \varphi',$$

болады деп ұйғарайық, мұндағы φ кванторлық тереңдігі n-нен аспайтын сөйлем. Біздің Соняға ұтатын стратегия бергіміз келеді. Соня былай ойнаса ұта алатынын көрсетеміз: k раундтан кейін кванторлық тереңдігі (n–k)-дан аспайтын $\psi(\overline{x})$ формула тауып, және еркін айнымалыларда $\overline{x} = x_1, \dots, x_k$-да

$$\mathcal{A} \vDash \psi(\overline{a}) \text{ және } \mathcal{B} \vDash \psi(\overline{b}),$$

болатын етіп инварианттарды ұстап отырады, мұндағы $\overline{a} = a_1, \dots, a_k$ және $\overline{b} = b_1, \dots, b_k$ осыған дейінгі сәйкес $\mathcal{A}$ және $\mathcal{B}$-да ойналған құмалақтар. Ұйғарым бойынша бұл $k = 0$-де дұрыс. Енді k үшін де дұрыс деп ұйғарайық.

(i) Егер

$$\psi(\overline{x}) \ = \ \psi_1(\overline{x}) \vee \psi_2(\overline{x}),$$

болса, онда

$$\psi'(\overline{x}) \ = \ \psi'_1(\overline{x}) \vee \psi'_2(\overline{x}),$$

сондықтан не

$$\mathcal{A} \vDash \psi_1(\overline{a}) \quad \text{және} \quad \mathcal{B} \vDash \psi'_1(\overline{b})$$

немесе

$$\mathcal{A} \vDash \psi_2(\overline{a}) \text{ и } \mathcal{B} \vDash \psi'_2(\overline{b}).$$

Жалпылықты ұстай отырып бірінші дұрыс деп айтайық. ψ -дің орнына кіші формула ψ_1 -ге аргументті жалғастырамыз.

(ii) Егер

$$\psi(\overline{x}) = \psi_1(\overline{x}) \wedge \psi_2(\overline{x})$$

болса, онда аргумент (i) -жағдайға ұқсас.

(iii) Егер $x_i \leq x_j$ -дің атомдық формуласы $\psi(\overline{x})$ болса, онда

$$a_i \leq a_j \text{ және } b_i \not\leq b_j,$$

бұл Соня үшін ұтыс болып табылады.

(iv) Егер

$$\psi(\overline{x}) = \neg p(\overline{x}),$$

болса, онда $x_i \leq x_j$ -дің атомдық формуласы $\rho(\overline{x})$ болса, онда

$$a_i \not\leq a_j \text{ және } b_i \leq b_j,$$

бұл да Соня үшін ұтыс болып табылады.

(v) Егер

$$\psi(\overline{x}) = \exists x_{k+1}\ \rho(\overline{x}, x_{k+1}),$$

болса, онда

$$\psi'(\overline{x}) = \forall x_{k+1}\ \rho'(\overline{x}, x_{k+1})$$

және

$$\mathcal{A} \vDash \exists x_{k+1}\ \rho(\overline{a}, x_{k+1}).$$

Экзистенциалды квантор үшін $a_{k+1} \in \mathcal{A}$ куәде

$$\mathcal{A} \vDash \rho(\overline{a}, a_{k+1})$$

болатындай етіп, Соня құмалақ ойнайды.

$$\mathcal{B} \vDash \forall x_{k+1}\ \rho'(\overline{b}, x_{k+1}),$$

болғандықтан, Дэвид қайда ойнаса да маңызды емес,

$$\mathcal{B} \ \vDash \ \rho'(\overline{b}, b_{k+1})$$

болатынын аламыз. Және кванторлық тереңдік бірге кем, сондықтан инвариант сақталады.

(vi) Егер

$$\psi(\overline{x}) \ = \ \forall x_{k+1} \ \rho(\overline{x}, x_{k+1})$$

болса, онда аргумент (v)-ке ұқсас, оған қоса Соня $\mathcal{A}$ -ның орнына $\mathcal{B}$ -да ойнайды.

Керісінше, $\mathcal{A}$ мен $\mathcal{B}$ барлық тереңдігі n кванторлы сөйлемдермен сәйкестенген дейік. Егер $\mathcal{A}, a_1, \dots, a_n$ және $\mathcal{B}, b_1, \dots, b_n$ барлық атомдық формулалармен сәйкестендірілген, яғни барлық кванторлық бейкванторлы формулалармен сәйкестендірілген болса, онда

$$\mathcal{A}, a_1, \dots, a_n \ \equiv_n^0 \ \mathcal{B}, b_1, \dots, b_n$$

деп анықтаймыз; яғни барлық бейкванторлы $\varphi(x_1, \dots, x_n)$ формула үшін,

$$\mathcal{A} \vDash \varphi(a_1, \dots, a_n) \quad \Leftrightarrow \quad \mathcal{B} \vDash \varphi(b_1, \dots, b_n).$$

Егер барлық $a_{k+1} \in \mathcal{A}$ үшін $b_{k+1} \in \mathcal{B}$ табылып,

$$\mathcal{A}, a_1, \dots, a_k, a_{k+1} \quad \equiv_{k+1}^{m-1} \quad \mathcal{B}, b_1, \dots, b_k, b_{k+1}$$

орындалса және кез келген $b_{k+1} \in \mathcal{B}$ үшін $a_{k+1} \in \mathcal{A}$ табылып

$$\mathcal{A}, a_1, \dots, a_k, a_{k+1} \quad \equiv_{k+1}^{m-1} \quad \mathcal{B}, b_1, \dots, b_k, b_{k+1}$$

орындалса, онда $m > 0$ үшін

$$\mathcal{A}, a_1, \dots, a_k \ \equiv_k^m \ \mathcal{B}, b_1, \dots, b_k$$

деп анықтаймыз.

Индуктивті аргумент көмегімен көрсетуге болады, егер

$$\mathcal{A} \ \equiv_0^n \ \mathcal{B}$$

болса, онда n -құмалақты ойында Дэвидтің ұтатын қарапайым стратегиясы болады: k раундтан кейін құмалақтарды инвариант сақталатындай етіп

$$\mathcal{A},\ a_1,\dots,a_k \quad \equiv_k^{n-k} \quad \mathcal{B},\ b_1,\dots,b_k$$

орналастыру қажет.

Енді біз $\equiv_k^m$ -эквиваленттілігінің әрбір класын k еркін айнымалылы формула және m кванторлық тереңдікпен өрнектеуге болатынын көрсетеміз. Басқаша айтсақ, $\equiv_k^m$ -ның әрбір эквивалентті класы E үшін, еркін $x_1,\dots,x_k$ айнымалылы және m квантор тереңдікті $\varphi_E(x_1,\dots,x_k)$ формуласы табылып, егер тек егер

$$\mathcal{A} \vDash \varphi_E(a_1,\dots,a_k)$$

болса, онда $\mathcal{A},a_1,\dots,a_k \in E$ болады.

Сонымен, егер кванторларының тереңдігі n болатын сөйлемдердің бәрінде $\mathcal{A}$ және $\mathcal{B}$ сәйкестелетін болса, онда олар эквиваленттілік класы $\equiv_0^n$-ны анықтайтын барлық сөйлемдерде де сәйкестенеді, сондықтан олар $\equiv_0^n$-эквивалентті болады. Нәтижесінде, Дэвидте ұтысқа апаратын стратегия болады.

φ_E формуласы индуктивті анықталады. Сондықтан $\equiv_n^0$-дегі әрбір эквиваленттілік класы $x_1,\dots,x_n$ айнымалылардан құрастырылған атомдық формулалар конъюкциясы немесе атомдық формулаларды теріске шығару арқылы анықталады. $m>0$ үшін, $\mathcal{A}$, $\overline{a}$ -ның эквиваленттілік класы $\equiv_k^m$ болады, мұндағы $\overline{a}=a_1,\dots,a_k$ төмендегі ережемен анықталады. Барлық мүмкін болатын таңдау $a_{k+1}\in\mathcal{A}$ үшін $\mathcal{A}$, $\overline{a}$, a_{k+1} -ның барлық $\equiv_{k+1}^{m-1}$ -эквиваленттілік класының жиыны $\mathcal{E}$ болсын. Осындай шартқа бағынатын a_{k+1} шексіз көп табылатын болғанымен, $\equiv_{k+1}^{m-1}$ -эквиваленттілік класы тек шектеулі болады (бұл факты да осы конструкциядан индуктивті алынады). Индукция гипотезасына сүйенсек, әрбір $E\in\mathcal{E}$ үшін, E-ні анықтайтын кванторлық тереңдігі $m-1$ болатын формула $\varphi_E(\overline{x},x_{k+1})$ бізде бар. Бұл формула $\mathcal{A}$, $\overline{a}$ -ның $\equiv_k^m$ -эквиваленттілік класын анықтайды, сондықтан

$$\bigwedge_{E\in\mathcal{E}} \exists x_{k+1}\ \varphi_E(\overline{x},x_{k+1}) \quad \wedge \quad \forall x_{k+1} \bigvee_{E\in\mathcal{E}} \varphi_E(\overline{x},x_{k+1})$$

m болады. Бұл кванторды жойып жіберу амалы арқылы алынған формула $\equiv_{k+1}^{m-1}$ -эквивалентті кластар жиынын сипаттайды.

8-үй жұмысының шешімі

1. Келесі бейдетерминирлі Бучи автоматында бәріне қабылдау ашық және тек (қатысты сипаттама функция) ω-ның шектеулі ішкі жиыны қатысады. Автомат ұйғарымы: енгізу жолының соңғы 1-ін көргеннен кейін ол соңғы күйге енеді, осыдан кейін ол тек 0-дерді көріп отыруы қажет; кері жағдайда ол өлі күйге енеді. Осы жыйынға қабылдау-ашық болатын детерминирлі Бучи автоматы болмайды. Осыны кері жору арқылы дәлелдейміз. Детерминирлі осындай n күйлі автомат M бар деп ұйғарайық. Автомат M -нің шексіз жолдағы $(10^{n+1})^{\omega}$ жұмысын қарастырайық. Қатарынан келген $(n+1)$ 0-дер тізбегінің k-ншы ішкі жолын сканерлеген кезде, M күйді қайталауы қажет және қайталау күйінің екі ену арасындағы цикл күйлердің біреуі міндетті түрде қабылдау-ашық күй болуы қажет, өйткені M-де жол $(10^{n+1})^k 0^{\omega}$-ға қабылдау-ашық. Сондықтан $(10^{n+1})^{\omega}$ енгізуінде M белгілі бір қабылдау-ашық күйде шексіз жиі болуы қажет, сондықтан ол қателікпен қабылдайды.

2. (a) Егер қосу амалы S1S -те анықталған болса, онда бейдетерминирлі Бучи автоматы табылар еді. Ол автомат $a+b=c$ орындалатын a, b, c сандарын $\{0,1\}^3$ алфавиттегі жолдармен қабылдайтын еді. Мысалы, 7+4=11 келесі жолмен өрнектеледі

$$
\begin{matrix}
0000000 1000000 \\
0000100000000 \cdots \\
0000000000100
\end{matrix}
$$

Біз енді керекті қайшылықты алу үшін помпа аргументін пайдаланамыз. M-нің n күйі бар деп ұйғарайық. Қосу проблемасына $(n+1)+(n+1)=2n+2$ сәйкестендірілген, M-де қабылдау-ашық енгізу жолын қарастырамыз. Енгізу жолының $n+1$ және $2n+2$ позицияларының арасындағы ішкі жолды $(0,0,0)^n$ сканерлегенде машина q күйді қайталауы қажет. Және q-дің екі енуінің арасындағы ішкі жол жойылуы қажет және M-дегі қабылдау-ашық қате.

(b) 25-лекцияда $y \le x$ және A шектеулі болатынын қалай айту керек екендігін көрсеткен едік. *A* және *B*-мен өрнектелген биттік векторларды қосу үшін бинарлық қосуды симуляция жасаймыз. Сол жақ ең шеткісі төменгі ретті бит болып табылады. Ауысу басқа *U* шектеулі жиынымен беріледі. Мысалы,

$$
\begin{aligned}
U &= 0000000011101000010100000000000\cdots \\
A &= 0010100111010100101000000000000\cdots \\
B &= 0101000101010001101000000000000\cdots \\
C &= 0111100001101101010100000000000\cdots
\end{aligned}
$$

U -ды ауысу жолы деп жариялау үшін біз былай деп жариялаймыз: U -дың төменгі ретті биті 0 дейміз, және кез келген *i* үшін егер A, B және U -дың кем дегенде екіден кем емес *i*-ші биттері 1-ге тең болса:

$$
\begin{aligned}
\kappa(A,B,U) \;=\; 0 \not\in U \wedge \forall x\ \mathbf{s}x \in U \leftrightarrow ((x \in A \wedge x \in B) \vee \\
(x \in A \wedge x \in U) \vee \\
(x \in B \wedge x \in U))
\end{aligned}
$$

онда U -дың *i+1*-ші биті 1-ге тең болады. C қосындысы A мен B -ның биттік жолының шығаратын-немесе $(\text{mod } 2 \text{ sum})$ арқылы беріледі, оған қосымша ауыстыру:

$$
\begin{aligned}
\varphi(A,B,C) \;=\; \exists U\ \kappa(A,B,U) \;\wedge \\
\forall x\ (x \in C \leftrightarrow x \in A \leftrightarrow x \in B \leftrightarrow x \in U)
\end{aligned}
$$

арқылы да беріледі.

3. (a) $B_1 = 0101010101\ldots$ болғандықтан,

$$
\begin{aligned}
\varphi_1(x,y) &= 1 \\
\psi_1(B) &= 0 \not\in B \wedge \forall x\ x \in B \leftrightarrow \mathbf{s}x \not\in B \\
\varphi_2(x,y) &= \exists B\ \psi_1(B) \wedge (x \in B \leftrightarrow y \in B)
\end{aligned}
$$

деп алуымызға болады. Енді біз $\varphi_n(x,y)$ және $\psi_n(B)$ -ларды құрастырып алдық деп ұйғарамыз. Проблеманың сипаттамасында жазылған анықтамада ұсынылғандай, B_n -ді *n* битті ішкі жолдарға бөлінген шексіз бинарлы жол ретінде қарастырамыз. Осы *n*-битті ішкі жолдарды *n-блоктар* деп атаймыз. Әрбір n-блоктың бірінші битінің орны n-ге еселі болады. Алдымен бірнеше қосымша формулаларды құрастырамыз:

$$\rho_n(x,y) = \varphi_n(y,0) \wedge y \le x \wedge \forall\omega\ (\varphi_n(\omega,0) \wedge \omega \le x) \to \omega \le y$$

= "y ең үлкен n-нің еселігі, ол x-тен кіші немесе тең"

$$= \varphi_n(x,y) \wedge \exists z\ y = \mathbf{s}z \wedge \rho_n(z,x)$$

= "x және y n-нің тізбекті еселіктері "

$$A = \{0\} = \forall x\ x \in A \leftrightarrow x = 0$$

$$A = \{0\} = \forall x\ x \in A \leftrightarrow \xi_n(0,x)$$

$$\div_n(A) = \forall z\ z \in A \to \rho_n(z,0)$$

$$= A \subseteq \{0,1,\dots,n-1\}$$

$$\omega_n(A,B) = \chi_n(A) \wedge \chi_n(B) \wedge (\varphi(A,\{0\},B) \vee \varphi(A,\{0\},B \cup \{n\}))$$

= " A,B бинарлық санды $0 \le n(A)$ өрнектейді, $n(B) = n(A) + 1 \pmod{2^n}$ болатындай $n(B) \le 2^n - 1$ "

(мұндағы $\varphi(A,B,C)$ 2(b)-жаттығуда анықталған формула)

$$\sigma_n(A,z,B,\omega) = \forall x\ \forall y\ (\rho(x,y) \wedge \rho(y,\omega) \wedge \varphi_n(x,y)) \to (x \in A \leftrightarrow y \in B)$$

= " A мен B -нын y және z -тен басталатын n-блоктары бірдей "

$$\upsilon_n(A,B,y) = \sigma_n(A,0,B,y) \wedge \chi_n(A)$$

= " $A \subseteq \{0,1,\dots,n-1\}$ және A -ның 0-ден басталатын, ал B -ның z -тен басталатын n-блоктары бірдей "

$$
\begin{aligned}
\tau_n(x,y,B) &= \forall z\ \forall \omega\ (\rho_n(x,z) \wedge \rho_n(y,\omega)) \to \sigma_n(B,z,B,\omega) \\
&= \text{“}B\text{-да } x \text{ пен } y\text{-ті қамтыйтын } n\text{-блоктар бірдей”}
\end{aligned}
$$

Келтірілген формулалар $\varphi_n(B)$-нен тәуелді, ал $\varphi_n(B)$-нен тәуелсіз, осыған назар аударыңыз. Енді біз анықтай аламыз

$$
\begin{aligned}
\varphi_{n2^n}(x,y) &= \varphi_n(x,y) \wedge \exists B\ \psi_n(B) \wedge \tau_n(x,y,B) \\
&= \text{“}B_n \text{-да } x \text{ пен } y\text{-тің } n\text{-блоктары бірдей,} \\
&\quad \text{және } x \text{ пен } y \text{ блокта бірдей позицияда} \\
&\quad \text{орналасқан; яғни, } x \equiv y(\operatorname{mod} n)\text{”} \\
&= x \equiv y(\operatorname{mod} n2^n).
\end{aligned}
$$

Қолға $\varphi_{n2^n}(x,y)$ түсе сала, біз қосымша формулаларды $\rho_{n2^n}(x,y)$, $\xi_{n2^n}(x,y)$ құрастыра аламыз және т.с.с. жоғарыда көрсетілгендей. Сонда

$$
\begin{aligned}
\psi_{n2^n}(B) &= \forall y\ \forall z\ \forall C\ \forall D\ (\xi_{n2^{2n}}(y,z) \wedge \upsilon_{n2^n}(C,B,y) \\
&\qquad \wedge \upsilon_{n2^{2n}}(D,B,z) \to \omega_{n2^{2n}}(C,D)) \\
&\quad \wedge \forall y\ \rho_{n2^{2n}}(y,0) \to y \notin B \\
&= \text{“}n2^n\text{-нің тізбектелген барлық еселі жұбы} \\
&\quad \text{үшін, осы екеуінен басталатын } n2^n\text{-блоктардың} \\
&\quad \text{орны, келесі диапазонда сандарды өрнектейді} \\
&\quad \{0,\ldots,2^{n2^n}-1\}\text{, олар } 1 \bmod 2^{n2^n} \text{ арқылы} \\
&\quad \text{ажыратылады, және бірінші блок тек} \\
&\quad \text{0-ден тұрады”} \\
&= B = B_{n2^n}.
\end{aligned}
$$

(b) Бөлік (a)-да берілген конструкцияны пайдаланып, ұзындығы $2^{2^{2^{\cdot^{\cdot^{n}}}}}$ -нен кем емес жолды сипаттайтын, ұзындығы n болатын формуланы құрастыруға болады. Нақты қосу теориясы үшін 23-лекцияда келтірілген төменгі шекараның дәлелдемесі сияқты, ол формулаларды Тьюринг машинасы есептеуінің қабылдау-ашық тарихының жыйынын сыйпаттауда “критерий” ретінде ол формулаларды қолдаиға болады және оның есептеу уақыты $2^{2^{2^{\cdot^{\cdot^{n}}}}}$ -ға тең.

9-үй жұмысының шешімі

1. Проблема $PSPACE$-толық болады. Бұл жерде алтернатив полиномиалды алгоритм. $\mathbb{Z}_n \models \exists x\ \varphi(x)$ болатынын тексеру үшін $a \in \mathrm{Z}_n$ деп ұйғарамыз да, экзистенциалды тармақталамыз. Осындай әрбір a бинарлы жүйеде $\{0,1,\ldots\ n-1\}$ сандарымен өрнектеле алады, өйткені n-нің бинарлы өрнектелуінде есептеу бұтағының тереңдігі сызықты. Осыдан кейін жапырақтағы әрбір процесс табылған a-ны пайдаланып, $\mathbb{Z}_n \models \varphi(a)$-ны тексереді. $\mathbb{Z}_n \models \forall x\ \varphi(x)$-ны тексеру процедурасы бұрынғыдай, тек бұл жағдайда универсал тармақталу пайдаланылады, осы айырмашылық болып табылады. Логикалық $\vee$ және $\wedge$ байланыстар сәйкес логикалық экзистенциалды және универсал тармақталулар арқылы баяндалуы мүмкін. Де Морган заңы және $\neg\exists x\ \varphi(x) \mapsto \forall x\ \neg\varphi(x)$ және $\neg\forall x\ \varphi(x) \mapsto \exists x\ \neg\varphi(x)$ ережелері арқылы теріске шығару $\neg$ амалы атомдық формулаға дейін сығылған деген ұйғарым жасаймыз. Атомдық формалар $s = t$ немесе $s \neq t$ қалды, мұндағы s және t $\mathbb{Z}_n$-тегі тұрақтылар арқылы анықталған базалық терминдер және $\cdot$ мен $+$ арифметикалық операторлары. Модуль n-де осыны кәдімгі арифметика көмегімен полиномиалды уақытта тексеріп шығуға болады.

QBF-тегі келтірімділік көмегімен $PSPACE$ үшін проблема күрделі болатынын көрсететін боламыз. Егер QBF-тың кванторлы логикалық формуласы

$$\mathrm{Q}_1 x_1 \mathrm{Q}_2 x_2\ \cdots\ \mathrm{Q}_n x_n\ \mathcal{B}(x_1,\ldots\ ,x_n)$$

берілсе, онда әрбір x_i логикалық айнымалыны атомдық формула $x_i = 0$-мен алмастырамыз. Бұл арқылы сандар теориясының тілінде сөйлем аланады, егер тек егер бастапқы кванторлы логикалық формула екі элементті $\{0,1\}$ логикалық алгебрада ақиқат болса, онда ол сөйлем $\mathbb{Z}_2$-де ақиқат немесе $n \geq 2$ болғанда $\mathbb{Z}_n$-де ақиқат болады.

Барлық бейтривиалды бірінші ретті теория T үшін осыған ұқсас мәлімет жұмыс жасайды. Бізге керектісі, $T \models \exists \overline{x}\ R(\overline{x})$ және $T \models \exists \overline{x}\ \neg R(\overline{x})$ орындалатын R қатынасының бар болуы (бейтривиалды деп біздің айтып жүргеніміз осы). Қолда бар қолданбаны пайдаланып, $R(x)$-ті $x = 0$ деп аламыз, структураның ең аз дегенде екі элементі бар деген шарт бойынша алынған $R(x)$ бейтривиалды.

2. Алдымен

$$M = (Q, \{0,1\}, \delta, s, \mathcal{F})$$

берілген Мюллер автоматы болсын. Күй q-ге сәйкестендірілген жиынды айнымалы Y_q болсын және енгізуге сәйкестенген қандай да бір жиынды айнымалы X болсын. Алдымен S1S формуланы $\mathsf{run}(X, \overline{Y})$ деп жазамыз, осы арқылы енгізу X-тегі M-нің жүгіртпесі сипатталады. Машина күй q-да болатын уақытты Y_q айнымалы береді.

$$\mathsf{run}(X, \overline{Y}) = 0 \in Y_s \tag{8}$$

$$\wedge \forall n \bigwedge_q (n \in Y_q \wedge n \notin X \to \bigvee_{p \in \delta(q,0)} s(n) \in Y_p \tag{9}$$

$$\wedge \forall n \bigwedge_q (n \in Y_q \wedge n \in X \to \bigvee_{p \in \delta(q,1)} s(n) \in Y_p \tag{10}$$

$$\wedge \forall n \bigwedge_{p \neq q} \neg (n \in Y_p \wedge n \in Y_q) \tag{11}$$

Ішкі формула (8), машина 0-ші уақытта алғашқы күйден бастайды деп айтады. Ішкі формула (9), енгізу 0-дегі ауысулар корректі деп айтады. Аналогты, ішкі формула (10), енгізу 1-дегі ауысулар корректі болады деп айтады. Және, сөз соңында ішкі формула (11) кез келген уақытта машина тек бірден аспайтын күйде ғана бола алады дейді. Егер енгізу A-да M-нің жүгіртпесін B_q сипаттайтын болса, онда кез келген жиындар A және B_q, $q \in Q$-да $\mathsf{run}(X, \overline{Y})$ S1S формула ақиқат болады. Айтылғанды мына мағынада түсінеміз: кез келген n үшін, егер n уақытта машина q күйде бола алса, сонда тек сонда $n \in B_q$ орындалады.

Енді қабылдау-ашық жағдайды баяндаймыз. Анықтаймыз:

$$\mathsf{finite}(Y) \;=\; \exists x\ \forall y\ y \in Y \to y \le x\,.$$

Сонда $\neg\,\mathsf{finite}(Y_q)$ арқылы M машинасы q күйде шексіз жиі болатыны айтылады. $F \subseteq Q$ үшін, анықтаймыз

$$\mathsf{io}_F(\overline{Y}) \;=\; \bigwedge_{q\in F} \neg\,\mathsf{finite}(Y_q) \wedge \bigwedge_{q\notin F} \neg\,\mathsf{finite}(Y_q).$$

Бұл арқылы $\overline{Y}$ -ның көмегімен сипатталған, F -тің IO жүгіртпелер жиыны екендігі айтылады. Соңына қарай, анықтаймыз

$$\mathsf{accept}(\overline{Y}) \;=\; \bigvee_{F\in\mathfrak{F}} \mathsf{io}_F(\overline{Y}).$$

Осы арқылы Мюллер қабылдау-ашық шарты арқылы M -де қабылдау-ашық болатынын білеміз. Енді

$$\varphi_M(X) \;=\; \exists \overline{Y}\ \mathsf{run}(X,\overline{Y}) \wedge \mathsf{accept}(\overline{Y})$$

деп аламыз.

3. Алдымен, сиретілген C оракулды полиномиалды уақытты оракул машина M^C көмегімен, полиномиалды өлшемді схемаларды $B_0, B_1, \ldots$ қалай симуляция жасауға болатынын көрсетеміз. C оракулде B_n схемасын кодтаймыз.

Схемаларда қабылдау-ашық жиын $A \in \{0,1\}^*$ болады деп ұйғарайық. Сондықтан $x \in \{0,1\}^n$ үшін егер тек егер $x \in A$ болса, онда $B_n(x) = 1$ болады. Схемалар полиномиалды өлшемді болғандықтан, d тұрақты және схемалар коды B_n жол $b_n \in \{0,1\}^{n^d}$ түрінде бар болып, b_n және $x \in \{0,1\}^n$ берілген жағдайда Тьюринг машинасы $B_n(x)$ -ті полиномиалды уақытта есептей алады.

Енді біз b_n жолды оракулде ұзындығы n болатын жол ретіндегі сипаттама функция ретінде сақтаймыз. Яғни, $n^d \le 2^n$ орындалатындай өте үлкен n үшін, егер тек егер b_n -нің i-ші биті 1-ге тең болса, сонда ғана ұзындығы n болатын i-ші жолды C-ға орналастырамыз. Сондықтан ұзындығы n болатын n^d -дан аспайтын жолдар үшін C-дан сұратым арқылы B_n -ны анықтауға болады. $n^d > 2^n$ орындалатын шектеулі көп n үшін M соңғы бақылауында B_n схема кодтала салады.

$|b_n| \le n^d$ болғандықтан, оракул сиретілген болады. Алдымен машина M^C, енгізу $x \in \{0,1\}^n$-де ұзындығы n болатын бастапқы n^d жол үшін B_n-ді анықтау мақсатында C-ға сұратым түсіреді, осыдан кейін $B_n(x)$-ді есептейді, егер мән 1-ге тең болса, онда қабылдау-ашық.

Басқа бағыт үшін сиретілген C, $|C \cap \{0,1\}^n| \le n^d$ оракулді A жиынына қабылдау ашық, полиномиалды уақытты оракул машина M^C берілген деп ұйғарайық. Біз M^C-ға эквивалентті полиномиалды өлшемді $B_0, B_1, \dots$ схемаларды (бейтекті) құрастырғымыз келеді. Қалай болса да оракул ақпаратты схемаға кодтауымыз қажет. Оны екі этаппен іске асырамыз. Алдымен әрбір n үшін ұзындығы n енгізулерде M-ге керекті барлық оракул ақпаратты кодтайтын полиномиалды ұзындығы n болатын y_n жолы табылатынын дәлелдейміз. Бұл y_n жол C-дағы барлық элементтердің, ең ұзын элементті қоса тізімін көрсетеді, ол ұзындығы n-ге тең енгізулерде M машинамен сұратылуы мүмкін. Бұл жолдар n-де полиномиалды ұзындықты, ал C сиретілген болғандықтан, y_n жол полиномиалды ұзындықты болады. Дәлірек айтайық, M-нің уақыт бойынша шектеуі n^k болсын. Ұзындығы n болатын енгізулерде, M машина ұзындығы n^k-дан аспайтын жолдарда ғана оракулге сұратым жасай алады, өйткені оларды жазып алуы да қажет. Ал C n^d-сиретілген болғандықтан,

$$|C \cap \{0,1\}^{\le n^k}| = \sum_{m=0}^{n^k} |C \cap \{0,1\}^m| \le \sum_{m=0}^{n^k} m^d \le n^{k(d+1)}$$

аспайтын нөлден өзгеше C-ның диапозондағы элементі табылады. Олар ұзындығы n енгізулерде M-мен сұратылуы мүмкін, және олардың барлығының ұзындығы n^k-дан аспайды. Және элементтер бір шеттен екінші шетке 2-мен ажыратылып, ұзындығы $O(n^{k(d+2)})$-ден аспайтын $z_n \in \{0,1,2\}^*$ жолда жазылуы мүмкін. Енді бинарлық жүйедегі y_n-ді алу үшін үштік санау жүйесіндегі z_n жолды түрлендіреміз. Сонда

$$|y_n| \le \log_2 3 \cdot |z_n| = O(n^{k(d+2)}).$$

Ұзындығы n болатын енгізу жолдарды ұқсату үшін бинарлық жол y_n барлық оракул ақпаратты қамтуы қажет. Яғни, детерминирлі полиномиалды уақытты TM болатын N машина табылып, ұзындығы n болатын кез келген x жол үшін, егер тек егер x-ке M^C-де қабылдау-ашық болса, онда N-де $x\#y_n$-ге қабылдау-ашық болады. Машина N алдымен y_n-ді z_n-ге түрлендіреді, содан кейін x-те M-ді симуляция жасайды; M өзінің C оракуліне консултация жасаған барлық жағдайда, z_n тізімнен N іздеу жүргізеді.

N машинасы үшін Ландер конструкциясымен алынған схемалар $C_0, C_1, \ldots$ болсын (6.1-теорема). Сондықтан кез келген n үшін, $C_{n+|y_n|}$-нің $n+|y_n|$ лигикалық енгізуі және $n+|y_n|$-де полиномиалды өлшемі болады. Ол n-де полиномиалды және n ұзындықты кез келген x-те, егер $C_{n+|y_n|}(x, y_n) = 1$ болса, сонда тек сонда N машинада $x\#y_n$-ге қабылдау-ашық болса және сонда тек сонда M^C машинада x-ке қабылдау-ашық болса. y_n-нің логикалық мәнін пайдаланып, енгізу $C_{n+|y_n|}$-ді y_n-ге арнайы сәйкестендіру арқылы B_n алынады.

10-үй жұмысының шешімі

1. Жалпы рекурсивті const-ты құрастыру үшін, π_1^2 проекциясының индексі k және s_1^1-тің индексі ℓ болсын. Сонда $\mathrm{const} = \varphi_{s_1^1(\ell,k)}$ деп алу үшін:

$$\begin{aligned} i &= \pi_1^2(i,x) \\ &= \varphi_k(i,x) \\ &= \varphi_{s_1^1(k,i)}(x) \\ &= \varphi_{\varphi_\ell(k,i)}(x) \\ &= \varphi_{\varphi_{s_1^1(\ell,k)}(i)}(x). \end{aligned}$$

Жалпы рекурсивті pair-ты құрастыру үшін s_2^1-дің индексі n ал дербес рекурсивті функция

$$< U \circ < \pi_1^3, \pi_3^2 >, U \circ < \pi_2^3, \pi_3^3 >>$$

индексі m болсын. Сонда, $\mathrm{pair} = \varphi_{s_1^2(n,m)}$-ды алу үшін:

$$\begin{aligned} < \varphi_i, \varphi_j > (x) &= <\varphi_i(x), \varphi_j(x)> \\ &= <U(i,x), U(j,x)> \\ &= <U \circ < \pi_1^3, \pi_3^2 >, U \circ < \pi_2^3, \pi_3^3 >> (i,j,x) \\ &= \varphi_m(i,j,x) \\ &= \varphi_{s_2^1(m,i,j)}(x) \\ &= \varphi_{\varphi_n(m,i,j)}(x) \\ &= \varphi_{\varphi_{s_1^2(n,m)}(i,j)}(x). \end{aligned}$$

2. Енгізу $<\upsilon, j>$-де функция индексін өндіретін h жалпы рекурсивті функция болсын, ол енгізу x-те:

(i) $\varphi_\upsilon(\upsilon, j)$-ді есептейді;

(ii) егер $\varphi_\upsilon(\upsilon, j)\downarrow$ болса, онда φ_j-ді $\varphi_\upsilon(\upsilon, j)$ -ге қолдады; және

(iii) егер $\varphi_j(\varphi_\upsilon(\upsilon, j))\downarrow$ болса, онда нәтижені индекс сияқты интерпретациялайды және x-ке осы индекспен функцияны қолданады.

Сондықтан, егер $\varphi_\upsilon(\upsilon, j)$ және $\varphi_j(\varphi_\upsilon(\upsilon, j))$ анықталса, онда

$$\varphi_{h(\upsilon,j)}(x) = \varphi_{\varphi_j(\varphi_\upsilon(\upsilon,j))}(x)$$

болады, кері жағдайда олар анықталмайды. h-тың өзі жалпы рекурсиялы функция: ол жоғарыда келтірілген (i)-(iii) қадамдардың ешқайсысын жасамайды, бірақ ол қадамдарды жасайтын функцияның индексін есептейді.

Енді h-тың индексі u болсын делік. Егер φ_j жалпы болса, онда φ_j -дің қозғалыссыз нүктесі $h(u, j)$ болады:

$$\varphi_{h(u,j)} = \varphi_{\varphi_j(\varphi_\upsilon(\upsilon,j))} = \varphi_{\varphi_j(h(u,j))}.$$

Сондықтан біз $\tau = \text{ё} j.h(u, j)$ -ды анықтай аламыз.

Формалдылықты біраз күшейтейік, ол үшін ℓ және m-ді

$$U \circ < U \circ < \pi_2^3, U \circ < \pi_1^3, \pi_1^3, \pi_2^3 >>, \pi_3^3 >$$

және s_2^1 функцияларының сәйкес индекстері болсын деп таниық, сонда

$$\begin{aligned}
\varphi_{\varphi_j(\varphi_\upsilon(\upsilon,j))}(x) &= U(U(j, U(\upsilon, \upsilon, j)), x) \\
&= U \circ < U \circ < \pi_2^3, U \circ < \pi_1^3, \pi_1^3, \pi_2^3 >>, \pi_3^3 > (\upsilon, j, x) \\
&= \varphi_\ell(\upsilon, j, x) \\
&= \varphi_{s_2^1(\ell,\upsilon,j))}(x),
\end{aligned}$$

болады да, біз $h = \text{ё} < \upsilon, j > .s_2^1(\ell, \upsilon, j))$ деп ала аламыз. Функция s_2^1 жалпы болғандықтан, h та жалпы функция. Сонда h-тың индексі

$$u \overset{\text{def}}{=} s_2^1(m,\ell),$$

формуласының көмегімен беріледі және біз

$$\tau = \lambda j.h(u,j) = \lambda j.\varphi_u(u,j) = \varphi_{s_2^1(u,u)}$$

деп ала аламыз.

3. Қозғалыссыз f нүктелердің құрастырылған шектеулі A жиыны бізде бар деп ұйғарайық. A жиынында жатпайтын басқа қозғалыссыз нүктені тиімді қалай алуға болатынын көрсетеміз. f'-ті алу үшін f-ті келесі түрде модификациялаймыз:

$$f'(i) = \begin{cases} \text{const}(0), & \text{егер } \quad i \in A \text{ болса} \\ f(i) & \quad \text{otherwise,} \end{cases}$$

мұндағы $\mathsf{const}(0)$ – ол тұрақты функцияның $\lambda x.0$ индексі. Рекурсия жайлы теореманы пайдаланып, f'-та жататын қозғалмайтын нүкте j-ді табамыз. Егер $j \not\in A$ болса, онда біз керекті нүктені тауып болдық. Кері жағдайда, біз $\varphi_j = ёx.0$ болатынын білеміз. Бұл жағдайда $f'(j) := \mathsf{const}(1)$ деп қайта анықтаймыз. Енді бізде j f'-тің қозғалыссыз нүктесі болуы мүмкін еместігінің кепілдемесі бар. Процесті жаңа f'-пен қайталаймыз. A-дан қозғалыссыз нүкте k алған әрбір жағдайда, $f'(k) := \mathsf{const}(1)$ деп қайта анықтаймыз және қайталаймыз. A-ның ешқандай элементі f'-тің қозғалыссыз нүктесі болмайтын болса, онда процесс $|A|$-дан артық рет қайталануы мүмкін емес. Рекурсия жайлы теореманың келесі қолданысы f' қозғалыссыз нүктесін A-ның сыртынан береді, ол нүкте қозғалмайтын нүкте f-ті де береді, өйткені f және f' екеуі A-ның сыртында сәйкестенеді.

11-үй жұмысының шешімі

1. 37-ші лекцияда $K^{\varnothing} = K$ жиыны Σ_1^0-толық болады деп тұжырымдалып еді. Индукцияны пайдаланып, A жиыны Σ_n^0-толық болғанда, K^A әрқашан Σ_{n+1}^0-толық болатынын дәлелдесе жеткілікті болады. Сонда

$$K^A \;=\; \{x \mid \varphi_x^A(x)\downarrow\} \;=\; \{x \mid \exists t\ \varphi_x^A(x)\downarrow^t\},$$

деп жазамыз, өйткені $A \in \Sigma_n^0$-да предикат $\varphi_x^A(x)\downarrow^t$ рекурсивті, A-да K^A р.с. болады, сондықтан 35-лекцияда келтірілген анықтама бойынша $K^A \in \Sigma_{n+1}^0$ болады.
Қазір A жиыны Σ_n^0-күрделі болған кезде, K^A жыйыны Σ_{n+1}^0-күрделі болатынын дәлелдейміз. Σ_{n+1}^0-дің кез келген элементі B болсын. 35.1-теорема бойынша B-ны былайша өрнектеуге болады:

$$B \;=\; \{x \mid \exists x\ x\#y \in C\},$$

мұндағы $C \in \Pi_n^0$. Ал A жиыны Σ_n^0-күрделі болғандықтан, оның толықтауышы $\sim A$ Π_n^0-күрделі болады, сондықтан жалпы рекурсиялық бейне σ табылып,

$$\sigma(x\#y) \in \sim A \quad \Leftrightarrow \quad x\#y \in C.$$

орындалады. Енді енгізу x-те A оракулді машина индексін беретін, жалпы рекурсиялы бейне τ-ды анықтаймыз. Және ол кез келген енгізуде:

- $y = 0,1,2,\ldots$ ретімен тізбелейді,
- әрқайсысы үшін $\sigma(x\#y)$-ті есептейді,
- өзінің оракуліне $\sigma(x\#y) \notin \sim A$,

орындалатынын тексеретін консултация береді және

- егер ол әйтеуір біреуін тапса болды тоқтайды.

Сонда

$$\begin{aligned} x \in B &\Leftrightarrow \exists y\ x\#y \in C \\ &\Leftrightarrow \exists y\ \sigma(x\#y) \not\in A \\ &\Leftrightarrow \varphi^A_{\tau(x)}(\tau(x))\downarrow \\ &\Leftrightarrow \tau(x) \in K^A, \end{aligned}$$

B-дан K^A-ға дейінгі келтірімділікті τ бейнесі көрсетеді. Ал B кез келген болғандықтан, K^A жиыны Σ^0_{n+1}-күрделі болады.

2. (a) $K^A \in \Sigma^0_n$ болады деп керісінше жорыйық. Ал A жиыны Σ^0_n үшін $\leq_m$- толық болатындықтан, жалпы рекурсиялы бейне σ табылып, барлық x үшін

$$x \in K^A \quad \Leftrightarrow \quad \sigma(x) \in A$$

қатынас орындалады.

Диогоналдаймыз, A оракулды оракул машинаның индексі m болсын, егер тек егер $\sigma(y) \not\in A$ болса, онда ол машина енгізу y-те тоқтайды. Сонда

$$\begin{aligned} \sigma(m) \in A &\Leftrightarrow m \in K^A \\ &\Leftrightarrow \varphi^A_m(m)\downarrow \\ &\Leftrightarrow \sigma(m) \not\in A, \end{aligned}$$

болады. Қайшылыққа тап болдық. Сондықтан $K^A \not\in \Sigma^0_n$.

(b) $n \geq 1$ деп ұйғарайық. $\Pi^0_n \not\subseteq \Sigma^0_n$ болатынын көрсетсек жеткілікті болады.

35.1-теорема бойынша B-ның кез келген Σ^0_{n+1} жиынын

$$B = \{x \mid \exists y\ x\#y \in A\},$$

түрінде өрнектеуге болады, мұндағы $A \in \Pi^0_n$. Егер $\Pi^0_n \subseteq \Sigma^0_n$ болса, онда $A \in \Sigma^0_n$, сондықтан 35.1-теореманың өрнегіндегі алғашқы және экзистенцияналды кванторлардың бірігуін пайдаланып $B \in \Sigma^0_n$ деп жазамыз. B кез келген болғандықтан $\Sigma^0_{n+1} \subseteq \Sigma^0_n$ болатыны шығады. Бұл (a) бөліктегі қорытындыға қайшы келіп тұр.

3. Енді (a) және (c) ақиқат. Рекурсия жайлы теорема дәлелдемесі (33.1 теорема) A-дан φ-ге сөзбе сөз декларация жасалады және A-дан σ-ға декларациялы немесе декларациясыз қайталанады. (b)-ны теріске шығарғымыз келсе, тоқтау проблемасы үшін қозғалыссыз нүктесі жоқ оракулді жалпы σ-ны құрастыра аламыз. $K=\{x \mid \varphi_k(x)\downarrow\}$ болсын, және σ^K бейне

$$\sigma^K(x) \;=\; \begin{cases} \mathrm{const}(\varphi_x(x)+1), & \text{егер} \quad \varphi_x(x)\downarrow, \\ \mathrm{const}(0), & \text{егер} \quad \varphi_x(x)\uparrow \end{cases}$$

анықталсын. Функция σ^K жалпы. Енгізу x-те ол өзінің оракуліне екі жағдайдың қайсысын қолдану жайлы консултация береді. Егер $\varphi_x(x)\uparrow$ болса, онда ол $\mathsf{const}(0)$-ді шығарады. Егер $\varphi_x(x)\downarrow$ болса, онда ол $\varphi_x(x)$-ті тікелей есептейді (тоқтау керек екенін ол біледі), содан кейін 1-ді қосады және алынған мәнге const-ты қолданады. Сонымен, барлық x үшін,

$$\begin{aligned} \varphi_{\sigma^K(x)}(x) \;&=\; \begin{cases} \varphi_x(x)+1, & \text{егер} \quad \varphi_x(x)\downarrow, \\ 0, & \text{егер} \quad \varphi_x(x) \end{cases} \\ &\neq\; \varphi_x(x), \end{aligned}$$

сондықтан σ^K-ның қозғалыссыз нүктесі жоқ.

4. Айталық, рекурсивті граф (ω, E) Тюринг машинасымен өрнектелген болсын, ал $(x,y)\in E$ болсын және x, y бинарлы жолдар болып графта жолдар жиынына $x\#y$ қабылдау-ашық болсын. M Тюринг машинасымен өрнектелген графты G_M деп белгілейік. Біз келесі жиынның

$$\mathrm{SC} \;=\; \left\{ M \mid G_M \ \text{қатаң байланысқан} \right\}$$

Π_2^0-толық болатынын көрсеткіміз келеді. SC жиыны Π_2^0-де жатады, сондықтан оны предикат Π_2^0-мен анықтауға болады:

$$\mathrm{SC} \ = \ \{M \mid \forall x \ \forall y \ \exists \sigma \ \exists \tau \ \mathsf{path}(M, x, y, \sigma, \tau)\},$$

мұндағы $\mathsf{path}(M, x, y, \sigma, \tau)$ айтатыны: $x = x_0$, $x_n = y$ болатындай $x_0, x_1, \ldots, x_m$ және $t_0, t_1, \ldots, t_{n-1}$ тізбектерінің натурал санды кодталуы сәйкес σ мен τ және $0 \le i \le n-1$ үшін бірнеше t_i қадам ішінде M -де $x_i \# x_{i+1}$ -ге қабылдау-ашық. Π_2^0 үшін SC күрделі болатынын көрсету үшін, кез келген Π_2^0 жиынды

$$\{x \mid \forall y \ \exists z \ R(x, y, z)\}$$

SC -ға келтіреміз. Егер x берілсе, онда графты

$$\{(2y, 2z+1) \mid R(x, y, z)\} \cup \{(2z+1, \omega) \mid \omega, z \in \omega\}$$

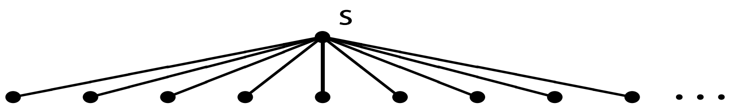
қарастырамыз. Егер тек егер $\forall y \ \exists z \ R(x, y, z)$ болса, онда бұл граф қатаң байланысқан болады. Берілген x-тегі осы қабырғалар жиынына қабылдау-ашық машинаны және R рекурсивті қатынас үшін де машинаны жеңіл құрастыра аламыз.

12-үй жұмысының шешімі

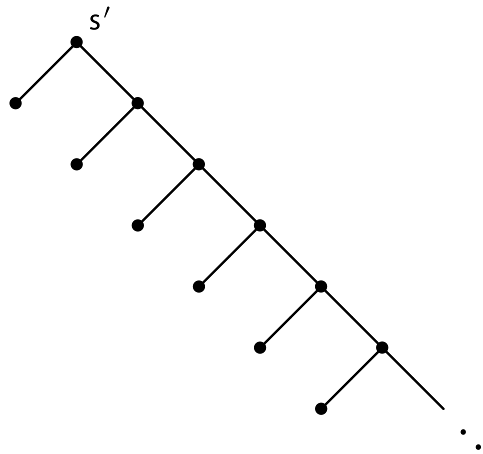
1. (a) Келесі кодпен алмастыру арқылы әрбір енуден $y := \exists$ құтыламыз.

$$\begin{array}{lll} y := 0 & & \\ \ell_1 : & \ell_2 \vee \ell_3 & \\ \ell_1 : & y := y+1 & \\ & \text{goto } \ell_1 & \\ & \ell_3 : & \ldots\ . \end{array}$$

Ол әрбір есептелетін экзистенциалды тармақталуды



келесі түрге алмастырады



бірақ егер тек егер s түйін 1-мен белгіленсе, онда s'-та 1-мен белгіленеді. Түйін s'-тен төмен қарай шексіз жол кететіндіктен, бұл $\forall$-тармақ үшін жұмыс жасамайды. Бұл жаңа программаны p деп белгілейік.

Енді бейдетерминирлі Тьюринг машинасын детерминирлі машинамен симуляция жасағанды пайдаланып, p конструкциядағы $\ell_i \vee \ell_j$ -ден құтыламыз, мұнда тек мынадай айырмашылық бар - ол кейбір қадамдарда анда-санда $y := \forall$ пайда болып тұрады.
Симуляция жасайтын p' программа p-ның конфигурациясының шектеулі тізімін $(\ell_1, \beta_1), \ldots ,(\ell_n, \beta_n)$ сақтайды, ол тізімде ағымдағы уақытта симуляция жасалады, мұндағы ℓ_i p-ның белгісінің тұжырымы және $\beta_i : \{y_1, \ldots ,y_k\} \to \omega$ p-ның айнымалыларының бағасы. Тізімдегі әрбір (ℓ, β) конфигурациясы үшін p-ның бір қадамын симуляция жасайтын және сәйкес (ℓ, β) -ны жаңартатын p' программасы циклдік режімде тізім бойымен жүгіріп шығады. Ал p программа жұмыста болған кезде, p' программа $\forall$ -тұйықталуды жасайды. (ℓ, β) -да p' программа p -ны симуляция жасағанда және ℓ формадағы $\ell : \ell_i \vee \ell_j$ тұжырым болған барлық жағдайда, p' программа (ℓ, β) -ны тізімнен жойып жібереді де оны (ℓ_i, β) және (ℓ_j, β) -мен алмастырады. Бұл екі есептеудің эквиваленттілігін итерацияланған дистрибутивтілік деп атауға болар еді. p' программаның тек қарапайым меншіктеулері $y := e(y)$ және универсал меншіктеулері $y := \forall$ бар. Сөз соңында, егер өзі симуляция жасайтын конфигурациялардың кез келгенінің біреуінде қабылдау-ашық тұжырым болса, онда p' тоқтайды және қабылдау-ашық күйге көшеді.

(b) Берілген рекурсивті қатынас фундирлі, Π_1^1-күрделі болатынын анықтау проблемасының шешімін көрсету үшін мына жағдайды ескерейік: егер тек егер p'-та қабылдау ашық болса, онда кез келген енгізуде есептеуші p' бұтақ фундирлі және рекурсивті бұтақ болады. Дегенмен, егер тек егер оригинал программа p-да қабылдау-ашық болса, онда p'-та да қабылдау ашық болады және қабылдау-ашық IND программа Π_1^1-күрделі болады, сондықтан Клин теоремасы (40.1-теорема) бойынша $\mathbb{N}$-де IND программада Π_1^1-ге тура қабылдау-ашық.
Осыған қосымша проблема Π_1^1-де жатады, өйткені 39-лекцияда көрсетілгендей оған IND программада қабылдау-ашық.

2. Біздің мұндағы жоспар, 111-ші аралас жаттығуда берілген жалқау шартты тексеруді

$$\varphi_{\mathrm{cond}(i,j)}(x,y) = \begin{cases} \varphi_i(y), & \text{егер} \quad x=0, \\ \varphi_j(y), & \text{егер} \quad x\neq 0 \end{cases}$$

және рекурсия жайлы теореманы (33.1-теорема) пайдаланып, төменде өте қажет

$$h(x,y) = \begin{cases} y, & \text{егер} \quad f(x,y)=0, \\ h(x,y+1), & \text{егер} \quad f(x,y)\neq 0, \end{cases} \tag{12}$$

болатын h функцияны құрастыру. Енді

$$g(x) \stackrel{\text{def}}{=} h(x,0)$$

деп аламыз.

Ендігі кезекте f, π_1^2, π_2^2 мұрагерлік функция және тепе-теңдік функциялардың j, a, b, s және i сәйкес индекстері болсын, және

$$\sigma(x) = \mathsf{pair}(\mathsf{cond}(b,\mathsf{comp}(j,\mathsf{pair}(a,\mathsf{comp}(s,b)))),\mathsf{pair}(x,i)). \tag{13}$$

болсын. Функция σ жалпы, сондықтан рекурсия теоремасы бойынша оның қозғалыссыз нүктелері $h=\varphi_x=\varphi_{\sigma(x)}$ болады, осының әсерінен

$$h = \varphi_x = \varphi_{\mathsf{pair}(\mathsf{cond}(b,\mathsf{comp}(j,\mathsf{pair}(a,\mathsf{comp}(s,b)))),\mathsf{pair}(x,i))}.$$

Қысқаша айтсақ, анықтамаларды жақсылап сығымдасақ (12)-ні аламыз.

σ-дің индексін j-ден тиімді түрде алуға болады, өйткені (13)-ке сүйенетін болсақ, j-мен және белгілі бір тұрақтының комбинациясын, композиция мен жұптар пайда болуын пайдаланып σ өрнектеледі. Және рекурсия жайлы теореманы (10-шы үй жұмысы, 2-ші жаттығу) тиімді редакциялап, σ-ның индексінен тиімді түрде h-тың индексін алуға болады.

Бірақ сіз басқаша ойлауыңыз мүмкін, сөйтсе де бұл жаттығу адамды шыдамдылыққа тексеру тәжірибесі ретінде құрастырылған жоқ. Оның мақсаты программалаудағы қиындықпен көпшілікті таныстыру еді, Гёдел және оның әріптестері, 1930 жылы қазіргі программалау тілдерін ойлап табу процесінде жоғарыда баяндалған қиындықтармен күресуіне тура келді. Бұл конструкциялар қаншалықты шытырман болғанымен, қазіргі таңда біз қолданып жүрген, аса бұратылған күрделі программалау конструкциясының алғышарттарын осыдан көруге болады.

3. Кез келген $T(n) \geq \log n$ үшін 2.5-теоремаға сай,

$$DTIME(T(n)) \subseteq DSPACE(T(n)) \subseteq DTIME(T(n)^{T(n)})$$

қатынасы орындалады. Үзіліс жайлы (32.1-теорема) теоремаға сүйенсек, T табылып

$$DTIME(T(n)) = DSPACE(T(n)) = DTIME(T(n)^{T(n)}).$$

қатынасы орындалады.

Таңдап алынған аралас жаттығуларға көмектер

8. (b) Қарастырыңыз $\{x\#x \mid x \in \Sigma^*\}$.

14. 11-ші аралас жаттығуды пайдаланыңыз.

15. Саусақтардың шектеулі саны.

27. Енгізуді толтырыңыз.

31. Шектеулі автоматты конвертация жасаңыз және 6-үй жұмысын, 2-жаттығу және 15-аралас жаттығуларды пайдаланыңыз. Регулярлы өрнектер жайлы ақпаратты [76, 7-9 лекцияларды] қараңыз.

36. 33-ші аралас жаттығуды пайдаланыңыз.

38. (a) Индукция.

40. Бүтін санды сортировкаға келтіріңіз.
42. Бәрімізге таныс бүтін сандарды бөлу амалын қолданып, m-ді n-ге бөлгенде, $n \neq 0$ болып және q, r сәйкес бөлінді және қалдық болсын. Сонда егер $m = nq + r$ болса, мұндағы $0 \leq r < n$, онда $\gcd(m,n) = \gcd(n,r)$ болатынын көрсетіңіз.

43. 42-ші аралас жаттығуды және a мен n-нің барлық бүтін мәнді комбинациясы $\gcd(a,n)$-ге еселі болады деген фактыны пайдаланыңыз.

45. Шартты математикалық күтім анықтамасын пайдаланып, егер $\Pr(F_i) \neq 0$ және $F = \bigcup_i F_i$ орындалатын F_i қиылыспайтын оқиғалар болса, және егер X кез келген кездейсоқ оқиға болса, онда

$$\mathcal{E}(X \mid F) = \sum_i \mathcal{E}(X \mid F_i) \cdot \Pr(F_i \mid F)$$

орындалатынын көрсетіңіз.

46. Бұл көмек (a) және (b) екеуіне де қолданылады. 3CNF формула арқылы берілген айнымалыларға ақиқатты меншіктеу s, t болсын. Тұжырымды қарастырыңыз, "Лексиграфикалық реттілікте ақиқатқа меншіктеу u табылып, $s \leq u \leq t$ орындалады және барлық ақиқатқа меншіктеу υ үшін, υ-мен қанағаттандырылған φ-дің клоздар саны u-мен қанағаттандырылған клоздар санынан артық емес". Осы фактіні пайдаланыңыз, егер $P = NP$ болса, онда $\Sigma_2^{\mathrm{p}} = P$. Бинарлы іздеу жасаңыз.

47. (b) Максималды клик өлшемі қабылдау-ашық ықтималдығымен өзара тығыз байланыста болатынын көрсету үшін (a)-ны пайдаланыңыз.

49. (a) QBF -ті кодтаңыз.

50. Тақтадағы жағдайды $\equiv_{n,k}^{m}$ эквиваленттілігінің эквиваленттілік класы ретінде қарастырыңыз, мұндағы $\equiv_{n,k}^{m}$ эквиваленттілік индуктивті түрде төмендегідей анықталады.

• Егер тек егер $\mathcal{A}, a_1,\dots,a_k \equiv_{0,k}^{m} \mathcal{B}, b_1,\dots,b_k$ болса, онда ұзындығы m-нен аспайтын $x_1,\dots,x_k$-дағы еркін айнымалыларда жиындар $\mathcal{A}, a_1,\dots,a_k$ және $\mathcal{B}, b_1,\dots,b_k$ сәйкестіндіріледі; яғни, егер осындай барлық формула φ үшін егер тек егер

$\mathcal{A}, a_1,\dots,a_k \vDash \varphi$ болса, онда $\mathcal{B}, b_1,\dots,b_k \vDash \varphi$ болады.

• Егер тек егер $\mathcal{A}, a_1,\dots,a_k \equiv_{n+1,k}^{m} \mathcal{B}, b_1,\dots,b_k$ болса, онда келесі шартардың екеуі де орындалады:

(a) Барлық $a_{k+1} \in \mathcal{A}$ үшін $b_{k+1} \in \mathcal{B}$ табылып
$\mathcal{A}, a_1,\dots,a_k,a_{k+1} \equiv_{n,k+1}^{m} \mathcal{B}, b_1,\dots,b_k,b_{k+1}$
қатынасы орындалады.

(b) Барлық $b_{k+1} \in \mathcal{B}$ үшін $a_{k+1} \in \mathcal{A}$ табылып
$\mathcal{A}, a_1,\dots,a_k,a_{k+1} \equiv_{n,k+1}^{m} \mathcal{B}, b_1,\dots,b_k,b_{k+1}$
қатынасы орындалады.

53. (b) Егер қабылдау-ашық жол жоқ болса, онда $|x|$ және $|y|$ үшін $2^{O(n\log n)}$-нен аспайтын форманың біреуі xy^{ω} табылатынын көрсетіңіз, мұндағы n күйдің саны.

60. (b) Барлық j үшін i, $i \geq j$ табылып $A_{ij} \neq 0$ болатындай $n \times n$ кездейсоқ A матрицаны құрастырамыз. Дәл айтсақ осындай $\Pi_{i=0}^{n-1}(q^n - q^i)$ матрица бар, бейсингулярлы матрица саны да осындай. Стандартты базис элементтерін және A матрицаның бұрын генерацияланған x_i бағандарын сызықты комбинацияның коэффициенттері ретінде пайдаланып, стандартты базистен басталатын (бірлік матрицаның бағандары) сызықты тәуелсіз $x_1,\dots,x_n$ векторлар тізбегін құрастырыңыз.

66. (iii) Күшейту (14.1-лемма) және қосу ережесін (13-лекция) пайдаланыңыз.

(iv) Күшейтуді пайдаланыңыз (14.1-лемма).

(v) 63-аралас жаттығуды пайдаланыңыз.

(vi) BP үшін, күшейтуді пайдаланыңыз (14.1-лемма).

68. 6-аралас жаттығуды пайдаланыңыз.

69. (c) M -ді симуляция жасау кезінде оракул сұратым жасамайтын, басқа Σ_k оракул N машинамен M -ді симуляция жасаңыз, ол N машина сұратым жолдарын жазып отырады және оракул жауабын іздейді, осыдан кейін есептеу соңында алынған жауапты тексереді. Есептеудің кез келген траекториясында кезектесу саны артып кетпеуі және іздеу не универсалды немесе экзистенциалды болатынын көрсету үшін (a) және (b)-ны пайдаланыңыз.

70. Структура φ-дегі индукция. Біршама қатаң индуктивті гипотезаны пайдаланыңыз: Логикалық $x_1,\ldots,x_n$ айнымалылы және басқа да мүмкін айнымалылы, кез келген $\varphi(x_1,\ldots,x_n)$ логикалық формула үшін

$$\begin{aligned}&\varphi(\operatorname{excl}(z,\sigma_1,\tau_1)),\ldots,\varphi(\operatorname{excl}(z,\sigma_n,\tau_n))\\ &\quad\equiv\ \operatorname{excl}(z,\varphi(\sigma_1,\ldots,\sigma_n),\varphi(\tau_1,\ldots,\tau_n))\end{aligned}$$

қатынастар орындалады.

89. Кеңейтілген Тейлор қатарының алғашқы бірнеше мүшесімен $\ln(1-\delta)$-ді аппроксимацияланыз. $|x|<1$ үшін орындалатын

$$\ln(1+x)\ =\ \sum_{n=1}^{\infty}\frac{(-1)^{n+1}}{n}x^n$$

теңдікті еске түсіріңіз.

93. 92-аралас жаттығуды пайдаланыңыз.

94. $A(m,n)$-ді $A_m(n)$ арқылы жазсақ, онда $A_{m+1}(n)=A_m^{n+1}(1)$ болады, мұнда f^n арқылы f-тің n-еселі өз-өзіне жасалған композициясы белгіленген:

$$f^0(x) \overset{\text{def}}{=} x$$

$$f^{n+1}(x) \overset{\text{def}}{=} f(f^n(x)).$$

91-аралас жаттығудың екінші сипаттамасын пайдаланыңыз.

96. Берілген x енгізуде кез келген нөл емес шешім, берілген ТМ M -нің қабылдау-ашық есептеу тарихы болатын, PCP үлгісін құрастырыңыз. Белгілі бір k мен G={0,1, #} үшін S={0,1,...,k-1} болсын. Және f(0)= # α_0 # және g(0)= # болсын, мұндағы α_0 x -тегі M -нің алғашқы конструкциясы.

98. Рекурсия жайлы теореманы пайдаланыңыз.

99. Белгілі бір ішкі жиынды өсу ретімен нөмірлеңіз.

100. (b) Енгізу x -те 0,1,...,x енгізулі M_i -ді симуляция жасайтын машина $M_{\sigma(i)}$ болсын және егер тек егер M_i машина барлық 0,1,...,x енгізуде тоқтаса және x -ке қабылдау-ашық болса, онда ол машинада қабылдау-ашық болады.

107. $(R\circ S)^{-1}=S^{-1}\circ R^{-1}$ және $(R^{-1})^{-1}=R$ болатынын көрсетіңіз. Осы фактіні пайдаланып, $R\circ R^{-1}\circ R$ H R қатынасы $R^{-1}\circ R\circ R^{-1}$ H R^{-1} қатынасқа меңзейтінін көрсетіңіз. Енді инварианттылықты сақтай отырып, әрқайсысы шектеулі анықталу аймағымен болатын аппроксимациялар $h_0, h_1,...$ тізбесін $h{:}\omega\rightarrow w$ бейнеге апару үшін 34-ші лекциядағыдай алға-артқа аргументін пайдаланыңыз, мұнда әрбір h_n өзінің анықталу аймағында бірдің-бірге қатынасы және h_n H R.

108. Жалпылықты сақтай отырып, σ мен τ бірдің-бірге қатысы деп ұйғару үшін тиімді толтыруды (34.2-лемма) пайдаланыңыз, осыдан кейін 107-аралас жаттығуды қолданыңыз.
109. 107-аралас жаттығуды қолданыңыз.

113. 112-аралас жаттығуды қолданыңыз.

114. (b) $\{x \mid \varphi_x(x)$ жұп $\}$ және $\{x \mid \varphi_x(x)$ тақ $\}$ жиындарын қарастырыңыз.

115. Рекурсия жайлы теореманы пайдаланыңыз.

116. 12-ші үй жұмысын, 2-жаттығуды пайдаланыңыз.

118. (a) Кез келген 0,1-мәнді күрделі функциялар табылатынын дәлелдеңіз.

121. Рекурсия жайлы теореманы пайдалансаңыз, дәлелдеме тек екі жолға сыяды.

124. Диогоналдаңыз, содан кейін 123 (b) аралас жаттығуды пайдаланыңыз.

125. (b) F кез келген болсын және $\Psi_i \overset{\text{def}}{=} \Phi_i + \varphi_i + 1$ -ді анықтаңыз.

127. (c) Тұрақты функцияны тізбелеу $\varphi_{\text{const}(k)}$, $k \geq 1$ көмегімен беріледі. Күрделіктің кез келген абстракциялық өлшемі F болсын. Күрделіктің жаңа өлшемін

$$\Psi_i(n) \overset{\text{def}}{=} \begin{cases} 0, & \exists k \quad i = \text{const}(k) \qquad M_k(k)\uparrow^n \\ \Phi_i(n)+1, & \end{cases}$$

арқылы анықтаймыз.

132. COF-кодтаңыз. 110 (b) аралас жаттығудағы A жиынын пайдаланыңыз, мұндағы айырмашылық A -ны құрастырмас бұрын L-дағы жиындар комплементін қосыңыз. $\mathrm{L}(\mathrm{M}_{\sigma(\mathrm{i})}) = \{\mathrm{f}(x) \mid x\mathrm{O}\, L(M_i)\}$ $\cup A$ болсын, мұндағы $\mathrm{f}(x)$ $\sim A$ -ның x-шы элементі.

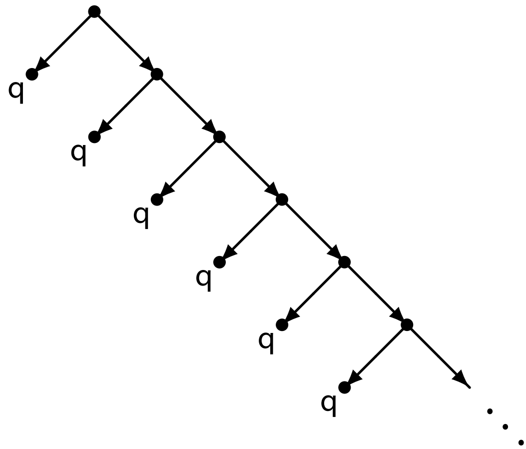
133. Үш қоянды бір оқпен атып алыңыз.

135. Дәлелдемені тізбелеңіз. Дәлелденетін жалпы рекурсиялы функцияның барлығынан асимтотикалық жылдам өсетін жалпы рекурсиялы функцияны құрастырыңыз.

137. Тиімді толтыруды қолданыңыз.

138. Нақты жауап, сіз ойлаған жауаптан басқа болуы мүмкін.

139. Бұл өте қауіпті есеп. Төменде келтірілген мысал контр-мысал болып шықпас үшін, шешімді қайта тексеріңіз.



Таңдап алынған аралас жаттығулардың шешімі

6. Енгізу x-те $G(x)$ кеңістігінде жұмыс жасайтын M машинасының көмегі арқасында $A \subseteq DSPACE(G(x))$ болады деп айталық. Және

$$A' \overset{\text{def}}{=} \left\{ x\#^k \mid M \text{ - енгізу } x \text{ -те } k + |x| \text{ кеңістігінде тоқтайды} \right\}$$

$$A'' \overset{def}{=} \left\{ x\#^k \mid M \text{ - тоқтайды және } k + |x| \text{ кеңістікте } x\text{-ке қабылдау-ашық} \right\}$$

$A', A'' \in DSPACE(n)$. Ұйғарым бойынша екеуі де $A', A'' \in DTIME(T(n))$, сәйкес M' және M'' машина-ларының әсерінен болады деп айталық, олар ұзындығы n болатын енгізулерде $T(n)$ уақыт жұмыс жасайды. Енді N машинасын келесі шартпен құрастырамыз: егер x берілсе, онда ол $i = 0,1,2,\ldots$ үшін енгізу $x\#^i$ -де M' -ті қабылдау-ашық болғанша іске қосады, ол процесс $i = G(x) - |x|$ уақыттан артық алмауы қажет. Табылған максималды i үшін $x\#^i$ -де N машина M'' -ті іске қосады, ондағы мақсат $x \in A$ ақиқат па соны анықтау және оған сәйкес қабылдау-ашық немесе қабылдау-жабық болады. Сонымен $L(N) = A$ болып шығады. Әрбір xa^i үшін симуляция $T(i + |x|)$ уақыт талап етеді, ал уақыт T -ның монотондығының әсерінен $T(G(x))$ -ден аспайды. Сондықтан жалпы уақыт $(G(x) - |x| + 2)T(G(n))$ -нен аспайды.

17. [63, 377ff. бетті] қараңыз. Мұнда алтернативті TM -ды пайдаланып едәуір қарапайым дәлелдеме келтіреміз. Детерминирлі уақыт және алтернатив кеңістік арасындағы қатынасқа (7.5-салдар) сүйеніп, бейдетерминирлі және детерминирлі $S(n)$ -кеңістікті шектеулі APDA -лар $S(n)$ -кеңістікті шектеулі алтернатив TM -дарға эквивалент болатынын көрсетсек жеткілікті болады.

Сонымен M бейдетерминирлі $S(n)$ -кеңістікті шектеулі APDA болсын. Жалпылықты сақтай отырып, қабылдау-ашыққа көшерде M өз стегін тазалап тастайды деп ұйғарайық. Конфигурация жұмыс таспасының мәнінен, соңғы бақылаудан, жұмысшы таспаның бас тиегінің орнынан және стек төбесіндегі белгі немесе арнайы жалаушадан тұрады, белгі немесе жалауша стек бос па соны көрсетіп тұрады. Жоғарғы символдан төменгі басқа стек мәнін ол қамтымайды. Конфигурация $S(n)$ кеңістікте көрсетілуі мүмкін.

Егер σ стекті α конфигурациясынан басталатын және σ стекті β конфигурациясымен аяқталатын есептеу бар болса, онда α, β конфигурация және σ стек үшін $\alpha \to \beta$ деп жазамыз. Бұл есептеу σ-ның ең жоғарғы элементін барлық уақытта өшірмейді. Есептеу стекте σ-дан жоғары тұрған элементтерді жоғары ығыстыра алады және ол элементтердің қаншасын жойғысы келсе, соншасын жоя алады, бірақ ол σ-ға тиісе алмайды. Назар аударыңыз, σ-ның мәні, α мен β-да көрсетілген ең жоғарғы символдан басқасы, есептеуге көрінбейтін болғандықтан, сұрақ $\alpha \to \beta$ σ-дан тәуелді емес.

Енді $\alpha \to \beta$ орындала ма, соны анықтау үшін рекурсивті процедураны сипаттаймыз. Бұл x-ке M -де қабылдау-ашық па - соны білу үшін қолданылады. M машина алдымен $\mathsf{start} \to \mathsf{accept}$ бола ма соны тексереді, мұндағы start және accept M -нің сәйкес бастапқы және қабылдаушы конфигурациялары, осыларды жалпылықты сақтай отырып ұйғара аламыз. Процедура $S(n)$ кеңістікте алтернатив TM -да іске асырылуы мүмкін.

Процедура былай жұмыс жасайды. Егер α және β берілсе, онда процедура алдымен $\alpha = \beta$ бола ма соны тексереді немесе α стекте ығыстырусыз немесе жоюсыз бір қадамда β-на ала ала ма соны тексереді, егер солай болса, онда ол бірден қабылдау-ашық күйге көшеді. Егер олай болмаса, онда процедура бейдетерминирлі түрде $\alpha \to \gamma$ және $\gamma \to \beta$ орындалатын аралық конфигурация γ табыла ма сонымен айналысады. Егер солай бола қалса, онда ол γ-ны $\vee$ -тармақ арқылы іздейді, содан кейін $\wedge$-тармақты пайдаланып параллельді түрде $\alpha \to \gamma$ және $\gamma \to \beta$ бола ма соны тексереді. Кері жағдайда $\alpha \to \beta$ болуы үшін төмендегі шарттарға бағынатын α' және β' табылуы қажет: α конструкция α'-ты стектегі символды ығыстыру арқылы бір қадамда өндіруі қажет, β' конструкция β-ны стектегі символды жою арқылы бір қадамда өндіруі қажет және $\alpha' \to \beta'$ қысқа есептеледі. Процедура α' пен β'-ты табады, α бір қадамда α'-ты өндіре ме және β' бір қадамда β-ны өндіре ме соны тексереді, осыдан кейін өзін-өзі құйрықты рекурсиямен шақырып, $\alpha' \to \beta'$ бола ма соны тексеріледі. Мұнда α мен β-ны есте сақтаудың қажеті жоқ, сондықтан $S(n)$-нен аспайтын кеңістік қажет болады.

Керісінше, $S(n)$ -кеңістікті шектелген алтернатив TM , сондықтан N машина детерминирлі $S(n)$ -кеңістікті шектелген APDA -мен симуляциялана алады. APDA есептеу бұтақ N -нің тереңінен іздеу жүргізеді, сонымен қатар бұтақты конструкциялайды және қабылдау-ашық/қабылдау-жабық мәндерді рекурсивті есептейді. Ол өзінің стегін тереңдете іздеу үшін қолданады, осы процесте ол бұтақтың қай жеріне келгенін стек арқылы білетін болады.

20. Жиынның шектеулі операторы тізбелі-шектеулі болады: егер τ шектеулі және $\mathcal{C}$ тізбе болса, онда

$$\begin{aligned}
\tau(\bigcup\mathcal{C}) = & \bigcup\{\tau(B) \mid B \subseteq \bigcup\mathcal{C},\ B \text{ шектеулі}\} \\
= & \bigcup\{\tau(B) \mid \exists C \in \mathcal{C} B \subseteq C,\ B \text{ шектеулі}\} \\
= & \bigcup \bigcup_{C\in\mathcal{C}} \{\tau(B) \mid B \subseteq C, \quad \text{шектеулі}\} \\
= & \bigcup_{C\in\mathcal{C}} \bigcup \{\tau(B) \mid B \subseteq C,\ B \quad \text{шектеулі}\} \\
= & \bigcup_{C\in\mathcal{C}} \tau(C).
\end{aligned}$$

Тізбелі-шектеулі оператордың шектеулі болатын фактісін, транс-шектеулі индукцияны пайдаланып келесі түрде көрсетуге болады. Еске салайық, егер $f : A \underset{\text{onto}}{\overset{1-1}{\longrightarrow}} B$ болатын f табылса, онда $A \equiv B$ болады. A жиынының қуаты, $\alpha \equiv A$ орындалатын ең кіші ординал. A-ның қуаты не шектеулі немесе шекті (limit) ординал, өйткені шексіз α үшін $\alpha + 1 \equiv \alpha$ болады ($\omega \le \beta < \alpha$ үшін $\beta \mapsto \beta$ деп бейнелейміз, ал $n < \omega$ үшін $n \mapsto n+1$ деп бейнелейміз, және $\alpha \mapsto 0$).

Енді кез келген тізбе $\mathcal{C}$ үшін, τ X-тегі оператор және $\tau(\bigcup\mathcal{C}) = \bigcup_{C\in\mathcal{C}} \tau(C)$ орындалады деп ұйғарамыз. Кез келген $A \subseteq X$ үшін α оның қуаты болсын және $f : \alpha \underset{\text{onto}}{\overset{1-1}{\longrightarrow}} A$ орындалсын. Егер A шектеулі болса, онда дәлелдейтін ештеңе жоқ. Кері жағдайда α шекті ординал. Кез келген $\beta < \alpha$ үшін $A_\beta = \{f(\gamma) \mid \gamma < \beta\}$ деп анықтайық. Сонда A_β тізбе құрайды және $\bigcup_{\beta<\alpha} A_\beta = A$ болады. Оған қосымша, A_β-ның қуаты аздау болады, оған себеп $A_\beta \equiv \beta < \alpha$ қатынасы, сондықтан индуктивті гипотезаға сүйенсек $\tau(A_\beta) = \bigcup \tau(B) \mid B \subseteq A_\beta,\ B$ шектеулі$\}$. Сондықтан

$$\begin{aligned}\tau(A) &= \tau\Big(\bigcup_{\beta<\alpha} A_\beta\Big)\\ &= \bigcup_{\beta<\alpha} \tau(A_\beta) \text{ үзіліссіздіктен}\\ &= \bigcup_{\beta<\alpha} \bigcup_{\substack{B\subseteq A_\beta\\ B \text{ finite}}} \tau(B)\\ &= \bigcup_{\substack{B\subseteq A\\ B \text{ finite}}} \tau(B).\end{aligned}$$

Соңғы теңдеу келесі факты (дерек) әсерінен жазылды: шектеулі B үшін егер тек егер $B \subseteq A$ болса, онда белгілі бір $\beta < \alpha$ үшін $B \subseteq A_\beta$ болады.

21. (b) ω-ның ішкі жиынында берілген жиын операторы

$$A \mapsto \begin{cases} A, \text{егер } A \text{ шектеулі}\\ \omega, \quad \text{шексіз болса}\end{cases}$$

монотонды, бірақ тізбелі-үздіксіз болмайды.

23. Егер τ тізбелі-үздіксіз болса, онда

$$\begin{aligned}\tau^{\omega+1}(\varnothing) &= \tau(\tau^{\omega}(\varnothing))\\ &= \tau\Big(\bigcup_{n<\omega} \tau^n(\varnothing)\Big)\\ &= \bigcup_{n<\omega} \tau(\tau^n(\varnothing))\\ &= \varnothing \cup \bigcup_{n<\omega} \tau^{n+1}(\varnothing)\end{aligned}$$

$$= \bigcup_{n<\omega} \tau^{n}(\varnothing)$$

$$= \tau^{\omega}(\varnothing).$$

Тұйық ординалы $\omega+1$ болатын τ үшін бір мысал: 40-лекцияның ішінде кескінделген бұтақ түйіндерінің жиыны *X* болсын және

$\tau(A) = \{x\text{-тың барлық мұрагерлері } A\text{-да орналасады}\}$

болсын. Сонда $\tau(\varnothing)$ жапырақ болады, $\tau^{2}(\varnothing)$ жапырақ және жапырақ үстіндегі түйіндер, т.с.с.; $\tau^{\omega}(\varnothing)$ түбірден басқа бүкіл түйіндер жиыны; $\tau^{\omega+1}(\varnothing)$ бүкіл түйіндер жиыны. Осы ең кіші тиянақты нүкте болады.

28. 2-лекцияда келтірілген Савич теоремасының дәлелдемесінің бұл жерде жақсартылған нұсқасы қарастырылады. Ұзындығы *n* болатын енгізуде *T*(*n*) уақытта және *S*(*n*) кеңістікте жұмыс жасайтын *M* бейдетерминирлі *TM* машина болсын (сондықтан *x* енгізуде *M* машинаның ешқандай есептеуі *S*(*n*)-нен артық кеңістікті және *T*(*n*)-нен артық уақытты жұмсай алмайды). Енгізу *x*-тегі *M*-нің алғашқы конфигурациясы start болсын, және *M* машинаның уникалды қабылдау-ашық конфигурациясы accept және уникалды қабылдау-жабық конфигурациясы reject бар деп ұйғарайық (сондықтан басқа ешқандай конфигурация тоқтай алмайды). Және де *M* машина қабылдау-ашық немесе қабылдау-жабық болу үшін, алдымен жұмыс таспасын өшіреді де сол жаққа тірелгенше жылжиды деп те ұйғарамыз. Енді Δ шектеулі алфавит Δ-да конфигурация *M* жол сияқты кодталады деп ұйғарамыз.

Егер *S*(*n*) және *T*(*n*) екеуі де *S*(*n*)log *T*(*n*) кеңістікте конструктивті болса, онда алдымен *T*(*n*) және *S*(*n*)-ді есептеп аламыз да, содан кейін SAV(start, accept, *T*(*n*), *S*(*n*))-ді шақырамыз, мұндағы SAV(α, β, t, s) 2-лекцияда сипатталған рекурсивті процедура. Бұл процедура ұзындығы *t*-дан аспайтын α-дан β-ға дейінгі есептеу жолын табуға тырысады, ол осы мақсатта ұзындығы *s*-тен аспайтын конфигурацияны пайдаланады. Әрбір рекурсивті SAV-іске асырғыш *S*(*n*) кеңістік талап етеді, ал рекурсия тереңдігі log *T*(*n*)-ге тең және ол кеңістікке *S*(*n*)log *T*(*n*)-ге тең шектеу қояды. *S*(*n*) мен *T*(*n*) конструктивті болмаған жағдайда, біз SAV процеду-раны аздап модификациялаймыз:

```
boolean EXACTSAV(α,β,t,s){
    if t = 0 then return α = β;
    if t = 1 then return α →(1) β;
    for ã ∈ Δ^s {
   if EXACTSAV(α,γ,⌈t/2⌉,s) ∧ EXACTSAV(γ,β,⌈t/2⌉,s)
            then return 1;
    }
    return 0;
}
boolean SAV(α,β,t,s){
    for t' := 1 to t{
    if EXACTSAV(α,β,t',s) then return 1;
    }
    return 0;
}
```

Мұндағы айырмашылық, ұзындығы s-тен аспайтын конструкция арқылы EXACTSAV (α, β, t, s) ұзындығы дәл t болатын α-дан β-ға дейінгі есептеу жолын табуға тырысады. Не SAV немесе EXACTSAV $s\log t$-дан асырып кеңістік пайдаланбайды.

Алдымен $S(n)$ мен $T(n)$-ді анықтап аламыз. $S=T=1$-дан бастаймыз және кезекпен не T немесе S өте аз ба соны тексереміз, егер иә болса, онда 1-ге арттырамыз да тағы бір рет қайталаймыз. T өте аз ба соны тексеру үшін, өз кезегінде барлық тоқтамайтын конфигурациялар $\alpha \in \Delta^S$ үшін EXACTSAV (start, α, T, S)-ті шақырамыз. Егер осы процедура әйтеуір бір уақытта қайтаратын болса, онда T өте аз. Ал S өте аз ба соны тексеру үшін барлық конфигурациялар $\alpha \in \Delta^S$ үшін SAV (start, α, T, S)-ті шақырамыз, ол бастиек жұмыс таспасының S-ұяшығын сканерлейді және жұмыс таспасының бас тиегін оң жаққа жылжытқысы келеді, сонымен ол $S+1$ кеңістікті пайдаланады. Егер осы процедура әйтеуір бір уақытта қайтаратын болса, онда S өте аз. Ең соңында, барлық есептеу осы шекен шықпайтын, S пен T-ның жеткілікті үлкен мәндерін табамыз. Осы сәтте SAV (start, accept α, T, S)-ті шақырамыз.

29. Проблеманың *PSPACE*-те жататынын көрсету үшін барлық M_i-де қабылдау-ашық жолды табамыз және ол қабылданды ма соны тексереміз. Біз әрбір M_i-дің алғашқы күйінің құмалағынан бастаймыз, содан кейін енгізу жолын символ артынан символ жіберіп таңдаймыз, осы жерде құмалақтарды M_i-дің күйі арқылы олардың алмасу функциясына сай жылжытамыз. Біз таңдалған жолды есте сақтауға міндетті емеспіз, тек осы сәттегі M_i-дің күйін есте сақтасақ жеткілікті. Егер әйтеуір бір кезде өздерінің автоматтарына сәйкес барлық құмалақтар қабылдау-ашық күйді ұстайтын ситуацияға келсек, онда бізде қабылдау-ашық. Ал құмалақтардың конфигурациясы полиномиалды кеңістікте көрсетілуі мүмкін болғандықтан, бұл бейдетерминирлі *PSPACE* есептеу. Оны Савич теоремасын пайдаланып детерминирлі жасауға болады.

Проблеманың *PSPACE*-те күрделі болатынын көрсету үшін N кез келген детерминирлі n^k-кеңістікті шектеулі ТМ болсын. Жалпылылықты сақтай отырып, N-де уникалды қабылдау-ашық конфигурация бар деп ұйғарайық. Егер ұзындығы n болатын x енгізуі берілсе, онда $O(n^k)$ күйлі n^k семиялы детерминирлі шектеулі автоматтар M_i-ді құрастырамыз, олардың әрқайсысының x енгізуіндегі қиылысуы N-нің қабылдау-ашық есептеу тарихының жиыны. Сонда егер тек егер $\bigcap_{i=1}^{n} L(M_i) = \varnothing$ болса, онда N машинада x-ке қабылдау-ашық емес.

Еске саламыз, ұзындығы n болатын енгізу x-те N машинаның қабылдау-ашық есептеудің тарихы

$$\# a_0 \# a_1 \# a_2 \# \quad \cdots \quad \# a_{m-1} \# a_m \#, \tag{14}$$

формадағы жолмен өрнектеледі, мұндағы әрбір a_i белгілі бір шектеулі Δ алфавиттегі ұзындығы n^k болатын жолды көрсетеді, олар x енгізуде N-нің конфигурациясын келесі түрде

(i) N-нің x-тегі алғашқы конфигурациясы a_0 ,

(ii) N-нің x-тегі қабылдау-ашық конфигурациясы a_m, және

(iii) N-нің алмасу ережесіне сүйеніп әрбір a_i-ден a_{i+1} алынады.

Егер жол қабылдау-ашық есептеуінің тарихы болса, онда ол дұрыс (14)-форматта көрсетілуі қажет және (і), (іі) және (ііі) шарттарын қанағаттандыруы қажет. Енгізу жол (14)-формада көрсетілгенін тексеру үшін келесі тексерулер атқарылады: енгізу жолы регуляр жиында $(\#\Delta^{n^k})^*\#$ жата ма, одан басқа да форматты белгілі бір қарапайым тексеру (конфигурацияда N тек бір күйде ғана болады, әрбір конфигурация соңғы маркерден басталып соңғы маркермен аяқталады, т.с.с.). Бұл $O(n^k)$ күйлі автоматты қажетсінеді. Шарттар (і) немесе (іі)-ні тексеру, енгізу басталды ма немесе ұзындығы n^k болатын белгілі бір жолмен аяқтала ма деген сияқты қарапайым тексерулерден тұрады. Тағы да, осы шарттардың әрқайсысын $O(n^k)$ күйлі автомат көмегімен тексеруге болады. Соңында, (ііі)-ті тексеру үшін Кук-Левин теоремасының дәлелдемесінен еске түсірейік: a_i-дің $j\text{-}1$-ші, j-ші және $j+1$-ші символдарын және a_{i+1}-дің j-ші символын қамтитын шектеулі жиынды локалды шарттар табылып, егер тек егер (ііі)-ші шарт орындалса, онда барлық j , $1 \leq j \leq n^k$ үшін осы локалды шарттар қанағаттандырылады. Локалды шарттар тек N-нің сипаттамасына тәуелді. (ііі)-тің орындалатынын тексеру үшін, $O(n^k)$ күйлі n^k автоматты қолданамыз. a_i-дің $j\text{-}1$-ші, j-ші және $j+1$-ші символдарын және a_{i+1}-дің j-ші символын қамтитын локалды шарттардың қанағаттандырылатынын, j-ші автомат сканерлейді және барлық i үшін тексереді. Бұл амал әрбір конфигурацияда қашықтықты анықтауды қажетсінеді, соңғы бағалауда үш символда есте сақтай отырып әрбір сыртқы $\#$-дан j-ға дейін қашықтықты анықтау, осыдан кейін келесі $\#$ көшіп осыдан бастап j қашықтықты анықтау және символдарды салыстыру. Әрбір машина санауды жүргізу үшін тек $O(n^k)$ күйді қажет етеді.

30. Алдымен, егер $P=NP$ болса, онда әрбір детерминирлі, полиномиалды уақытта есептелінетін, ұзындығын сақтайтын бейне $f:\Sigma^* \to \Sigma^*$ керіленетін бейне болатынын дәлелдейміз.

Назар аударыңыз, келесі шешім дұрыс емес.

NP-да x-ті тани аламыз және $f(x) = y$ болатынын тексере аламыз. Ал $P=NP$ болғандықтан, осының бәрін детерминирлі жасай аламыз.

Бұл дұрыс емес, өйткені x табылған жағдайда, оны қалай детерминирлі өндіру керектігін көрсетпедік.

Алфавит бинарлы деп ұйғарайық, айталық $\{0,1\}$ болсын. Жолдағы префикс реттілік $\leq$-мен белгіленсін; сондықтан егер тек егер $u \leq x$ болса, онда $x = u\upsilon$ орындалатын υ табылады. Жиын

$$\{(x, y, u) \mid \ |x| = |y|, f(x) = y, \text{ және } u \leq x\}$$

P-да жатады, өйткені барлық үш шартты да полиномиалды уақытта детерминирлі тексеруге болады; сондықтан, жиын

$$B = \{(y, u) \mid \exists x \ |x| = |y|, f(x) = y, \text{және } u \leq x\}$$

NP-да жатады. Ұйғарым бойынша *P*= *NP*, *B* жиын *P*-да жатады. Осы фактіні пайдаланып, егер ұзындығы *n* болатын *y* берілсе, онда *f(x) = y* орындалатын *x*-ті табу үшін ұзындығы *n* болатын жолдардан бинарлы іздеу жасай аламыз.

Алдымен $(y, \varepsilon) \in B$ ақиқат па деп сұраймыз. Егер жоқ болса, онда бізге керекті x жоқ; тоқтаймыз және сәтсіздікті хабарлаймыз. Ал егер иә бола қалса, онда $(y, 0) \in B$ ақиқат бола ма деп сұраймыз. Егер иә болса, онда *f(x) = y* орындалатын бірінші биті 0-ге тең x табылады, егер жоқ болса, онда осындай x-тің бәрінің бірінші биті 1-ге тең болады. Енді алдынғы жауапқа тәуелді түрде, жағдайды ескере отырып $(y, 00) \in B$ немесе $(y, 10) \in B$ ақиқат па деп сұраймыз. Жауапты x-тің екінші биті анықтайды. *f(x) = y* орындалатын белгілі бір x-тің барлық биті анықталып біткенше осыны жалғастыра береміз.

Басқа бағыт үшін, φ арқылы *m* айнымалылы логикалық формула белгіленген болсын және ұзындығы *m* болатын биттік жол *t* болсын, ол φ-дің айнымалыларына ақиқаттықты меншіктеуді белгілейді. Функцияны қарастырамыз

$$f(\varphi \# t) = \begin{cases} \varphi \# 1^{|t|}, & \varphi(t) = 1, \\ \varphi \# 0^{|t|}, & \varphi(t) = 0. \end{cases}$$

Енді $\varphi \# t$ формасы жоқ *y* үшін *f(y) = y* болсын. Сонда ұзындығын сақтайтын *f* үшін полиномиалды уақытта есептейміз: алдымен енгізудің формасы $\varphi \# t$ түрінде бола ма соны анықтаймыз және егер осындай болса, онда *t*-да φ-ді бағалаймыз. Егер *f* керіленетін болса, онда *P*= *NP*, өйткені егер тек егер $f(x) = \varphi \# 1^{|t|}$ орындалатын *x* табылса, онда φ орындалады.

32. (a) Егер $A \in NP$ берілсе, онда A-ға қабылдау-ашық бейдетерминирлі n^c-уақытпен шектеулі ТМ машина M болсын. x-тің A-да жата алушылығын келесі түрде айта аламыз:
Енгізу x-те M-нің қабылдау-ашық есептеу тарихын сипаттайтын ұзындығы n^c-дан аспайтын M-нің тізбекті конфигурациясы бар.
Бірінші ретті логикаға сәйкес формализацияланған бұл формуланың қажетті формасы бар.
Және керісінше, егер A-да жату шарты берілген болса

$$\exists y \mid y \mid \ \leq \ \mid x \mid^c \wedge R(x, y) \ ,$$

онда A үшін бейдетерминирлі полиномиалды уақытты машина N -ді құрастыра аламыз, ол енгізу x-те ұзындығы $\mid x \mid^c$ -дан аспайтын y-тің куәгерін табады және $R(x, y)$ -тің ақиқаттығын детерминирлі тексереді, мұндағы R детерминирлі полиномиалды предикат.

41. (a) σ детерминирлі logspace-есептелетін функция болсын. Барлық n үшін $\mid \sigma(x) \mid \ \leq \ \mid x \mid^c$ орындалатын c тұрақты табылады. Полиномиалды өлшемді n^c шығу порты бар, σ-ны есептейтін, polylog-тереңдікті logspace-бірқалыпты B_n схемалар жиынтығын құрастырамыз. Ал σ детерминирлі logspace-те есептелетіндіктен, келесі формуламен анықталатын σ' функция да $\sigma'(x,i) \stackrel{\text{def}}{=}$ i-ші биті $\sigma(x)$ -та жататын детерминирлі logspace-те есептеледі (бинарлы жүйеде өрнектелген i-мен дейік) және σ' -тің логикалық мәндері бар. Барын айтсақ, ол *LOGSPACE* жиынында сипаттама функция болады. Ал $LOGSPACE \subseteq NC$ қатынасы σ' -ты есептейтін *NC* семьясындағы C_m схема; яғни $\mid x \mid \ = n$ үшін $C_m(x,i) = \sigma'(x.i)$ орындалады, мұндағы $m = n + c\log n$. C_m -нің x-тегі алғашқы енгізу n порты және бинарлы жүйеде i үшін соңғылары $c\log n$ болады. Сонда $1 \leq i \leq n^c$ болғандағы $C_m(x,i)$ арқылы $\sigma(x)$ беріледі. C_m -нің n^c қиылыспайтын көшірмесінен B_n -ді құрастырамыз. C_m -нің әрбір көшірмесінің алғашқы n енгізу портына x-ті береміз, содан кейін C_m -нің i-ші көшірмесінің соңғы $c\log n$ енгізу портына бинарлы тұрақты i-ді береміз.
Бұл конструкция logspace бірқалыпты, өйткені C_m үйірі де бірқалыпты және бірнеше көшірме жасау үшін бізге n-ға дейін санай алатын сыртқы цикл қажет болады.

(b) $\mathrm{CVP} \in P$ екендігін біз білеміз, сондықтан егер $P = NP$ болса, онда $\mathrm{CVP} \in NC$ болады. Ал P үшін CVP $\leq_{\mathrm{m}}^{\log}$-толық болатындықтан, кез келген $A \in P$ үшін $A \leq_{\mathrm{m}}^{\log} \mathrm{CVP}$ орындалады. Енді (a)-ға сүйенсек, полиномиалды өлшемді, polylog-тереңдікті logspace-бірқалыпты B_n схемалар үйірі табылып, егер тек егер $B_{|x|}(x) \in \mathrm{CVP}$ болса, онда $x \in A$ болады.
Осы схемалардың шығаруларын CVP -дегі NC схеманың енгізулеріне қойып, A үшін NC схемалар үйірін аламыз.

49. (a) Егер $\mathcal{A} = (A, R, \ldots)$ структурада $k \geq 1$ болғанда жоқ дегенде бір k-арлы өзгешеленетін қатынас R бар болса, онда ол структура бейтривиалды деп аталады. Мұнда белгілі бір k-кортеж $\overline{a} \in A^k$ үшін $R(\overline{a})$ ақиқат болады және белгілі бір k-кортеж $\overline{b} \in A^k$ үшін $R(\overline{b})$ ақиқат болмайды. Еске түсірейік, теңбе-теңдік $=$ қатынас структураның өзгешеленетін қатынасы болмауы қажет еді және мұнда $\overline{a}$ және $\overline{b}$ үшін тұрақты символдар болмауы қажет. $PSPACE$ -күрделікті көрсету үшін, QBF -ті кодтаймыз. Егер QBF формула

$$\mathrm{Q}_1 x_1 \ \cdots \ \mathrm{Q}_n x_n B(\overline{x}),$$

берілген болса, онда әрбір $\mathrm{Q}_i x_i$ кванторды k кванторлармен $\mathrm{Q}_i x_i^1 \ \cdots \ \mathrm{Q}_i x_i^k$ алмастырамыз және $B(\overline{x})$ -дағы әрбір ену x_i -ді $R(x_i^1, \ldots, x_i^k)$ -мен алмастырамыз.
Егер бұған қосымша $\mathcal{A}$ шектеулі болса және функциялар мен структуралар қатынастары кесте арқылы берілсе, онда $\mathcal{A}$ тілінің сөйлемдері келесі $APTIME$ алгоритмінің көмегімен шешілуі мүмкін. Алдымен φ -ді перекис формаға қоямыз. Сонда

$$\mathrm{Q}_1 x_1 \ \cdots \ \mathrm{Q}_n x_n B(\overline{x})$$

аламыз, мұндағы $B(\overline{x})$-те квантор жоқ. $x_1 \in \{0,1,2\}$ -ді экзистенционалды ұйғарамыз, егер $\mathrm{Q}_1 = \exists$ универсал болса, егер $\mathrm{Q}_1 = \forall$ болса және x_2, x_3 үшін осыны қайталаймыз т.с.с. Кванторлармен жұмысты аяқтасымен, $\overline{x}$ -тің ұйғарылған мәндерінде $B(\overline{x})$ -ті кесте арқылы бағалаймыз.

50. Бұл жаттығу үшін 50-аралас жаттығуға берілген көмекті пайдаланып, қатынасты $\equiv_{n,k}^{m}$ анықтаймыз. $\mathcal{A},\overline{a}$ -ның $\equiv_{n+1,k}^{m}$ -эквиваленттілік класын $[\mathcal{A},\overline{a}]_{n+1,k}^{m}$ деп белгілейік, мұндағы $\overline{a}=a_1,\dots\ ,a_k$. Сонда $\equiv_{n,k}^{m}$ -ның тек шектеулі көп класы болады, өйткені ұзындығы m болатын шектеулі көп формула ғана бар. MOVE -тыбылайанықтаймыз:егертекегер $([\mathcal{A},\overline{a}]_{n+1,k}^{m},\alpha)\in$ MOVE болса, онда белгілі бір a_{k+1} үшін $\alpha=[\mathcal{A},\overline{a},a_{k+1}]$ болады. Карастырылған конструкция теорияны шешілетін жасай алмайды, өйткені ол толықтай тиімді емес.

57. (a) Интервалдағы натурал логорифмнің қатаң монотондығына сәйкес, сұрақта берілген теңсіздік үшін барлық $z>1$ үшін $z\ln(1-\frac{1}{z})\le -1$ болатынын көрсетсек жеткілікті болады немесе эквивалентті түрде барлық $0<x<1$ үшін $\ln x\le x-1$ болса жеткілікті. Шынында, бұл теңсіздік барлық нақты оң x -тер үшін орындалады. Қисықтар $y=\ln x$ және $y=x-1$ $(1,0)$ нүктеде бір-біріне жанама, өйткені осы нүктеде екеуінің де көлбеуі 1-ге тең; және $x>0$ кесіндінің бәрінде $y=\ln x$ -тің қатаң кемімелі көлбеуі бар, өйткені оның екінші туындысы теріс, осы мезетте $y=x-1$ түзу болады, сондықтан $y=\ln x$ қисығы $y=x-1$ түзуден төмен орналасады.

Сұрақта берілген шекті мінездеме үшін (осы жаттығудың басқа жағында оның қажеті жоқ), интервалдағы натурал логорифмнің қатаң монотондығына сәйкес, $\lim_{z\to\infty} z\ln(1-\frac{1}{z})=-1$ болатынын көрсетсек жеткілікті болады. Тейлор қатарына $z\ln(1-\frac{1}{z})$ функциясын жіктесек

$$\begin{aligned} z\ln(1-\tfrac{1}{z}) &= z\left(\frac{-1}{z-1}+\frac{1}{2}\left(\frac{-1}{z-1}\right)^2+\frac{1}{3}\left(\frac{-1}{z-1}\right)^3+\cdots\right) \\ &= \frac{-z}{z-1}+z\left(\frac{1}{2}\left(\frac{-1}{z-1}\right)^2-\frac{1}{3}\left(\frac{-1}{z-1}\right)^3+\cdots\right). \end{aligned}$$

$z\to\infty$ болса, онда бірінші мүше -1-ге ұмтылады. Қалған мүшелердің абсолют шамасы келесі шамамен шектелген

$$z\left(\left(\frac{-1}{z-1}\right)^2+\left(\frac{-1}{z-1}\right)^3+\cdots\right) \;=\; \frac{z}{(z-1)(z-2)},$$

және $z\to\infty$ ұмтылғанда, ол 0-ге ұмтылады.

66. (ii) Егер $L\in\Pi^{\log}\cdot\oplus\cdot\mathcal{C}$ болса, онда $A\in\oplus\cdot\mathcal{C}$ және $k\geq 0$ табылып, барлық x үшін

$$x\in L \quad\Leftrightarrow\quad \forall\omega\,|\omega| = k\log|x| \;\;\Rightarrow\;\; x\#\omega\in A$$

орындалады. Одан басқа, $B\in\mathcal{C}$ және $m\geq 0$ табылып, барлық $x\#\omega$ үшін

$$x\#\omega\in A \quad\Leftrightarrow\quad |\{z\,||\,z\,| \;=\; |\,x\#\omega\,|^m \wedge x\#\omega\#z\in AB\}|$$

тақ

$$\Leftrightarrow\quad |W(n^m,B,x\#\omega)| \text{ тақ}$$

орындалады. Қарапайымдылық үшін $|x|$ таңбаның дәрежесі 2-ге тең деп ұйғарамыз. Ұзындығы $k\log|x|$ болатын барлық бинарлы жолдар жиыны $\{\omega_0,\omega_1,\ldots\ ,\omega_{N-1}\}$ болсын, мұндағы $N=|\,x\,|^k$. Және

$$B' \;\stackrel{\text{def}}{=}\; \{x\#z_0z_1\cdots z_{N-1}\,||\,z_i\,| \;=\; |\,x\#\omega_i\,|^m \wedge x\#\omega_i\#z_i\in B,\\ 0\leq i\leq N-1$$

болсын. Сонда $\leq_{\mathrm{T}}^{\mathrm{p}}$-да $\mathcal{C}$ жиыны төменнен жабық болғандықтан, $B'\in\mathcal{C}$ болады. Тағы

$$p(n) \;\stackrel{\text{def}}{=}\; n^k(n+k\log n)^m$$

болсын, сонда

$$W(p,B',x) \;=\; \{z\,||\,z\,| = p(|\,x\,|)\wedge x\#z\in B'\}$$

$$= \prod_{i=0}^{N-1}\{z\,||\,z\,| \;=\; |\,x\#\omega_i\,|^m \wedge x\#\omega_i\#z\in B\}$$

$$= \prod_{i=0}^{N-1} W(n^m, B, x\#\omega_i),$$

мұндағы өрнек

$$UV \stackrel{\text{def}}{=} \{u\upsilon \mid u \in U,\ \upsilon \in V\}.$$

жиынды теориялық конканетация амалына қатысты анықталған көбейтінді. Сонда

$$\begin{aligned} x \in L \quad &\Leftrightarrow \quad \forall i < N\ x\#\omega_i \in A \\ &\Leftrightarrow \quad \forall i < N \mid W(n^m, B, x\#\omega_i) \mid \ \text{тақ} \\ &\Leftrightarrow \quad \Big| \prod_{i=0}^{N-1} W(n^m, B, x\#\omega_i) \Big| \ \text{тақ} \\ &\Leftrightarrow \quad \mid W(p, B', x) \mid \ \text{тақ}, \end{aligned}$$

сондықтан $L \in \oplus \cdot \mathcal{C}$.

(iii) Егер $A \in BP \cdot \mathcal{C}$ болса, онда $B \in \mathcal{C}$ и $m \geq 0$ табылып, барлық y үшін

$$\begin{aligned} y \in A \quad &\Rightarrow \quad \Pr_\omega(y\#\omega \in B) \geq \tfrac{3}{4}, \\ y \notin A \quad &\Rightarrow \quad \Pr_\omega(y\#\omega \in B) \geq \tfrac{3}{4}, \end{aligned}$$

орындалады, мұндағы ω ұзындығы $\mid y \mid^m$ болатын барлық жолдар ішінен кездейсоқ таңдап алынады. Күшейту леммасына сүйеніп (14.1-лемма), қайталанатын тәжірибелердің арқасында қателіктің ықтималдығын экспоненциалды азайып барып жоғалатындай жасауға болады. Жеке жағдайда, B және m

$$\begin{aligned} y \in A \quad &\Rightarrow \quad \Pr_\omega(y\#\omega \notin B) \leq 2^{-(|y|+2)}, \\ y \notin A \quad &\Rightarrow \quad \Pr_\omega(y\#\omega \in B) \leq 2^{-(|y|+2)} \end{aligned} \tag{15}$$

орындалатындай таңдап алынған деп ұйғара аламыз. (14.1-леммада $BPP = BP \cdot P$ болатыны дәлелденіп еді, бірақ бұл дәлелдемені жылдам тексеру мынаны көрсетеді: P-ның жалғыз қажетті қасиеті, ол Тьюрингтің полиномиалды келтірімділігінің тұйықтылығы, оны біз $\mathcal{C}$-ден дәл ұйғарымдаған едік.) Кез келген n үшін қосынды заңы бойынша:

$$\begin{aligned} &\Pr_\omega(\exists y \in \{0,1\}^n \; y\#\omega \in B \Leftrightarrow y \notin A) \\ &\quad\leq \sum_{|y|=n} \Pr_\omega(y\#\omega \in B \Leftrightarrow y \notin A) \\ &\quad\leq 2^n \cdot 2^{-(n+2)} \\ &\quad= \tfrac{1}{4}. \end{aligned} \tag{16}$$

Енді $L \in \oplus \cdot BP \cdot \mathcal{C}$ деп ұйғарайық. Сонда $A \in BP \cdot \mathcal{C}$ және $k \geq 0$ табылып, барлық x үшін

$$x \in L \quad \Leftrightarrow \quad |\{z \,\|\, z| = |x|^k \wedge x\#z \in A\}| \text{ тақ} \tag{17}$$

болады. (15)-ті қанағаттандыратын $B \in \mathcal{C}$ мен $m \geq 0$ таңдағанда, (16)-ға сүйеніп, барлық x үшін

$$\Pr_\omega(\exists z \in \{0,1\}^{|x|^k} \; y\#z\#\omega \in B \Leftrightarrow x\#z \notin A) \;\leq\; \tfrac{1}{4}. \tag{18}$$

болатынын ескереміз. Енді

$$\begin{aligned} B' &\overset{\text{def}}{=} \{x\#\omega\#z \mid x\#z\#\omega \in B\} \\ B'' &\overset{\text{def}}{=} \left\{ \; x\#\omega \,\|\, \{z \,\|\, z| = |x|^k \wedge x\#\omega\#z \in B'\}| \;\text{ тақ} \right\} \\ &\overset{\text{def}}{=} \left\{ \; x\#\omega \,\|\, \{z \,\|\, z| = |x|^k \wedge x\#z\#\omega \in B'\}| \text{ тақ} \right\} \end{aligned}$$

болсын. Сонда $B' \in \mathcal{C}$ және $B'' \in \oplus \cdot \mathcal{C}$. Енді (17)-ні (18)-бен комбинацияласақ,

$$\begin{aligned} x \in L \quad &\Leftrightarrow \quad |\{z \,\|\, z| = |x|^k \wedge x\#z \in A\}| \text{ тақ} \\ &\Rightarrow \quad \Pr_\omega\Big(|\{z \,\|\, z| = |x|^k \wedge x\#z\#\omega \in B\}| \text{ тақ} \Big) \geq \tfrac{3}{4} \\ &\Leftrightarrow \quad \Pr_\omega(x\#\omega \in B''|) \geq \tfrac{3}{4}, \end{aligned}$$

$$x \notin L \quad \Leftrightarrow \quad |\{z \,||\, z| = |x|^k \wedge x\#z \in A\}| \text{ жұп}$$

$$\Rightarrow \quad \Pr_\omega\Big(|\{z \,||\, z| = |x|^k \wedge x\#z\#\omega \in B\}| \text{ тақ}\Big) \leq \tfrac{1}{4}$$

$$\Leftrightarrow \quad \Pr_\omega(x\#\omega \in B''|) \leq \tfrac{1}{4}.$$

Бұл $L \in BP \cdot \oplus \cdot \mathcal{C}$ болатынын дәлелдеп тұр.

67. (d) $f \in \#P$ болсын. Сонда енгізу x-те дәл $f(x)$ қабылдау-ашық есептеу жолы бар, бейдетерминирлі полиномиалды уақытты ТМ M машина табылады. Енгізу x-те дәл $g(x)(f(x))$ қабылдау-ашық есептеу жолды жаңа N машинаны құрастырамыз. N машина алдымен барлық $g(x) \in \mathbb{N}[z]$ коэффициенттерді есептейді. Белгілі бір тұрақты d үшін $g(x)$-тың дәрежесі $|x|^d$-дан аспайды, ал ұйғарым бойынша коэффициенттер полиномиалды уақытта есептелінеді. Осыдан кейін енгізу $g(x)$-те N машина рекурсивті функция R-ді шақырады.

Енгізу $p \in \mathbb{N}[z]$-да рекурсивті функция R төменде айтылған түрде жұмыс жасайды.

• Егер p бірден артық термнен тұрса, онда p-ны $q + r$ түрінде жазамыз, мұндағы q мен r - полиномдар, оның әрқайсысы шамамен p-ның жарты термінен тұрады. Бейдетерминирлі тармақтар құрастырамыз, екі тармақ R-ді рекурсивті q және r -мен сәйкес шақырады.

• Егер p коэффициенті $a \geq 2$ болатын жалғыз терм az^i -ден тұратын болса, онда бейдетерминирлі тармақтар құрастырамыз, екі тармақ R-ді рекурсивті $\lceil a/2 \rceil z^i$ және $\lfloor a/2 \rfloor z^i$ -мен сәйкес шақырады, мұндағы a бинарлы жүйеде полиномиалды көп биттермен өрнектелген.

• Егер p $i \geq 1$ болып жалғыз терм z^i -ден тұратын болса, онда M тармақталғанда, x енгізуде M-ді тармақтала іске қосамыз. M -нің қабылдау-жабыққа апаратын әрбір есептеу жолында, қабылдау-жабық. M -нің қабылдау-ашыққа апаратын әрбір есептеу жолында, z^{i-1} -мен R-ді рекурсивті шақырамыз.

• Егер p жалғыз терм 1-ден тұрса, онда қабылдау-ашық.

Енгізу p-да R-мен генерация жасалатын қабылдау-ашық есептеу жолдарының саны, дәл $p(f(x))$ болатынын индуктивті көрсетуге болады, сондықтан енгізу x-те қабылдау-ашық болатын N-нің есептеу жолдарының саны $g(x)(f(x))$-ке тең болады. N-нің жұмыс уақыты бұрынғыдай полиномиалды, өйткені әрбір қадам полиномиалды уақыт ұстайды және рекурсияның тереңдігі $\log d + c + dn^k$ санымен шектелген, мұндағы d саны $\mathrm{g}(x)$-тың дәрежесі, ал c $\mathrm{g}(x)$-тың кез келген коэффициентін өрнектеуге қажетті бит санына қойылған шектеу және n^k M-нің жұмыс жасауына кететін уақыт.

69. (c) A оракулды Σ_k оракул машина M болсын. Жаттығуда берілген көмекті пайдаланып, басқа Σ_k оракул машина N-мен M-ді симуляция жасаймыз.

Симуляцияның кез келген нүктесінде, симуляция жасайтын машина N, өзінің таспасына M-нің ағымдағы конструкциясы α-ны, осы сәтке дейін M-нің есептеулерде жасаған оракул сұратымдарын $y_1, \ldots, y_m$ және қабылдау-ашық шартын кодтай отырып жалғыз оң енулі логикалық $\psi(x_1, \ldots, x_m, z)$ формуланы жазып отырады. Мұнда айтылған қабылдау-ашық шарт

$$\psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha, A)),$$

мұндағы $A(y_i)$ (қазірше анықталмаған) y_i сұратымға оракулдың берген жауабы және A оракулды α конфигурация басталғанда M-де қабылдау-ашық деген тұжырымды $\mathrm{acc}(\alpha, A)$ білдіреді. Басында N машина $\psi = z$-тен, сұратым тізімі нөл болып, және M-нің бастапқы конфигурациясынан бастайды. Егер α мұрагерлері α_0 мен α_1 болатын $\vee$-конфигурация болса, онда ол әр тармақта бір мұрагерден болып тармақталады. Сұратым тізімі және ψ өзгермейді. Бұл дұрыс, өйткені (b)-ға сүйенсек:

$$\begin{aligned}
&\psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha, A)) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha_0, A) \vee \mathrm{acc}(\alpha_1, A)) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha_0, A)) \\
&\qquad\qquad \vee \psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha_1, A)).
\end{aligned}$$

M машинаның $\wedge$-конфигурациясының симуляция процедурасы осыған ұқсас.

N-нің "иә" мұрагер α_1 және "жоқ" мұрагер α_0 оракул y_{m+1} сұратымы, экзистенционалды тармақталады, егер алдыңғы тармақ экзистенционалды болса және универсалды тармақталады, егер алдыңғы тармақ универсалды болса. Егер әлі ешқандай тармақ болмаса және $k \geq 1$ болса, онда N экзистенционалды тармақталады. (Егер $k = 0$ болса, онда M детерминирлі, мұнда істейтін іс жоқ.) Егер тармақ экзистенционалды болса, онда "иә" таңдайтын мұрагер $\alpha_1, y_1, \ldots, y_{m+1}$ және $\psi(x_1, \ldots, x_m, x_{m+1} \wedge z)$-мен жалғастырады, ал "жоқ" таңдайтын мұрагер $\alpha_0, y_1, \ldots, y_{m+1}$ және $\psi(x_1, \ldots, x_m, x_{m+1} \wedge \neg z)$ -мен жалғастырады. Бұл дұрыс, өйткені (b)-ға сәйкес:

$$
\begin{aligned}
&\psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha, A)) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), A(y_{m+1}) \wedge \mathrm{acc}(\alpha_1, A) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vee(\neg A(y_{m+1}) \wedge \mathrm{acc}(\alpha_0, A))) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), A(y_{m+1}) \wedge \mathrm{acc}(\alpha_1, A)) \\
&\qquad\qquad \vee \psi(A(y_1), \ldots, A(y_m), \neg A(y_{m+1}) \wedge \mathrm{acc}(\alpha_0, A)).
\end{aligned}
$$

Егер тармақ универсалды болса, онда аргумент ұқсас, бірақ айырмашылық бар. Айырмашылық, шарттағы тармақты универсал жасау үшін біз (a)-ны пайдаланамыз:

$$
\begin{aligned}
&\psi(A(y_1), \ldots, A(y_m), \mathrm{acc}(\alpha, A)) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), A(y_{m+1}) \to \mathrm{acc}(\alpha_1, A) \\
&\qquad\qquad\qquad\qquad\qquad\qquad \vee(\neg A(y_{m+1}) \to \mathrm{acc}(\alpha_0, A))) \\
&\quad\equiv \quad \psi(A(y_1), \ldots, A(y_m), A(y_{m+1}) \to \mathrm{acc}(\alpha_1, A)) \\
&\qquad\qquad \vee \psi(A(y_1), \ldots, A(y_m), \neg A(y_{m+1}) \to \mathrm{acc}(\alpha_0, A)).
\end{aligned}
$$

80. φ-ді $\varphi_0 \wedge \varphi_1$ арқылы жазайық, мұндағы φ_1 $\rho(\ell) = 1$ орындалатын литерал ℓ-ды қамтитын барлық клоз φ-ден тұрады және φ-дің қалған клоздары φ_0-ді құрайды. Сонда $\rho(\varphi_1) = 1$ және $\rho(\varphi_1) \leq \varphi_0$, мұндағы $\leq$ амалы X өндіргіштегі еркін логикалық алгебрадағы натурал реттілік. Сондықтан $\rho(\varphi) \leq \varphi_0$.

Теңдік $\rho(\ell) = 1$ орындалатын барлық ℓ литералдар конъюнкциясы K болсын. Сонда K-ның әрбір φ_1 клозбен ортақ литералы бар, сондықтан $K \leq \varphi_1$. Тағы да $N \leq \rho(\varphi_0) \leq \varphi_0$ және қиылыспайтын айнымалылар жиынындағы N мен K конъюнкциялар. Сонда $NK \leq \varphi_0 \varphi_1 = \varphi$ болады, сондықтан φ-де жататын M-нің белгілі бір минтермі үшін

$NK \leq M$ орындалады. Одан басқа, M-ді $N'K'$ формада жазуға болады, мұнда $N \leq N'$ және $K \leq K'$. Сонда $\rho(M) = \rho(N')\rho(K') = N' \leq \rho(\varphi)$; бірақ N минтерм $\rho(\varphi)$ болғандықтан, $N = N'$.

81. $\varphi = x_1, \ldots , x_n$, $\psi = x_2, \ldots , x_n$ және $s = 1$ деп аламыз. Сонда

$$\Pr(\exists M \in \mathrm{m}(\rho(\varphi)) \ |M| \ \geq \ 1) \ = \ 2^{-n}((1+p)^n - (1-p)^n)$$

$$\Pr(\exists M \in \mathrm{m}(\rho(\varphi)) \ |M| \ \geq \ 1 \ | \ \rho(\psi) = 1) \ = \ (1-p)/2.$$

90. Әрбір программа келесі форманың

p; while b do q

біреуіне эквивалент екендігін индукция көмегімен көрсетеміз, мұндағы p мен q циклсыз *while.* Оны келесі түрлендіруді пайдаланып жасауға болады.

(a) Форманы

while b do q; p

келесі формамен алмастырамыз

```
c := b;
while c {
    q;
    c := b;
    if ¬c then p;
}
```

мұндағы c – бұрын p немесе q-да кездеспейтін жаңа логикалық айнымалы. (Іс жүзінде, бізде логикалық айнымалы болмайды, бірақ біз оны симуляция жасай аламыз.)

(b) Форманы

```
while b1 do p1;
while b2 do p2;
   ⋮
while bk do pk;
```

келесі түрмен алмастырамыз:

$i := 1;$

while $i \le k$ $\{$

case i of $\{$

1: if b_1 then p_1 else $i := i + 1;$

2: if b_2 then p_2 else $i := i + 1;$

$\vdots$

k: if b_k then p_k else $i := i + 1;$

$\}$

$\}$

мұндағы i – жаңа бүтін санды айнымалы және k – тұрақты. Біз if - then - else арқылы жазылған case тұжырымды симуляция жасай аламыз.

(c) Берілген форманы

if b

then p_1; while c_1 do q_1;

else p_2; while c_2 do q_2;

келесі формамен алмастырамыз

$c := b;$

if c then p_1 else p_2;

while $(c \wedge c_1) \vee (\neg c \wedge c_2)$ $\{$

if c then q_1 else q_2

$\}$

мұндағы c жаңа логикалық айнымалы.

(d) Форманы

```
while b {
   p;
   while c do q
}
```

формамен

```
if b {
 p;
 while b ∨ c {
    if c then q else p
 }
}
```

алмастырамыз және (c)-ны қолданамыз.

91. $f \in \mathcal{P}$ анықтамасынан индукцияны пайдаланып, алдымен $P \subseteq \mathcal{C}$ болатынын көрсетеміз. Примитивті рекурсиямен f анықталатын жағдайдан басқа жағдайдың барлығы қарапайым. Бұрын анықталған:

$$\begin{aligned} f(0,\overline{x}) &= h(\overline{x}) \\ f(s(y),\overline{x}) &= f(y,\overline{x},f(y,\overline{x})). \end{aligned}$$

$h:\omega^m \to \omega^n$ және $g:\omega^{m+n+1} \to \omega^n$ қатынастардан $f:\omega^{m+1} \to \omega^n$ бейне примитивті рекурсия арқылы анықталады деп ұйғарайық. Біз $f \in \mathcal{C}$ болатынын көрсеткіміз келеді. Индукциялық гипотеза бойынша h пен g $\mathcal{C}$-де жатады. Анықтаймыз

$$\begin{aligned} \hat{g}(y,\overline{x},\overline{z}) &= (\mathrm{s}(y),\overline{x},g(y,\overline{x},\overline{z})) \\ \hat{f}(n,y,\overline{x},\overline{z}) &= \hat{g}^{\,n}(y,\overline{x},\overline{z}). \end{aligned}$$

$\hat{f}$ және $\hat{g}$ екеуі де $\mathscr{C}$-де жатады. Аргумент, n арқылы жүргізілген индукция көмегімен аламыз,

$$\begin{aligned}
\overline{g}^{0}(y,\overline{x},\overline{z}) &= (0,\overline{x},h(\overline{x})) \\
&= (0,\overline{x},f(0,\overline{x})) \\
\hat{g}^{s(n)}(0,\overline{x},h(\overline{x})) &= \hat{g}(\hat{g}^{n}(0,\overline{x},h(\overline{x}))) \\
&= \hat{g}(n,\overline{x},f(n,\overline{x})) \text{ (индукциялы гипотеза)} \\
&= (s(n),\overline{x},g(n,\overline{x},f(n,\overline{x}))) \\
&= (s(n),\overline{x},f(s(n),\overline{x})).
\end{aligned}$$

Проекция, кортежге түрлендіру және композицияны пайдаланып, f -ті $\hat{g}$ -ден проекция көмегімен етіп

$$\begin{aligned}
e(n,\overline{x}) &= \hat{f}(n,z(n),\overline{x},h(\overline{x})) \\
&= \hat{f}(n,0,\overline{x},h(\overline{x})) \\
&= \hat{g}^{n}(0,\overline{x},h(\overline{x})) \\
&= (n,\overline{x},f(n,\overline{x})),
\end{aligned}$$

анықтаймыз˙

Енді кері жағдайды, яғни $f \in \mathcal{P}$ анықтамасынан индукцияны пайдаланып $\mathscr{C} \subseteq P$ болатынын көрсетеміз. Тағы да, g -дан n-ретті композицияны пайдаланып f-ті алғаннан басқа жағдайлардың барлығы қарапайым, есептің қойылымында берілгені:

$$f(n,\overline{y}) \quad = \quad \mathrm{g}^{n}(\overline{y}).$$

Индукция гипотезасы бойынша $\mathrm{g} \in \mathcal{P}$. Сондықтан

$$\hat{g}(x,\overline{y},\overline{z}) \quad = \quad \mathrm{g}(\overline{z}),$$

және $\hat{g}$ -ден f примитивті рекурсия көмегімен анықталады:

$$\begin{aligned} f(0,\overline{y}) &= \overline{y} \\ &= \mathrm{g}^{0}(\overline{y}) \\ f(s(n),\overline{y}) &= \hat{g}(n,\overline{y},f(n,\overline{y})) \\ &= \hat{g}(n,\overline{y},g^{n}(\overline{y})) \\ &= \mathrm{g}(g^{n}(\overline{y})) \\ &= \mathrm{g}^{s(n)}(\overline{y})). \end{aligned}$$

93. Осында p - *for* программа болсын. Және p-да кездеспейтін жаңа айнымалы t болсын. Әрбір тұжырымның алдына $t := t+1$ -ді қоямыз және программаның басына $t := 0$-ді қоямыз. Нәтижесінде алынған p' программа *for* программа болады, мұнда енгізу x-те бастапқы p программада жасалған қадамдар санына x енгізуде p' программадағы t-ның соңғы мәні тең болады. Бұл функция сөзсіз, примитивті рекурсиялы болады, өйткені ол p' программада *for* көмегімен есептеледі.

Керісінше, уақытқа примитивті рекурсивті шектеу қойылған *while* программа p болсын. Сонда x енгізуінде p-ның есептеу уақытын санайтын және оны t айнымалысында қалдыратын, *for* программа q табылады. Жалпылықты сақтай отырып, x-тен басқа q-де p-мен ортақ айнымалы жоқ және ол x-тің мәнін өзгертпейді деп ұйғарамыз. Ал енгізу x-те кез келген *while* цикл p программада t реттен артық орындалмайтындықтан, бастапқы p программа *for* программа $q; p'$-ке эквивалентті, мұндағы p' программа p-дан әрбір *while* циклды

```
while b do r
```

for циклге

```
for t {
   if b then r
}
```

алмастыру арқылы алынады.

98. Бұл сөзсіз шексіздік, өйткені шексіз көп әртүрлі дербес рекурсивті функция бар. Ол тізбелегіш M машинамен тізбеленетін шексіз р.с. ішкі жиын қамтиды деп ұйғарайық. M-мен тізбеленген $f(i)$-ді i-ден үлкен бірінші элемент ретінде анықтаймыз. Сонда f қозғалыссыз нүктесі жоқ жалпы рекурсиялы функция болып шығады, ал ол рекурсия жайлы теоремаға қайшы келеді.

107. Алға-артқа аргументін қолданып, $h:\omega \to \omega$ үшін шектеулі аппроксимациялар тізбегін құрастырамыз. Толық анықталмаған h-тан бастаймыз. Енді біз n этаптан кейін h бірдің-бірге қатынасы және $\text{graph } h \subseteq R$ болатын, шектеулі анықталу аймағы белгілі h бейнені құрастырдық деп ұйғарайық. Егер n жұп болса, онда h-тың анықталу аймағында жатпайтын x ең кіші элемент болсын. Келесі програманы іске қосамыз:

$$
\begin{aligned}
& y := f(x) \\
& \text{while } (y \in \mathrm{range}(h)) \ \{ \\
& \quad z := h^{-1}(y); \\
& \quad y := f(z); \\
& \} \\
& h(x) := y;
\end{aligned}
$$

Алғашқы меншіктеуден кейін $(x, y) \in R$ болады. Ал $\text{graph} f \subseteq R$, $\text{graph } h \subseteq R$ және $R \circ R^{-1} \circ R \subseteq R$ болғандықтан, $(x, y) \in R$-де цикл инвариантты сақтайды. Сондықтан, егер цикл тоқтаса, онда y-тің соңғы мәні h-тың мәндер жиынында жатпайды, сондықтан инварианттылықты сақтай отырып h-тың анықталу аймағын бір элементке ұлғайттық, сонда өзінің анықталу аймағында h бірдің-бірге қатынасы және $\text{graph } h \subseteq R$.

Цикл тоқталысымен аргументтеу үшін цикл жұмыс үстінде болғанда әйтеуір бір кезде y пен z айнымалылары қабылдайтын мәндер сәйкес Y және Z элементтер жыйынын құрасын делік. Сонда $Y = f(\{x\} \cup Z)$ және $Z = h^{-1}(Y \cap \mathrm{range}\ h)$. Егер цикл еш уақытта тоқтамаса, онда $Y \subseteq \mathrm{range}\ h$ болады, сондықтан $Z = h^{-1}(Y)$ және $Y = h(Z)$. Ал f пен h бірдің-бірге қатынасы және h-тың анықталу

аймағы шектеулі болғандықтан, $|Z| = |Y| = |\{x\} \cup Z|$ болады, соның әсерінен $x \in Z = h^{-1}(Y)$. Бірақ бұл h-тың анықталу аймағында x жатпайды деген ұйғарымға қайшы келеді.
Егер n жұп болса, онда біз h-тың анықталу аймағында жатпайтын x-ті ең кіші элемент есебінде аламыз және басқа бағытта жұмыс жасаймыз, яғни жоғарыда келтірілген аргументте f-ті g-ге және R-ді R^{-1}-ге алмастырамыз. Бізге $R^{-1} \circ R \circ R^{-1} \subseteq R^{-1}$ қасиет қажет. Осы жаттығуға берілген көмекте көрсетілгендей, $R \circ R^{-1} \circ R \subseteq R$ ұйғарымнан элементар жиынды теориялық талдау көмегімен ол қасиет алынады.

110. (a) Полиномиалды уақыт сағатымен жабдықталған жалпы TM тізбесі $M_0, M_1, \ldots$ болсын, олар машинаны n^k қадамнан кейін тоқтатады. Тізбе әрбір k үшін n^k сағатты әрбір TM-ның көшірмесін сақтайды. Сонда

- әрбір M_i полиномиалды уақытта жұмыс жасайды, және
- P-дағы әр жиынға белгілі бір M_i-де қабылдау-ашық.

A және $\sim A$-ны тізбелейтін процедура алдарыңызда. Басында бос болған индекстердің шектеулі тізімін сақтап отырамыз. Тізімдегі әрбір индекс не белгіленген немесе белгіленбеген. Стадия x-те, x-ті тізімге белгіленбеген ретінде орналастырамыз. Осыдан кейін тізімдегі енгізу x-тегі әр машинаны симуляция жасаймыз және M_i-де x-ке қабылдау-ашық болатындай етіп, тізімнен ең кіші i-ді таңдап аламыз. Егер ондай M_i жоқ болса, онда $x \in A$-ға орналастырамыз және $x+1$-ші стадияға ауысамыз. Егер i белгіленбеген болса, онда $x \in A$-ға орналастырамыз және i-ді белгілейміз. Егер i бұрын белгіленген болса, онда $x \in \sim A$ орналастырамыз және i-ді тізімнен сызып тастаймыз. Соңында әрбір M_i тізімде орналасқан болып шығады. Егер M_i тізімде болып және x-ке қабылдау-ашық болса, онда M_i-дің x стадияда белгілеу немесе жойылу үшін таңдап алынбауының жалғыз жолы бар, ол мынада: егер тізімде x стадияда белгіленетін немесе жойылатын төмен индексті машина табылса және ол $2i$ реттен аспайтындай орындалса. Сондықтан егер M_i-де шексіз жиынға қабылдау-ашық болса, онда соңғы нәтиже бойынша ол тізімдегі приоритеті ең үлкен машина болады және соңында ол белгіленеді және жойылады.

M_i белгіленген кезде, $L(M_i) \not\subseteq \sim A$ болатынына кепілдік бере отырып, ағымдағы $x \in A$ орналастырамыз. Ал M_i жойылған кезде, $L(M_i) \not\subseteq A$ болатынына кепілдік бере отырып, ағымдағы $x \in \sim A$ орналастырамыз.
A және $\sim A$ екеуі де р.с., өйткені жоғарыда баяндалған конструкция оларды тізімдейді, сондықтан A рекурсивті. A және $\sim A$ екеуі де шексіз, өйткені шексіз көп машина белгіленеді және жойылады.

126. Кез келген S үшін,

$$DSPACE(S_0) \cup DSPACE(S_1) \neq DSPACE(S)$$

орындалатындай жалпы рекурсивті S_0 және S_1 шекараны анықтаймыз. Енді $L(M)$ -ге қабылдау-ашық әрбір ТМ, барлық дерлік енгізуде таспаның жоқ дегенде бір ұяшығын пайдаланатын, жалпы ТМ M болсын (117-ші аралас жаттығуды қараңыз). Ұзындығы n енгізулерде M -нің максималды қолданатын кеңістігі $T(n)$ болсын және анықтаймыз

$$S_0(n) \stackrel{\text{def}}{=} \begin{cases} T(n), & \text{if n even,} \\ 0, & \text{if n odd} \end{cases}$$

$$S_1(n) \stackrel{\text{def}}{=} \begin{cases} T(n), & \text{if n odd,} \\ 0, & \text{if n even.} \end{cases}$$

Сонда

$$A_0 \stackrel{\text{def}}{=} L(M) \cap \{x \,||\, x| \text{ жұп } \} \in DSPACE(S_0)$$

$$A_1 \stackrel{\text{def}}{=} L(M) \cap \{x \,||\, x| \text{ тақ } \} \in DSPACE(S_1).$$

Егер $DSPACE(S_0) \cup DSPACE(S_1) \subseteq DSPACE(S)$ болса, онда A_0 және A_1-дің екеуі де $DSPACE(S)$-те жатады, сондықтан олардың бірігуі $L(M)$-де сонда жатады. Дегенмен $L(M) \notin DSPACE(S_0)$ және $L(M) \notin DSPACE(S_1)$, өйткені

$L(M)$ үшін барлық машина таспаның, жоқ дегенде бір ұяшығын б.ж.д. қажетсінеді.

131. (b) $\varphi_i(x) \downarrow^t = y$ арқылы "φ_i-ді есептейтін машина t немесе одан да аз қадамда енгізу x-те тоқтайды және y-ті шығарады" деген рекурсивті предикаты белгіленсін. Жиын EQUAL Π_2^0-де жатады, өйткені оны келесі түрде өрнектеуге болады:

$$\begin{aligned}\text{EQUAL} \;=\; \{(i,j) \mid \forall x\ \forall t\ \forall y\ \exists s\ &(\varphi_i(x)\downarrow^t = y \Rightarrow \varphi_j(x)\downarrow^s = y)\\ &\wedge(\varphi_j(x)\downarrow^t = y \Rightarrow \varphi_i(x)\downarrow^s = y)\}.\end{aligned}$$

EQUAL-дың Π_2^0-күрделі болатынын дәлелдеу үшін, (a) бөліктегі ALL-ды соған келтіреміз. (Сіз (a) бөлікті орындаған шығарсыз?) Келесі

$$\varphi_{\tau(i)}(x) \;=\; \begin{cases} 0, & \text{if } \mathrm{M}_i \text{ accepts } \mathrm{x} \\ \text{undefined}, & \text{otherwise} \end{cases}$$

дербес рекурсиялы функцияның индексі $\tau(i)$ болсын. i-ден индекс $\tau(i)$ тиімді түрде алынуы мүмкін және $\varphi_{\tau(i)}$ анықталу аймағы $L(M_i)$ дәлдігінде болады. Енді анықтаймыз

$$\sigma(i) \;=\; (\tau(i),\ \mathsf{const}(0)),$$

мұндағы $\mathsf{const}(0)$ тұрақты функцияның ё$x.0$ индексі. Сонда

$$i \in ALL \;\Leftrightarrow\; \varphi_{\tau(i)} = \lambda x.0 \;\Leftrightarrow\; \sigma(i) \in \text{EQUAL}.$$

134. (a) Иә. Алдымен

$$A \;\stackrel{\text{def}}{=}\; \{M \mid M \text{ полиномиалды уақыт жұмыс жасайды}\}$$

$$B \;\stackrel{\text{def}}{=}\; \{M \mid \text{“}M \text{ полиномиалды уақыт жұмыс жасайды”} \text{ предикаты мүмкін PA-да шығар}\}$$

Сөзсіз $B \subseteq A$, өйткені PA қатаң. $B \in \Sigma_1^0$ және A Σ_2^0-толық болатынын көрсетеміз, сондықтан екі жиын тең болуы мүмкін емес.

$B \in \Sigma_1^0$ болатыны сөзсіз, өйткені егер тек егер M полиномиалды уақытта жұмыс жасаса, онда $\exists k\ \forall x\ M(x)\downarrow^{|x|^k}$ болады, ал соңғы сөйлем Σ_2^0 предикат. Қосымша, Σ_2^0 үшін A күрделі болады, өйткені біз $\mathrm{FIN} = \{M \mid L(M)$ шектеулі$\}$-ні соған келтіре аламыз, ал оның Σ_2^0-күрделі болатынын білеміз (35-лекция). M машина берілген, біз M' машинасын құрастырғымыз келеді, ол егер тек егер полиномиалды уақытта жұмыс жасаса, онда $L(M)$ шектеулі болады. Енгізу x-те M' келесі амалдарды жасайды делік.
(i) $|x|$ қадамда барлық енгізулер y үшін $|y| < \log|x|$ болатындай етіп, M-ді симуляция жасайды.
(ii) Егер осының c симуляциясы тоқтаса және қабылдау-ашық болса, онда тағы да $|x|^c$ қадамға жұмыс жасалады және тоқтайды. Қадам (i) симуляцияның $|x|^2$ қадамын керек етеді. Қадам (ii) симуляцияның $|x|^c$ қадамын керек етеді. Енді $L(M)$ шектеулі, айталық $|L(M)| = c$ болса, онда M' n^c уақыт жұмыс жасайды. Егер шектеулі болмаса, онда c-ға жоғарыдан шектеу жоқ, және M' полиномиалды уақыт жұмыс жасамайды.
(b) Жоқ. A бөлік (a)-дағы сияқты болсын. Біз егер $M \in A$ болса, онда M $n^{f(M)}$ уақыт жұмыс жасайтын, барлық A-да анықталған дербес рекурсивті функция табыла ма деп сұраймыз. Сондай f бар деп ұйғарайық. Енді

$$C \overset{\text{def}}{=} \left\{M \mid \exists t\, f(M)\downarrow^t \wedge \forall x\ \forall t\ \left(f(M)\downarrow^t \Rightarrow M(x)\downarrow^{|x|^{f(M)}}\right)\right\}$$

болсын. Сонда $C \subseteq A$ болады және $A \subseteq C$ ақиқат бола ма деп сұраймыз. (a) бөлікте көрсетілгендей, A Σ_2^0- толық болады, бірақ Σ_1^0 жиын мен Π_1^0 жиынның қиылысуы C болады, соның әсерінен ол Δ_2^0-де жатады, сондықтан ол Σ_2^0-толық бола алмайды.

136. Мүмкін формализацияның тағы біреуін келтірейік. Сандар теориясының кез келген дедуктивті жүйесі F болсын (мысалы, Пеано арифметикасы). Жиын $\{x \mid \varphi(x)\}$ рекурсивті жиын болатын, бірақ кез келген ТМ M-де $F \vdash \forall x\ M(x)\downarrow$ орындалса да $L(M)$ болмайтын, сандар теориясында $\varphi(x)$ формуласы табылады.
Осыны дәлелдеу үшін N машина ТМ болсын делік, ол

$\forall x\ M(x)\downarrow$ формадағы сөйлемнің дәлелдемесін F-тен тізбелейді (яғни, M жалпы болады). Осындай теорема тізбеленетін барлық жағдайда, x стадиясында деп айталық, x-те N M-ді іске қосады және егер тек егер x-ті тізбелесе, онда M-де x-ке қабылдау-жабық. F қатаң деп ұйғарсақ, бұл шешіледі, өйткені M шынымен жалпы. Енді $x \in L(N)$-де $\varphi(x)$ формула болсын. Қатынас $x \in L(N)$ ақиқат бола ма соны анықтау үшін шектеулі көп қадамда тек N-ді іске қоссақ жеткілікті, сонда жиын $\{x \mid \varphi(x)\} = L(N)$ рекурсивті болады. Бірақ ол кез келген $F \vdash \forall x\ M(x)\downarrow$ орындалатын M үшін, ол $L(M)$ бола алмайды, өйткені осындай барлық машинадан қашықта диогоналдадық.

139. Проблема (a) Σ_2^0-толық болады және проблема (b) Π_1^1-толық болады. Енді (a)-ның Σ_2^0-де екендігін көрсету үшін жалпылықты жоғалтпай алдымен M ешқашан тоқтамайды деп ұйғарамыз (q-ды қамтымайтын шексіз циклге кіретіндей етіп, тоқтаудың күйлерін өзгертеміз; оның M (a)-ны қанағаттандыра ма соған әсері жоқ). Егер тек егер (a) орындалса, онда
(A) түбірден басталатын шектеулі π есептеу траекториясы табылып, ол траекторияда барлық $n > |\pi|$ үшін, π-дың ұзындығы n болатын ρ_n кеңейтілуі бар болып, префикс π-дің сыртында ρ_n енуді қамтымайды.
Егер осы ұйғарымға сенсеңіз, онда проблема Σ_2^0-де жатады, өйткені рекурсивті предикаттан алынатын $\exists\pi\forall n$ форма түрінде (A) өрнектелуі мүмкін.
Ұйғарымды дәлелдеу үшін, алдымен (A)-ны (a) жеңіл меңзейтінін еске түсірейік: егер (a)-ны қанағаттандыратын шексіз траектория σ берілсе, онда q-дың барлық енуін сақтайтын шектеулі префикс π болсын. Сонда барлық $n > |\pi|$ үшін (A)-ның шарттарын қанағаттандыратын ұзындығы n болатын σ-ның префиксі, π-дің кеңейтілуі ρ_n болады.
Басқа бағыт Кёниг леммасын қажетсінеді. (A) орындалады деп ұйғарайық. Әрбір $n > |\pi|$ үшін (A)-ның көмегімен берілген, π және ρ_n-нің барлық кеңейтілуінен тұратын ішкі бұтақты қарастырамыз. Бұл шексіз шектеулі салмақты бұтақ, сондықтан Кёниг леммасы бойынша шексіз траекторияны қамтиды, ол траектория (a)-ның шарттарын қанағаттандыруы қажет.

Енді (a)-ның Σ_2^0-күрделі екендігін көрсету үшін біз FIN-ді соған келтіреміз. Егер M машина берілсе, онда күйі q болатын N бейдетерминирлі машинаны құрастырғымыз келеді, ол N машинада егер тек егер тек оның шектеулі көп енулер q-да есептеу траекториясы бар болса, онда M-де шектеулі жиынға қабылдау-ашық. Жалпылықты сақтай отырып, M-де ешқашан қабылдау-жабық болмайды деп ұйғарайық (қабылдау-жабық күйді өзгертіп, шексіз циклге кіретін жасаймыз). Алдымен N машина n-ді бейдетерминирлі табатын болсын. Бұл амалды ол былай атқарады: бірнеше мәрте санақшыға бірді қосатын циклге кіреді, осыдан кейін циклден шығу керек пе немесе ары қарай жалғастыра бере ме соны бейдетерминирлі таңдайды. Осы циклдің бірінші күйі q болсын. Табылған сан n-мен циклден шығатын әрбір бұтақ үшін уақытты бөлшектеу стилінде ұзындығы n-нен үлкен барлық енгізулер үшін M-ді симуляция жасаймыз. Бұл симуляцияларда ешуақытта күй q-ға ене алмаймыз. Егер осы симуляцияның кез келгенінде қабылдау-ашық болса, онда таспаны өшіреміз және барлық есептеуді басынан қайта бастаймыз.

Енді, егер M-де шектеулі жиынға қабылдау-ашық болса, онда ұзындығы n-нен үлкен жолға M-де қабылдау-ашық болмайтын n табылады және табылған n-ге сәйкес N-нің есептеу траекториясында (немесе кез келген үлкен сан үшін) q ешқашан қайтадан пайда болмайды, өйткені M-ді симуляция жасаймын деп N мәңгілік тығырыққа тіреледі. Басқа жағынан, егер M-де шексіз жиынға қабылдау-ашық болса, онда табылған n санына сәйкес N-нің әрбір есептеу траекториясы, ұзындығы n-нен асатын $x \in L(M)$-ты табады және есептеуді басынан қайта бастайды, сондықтан ол қайтадан күй q-ге кіреді. N-ге сәйкес шексіз есептеу траекториясы әрқашан циклде қалады және n-ді табу ешқандай қиындық тудырмайды, өйткені ол күй q-ге шексіз рет түседі.

Енді (b)-ның Π_1^1-де жататынын көрсету үшін шарт (b)-ны келесі түрде

$$\forall \pi \; \exists n \; \forall m \geq n \; \mathrm{state}(\pi(m)) \neq q \, ,$$

өрнектеуге болатынын байқаймыз. Бұл өрнек екінші ретті универсал квантордан тұрады: ол есептеу бұтағында бірінші ретті предикатпен алынатын π траекторияны (мүмкін шексіз) көктей өтеді, және π предикат тек шектеулі көп ену q-ды қамтитынын айтады.

Ал (b)-ның Π_1^1-күрделілік екендігін көрсету үшін, әділетті аяқталу проблемасын оған келтіреміз. Бинарлы-салмақталатын бейдетерминирлі машина M үшін 41-лекциядағы әділеттілік шартты (true,last(0)) қарастырамыз. Сондықтан егер тек егер шексіз траектория әділетті болса, онда ол шексіз көп сол тармақты қамтиды. M машина сол тармақты алған әрбір кезде, күйге бірден енетін етіп M машинаны модификациялаймыз. Сонда егер тек егер модификацияланған машина әділетті анықталса, онда шексіз көп әділетті жол жоқ, егер тек егер әрбір траектория тек шектеулі көп ену q-ды қамтыса.

140. (b) Бұл уәде проблемасына мысал. Бізге проблемалар үлгісі ғана көрсетіледі деген уәде беріліп еді, олар белгілі бір қасиетке ие болады - біздің жағдайда ол берілген жалпы машина - оның шешілмеуі де мүмкін. Дегенмен, біз оны шешуге міндетті емеспіз; ол бізге көрсетілген енгізулер үшін әрқашан орындалады деп санауға болады.

Формалды түрде, уәде проблемасы $(A,P)\in\Sigma^*\times\Sigma^*$ жұбы болып табылады, мұнда A шешілетін проблема және P уәде. (A,P) уәде проблемасы күрделілік класы $\mathcal{C}$-да жатады, егер P-ның енгізулері үшін ол осы шектеулі ресурстармен жұмыс жасайтын болса және $A\cap P$-ның жолдарына қабылдау-ашық болатын машина табылса, онда ол машина ресурсына деген белгілі бір шектеумен анықталады. A жолдарына деген қабылдау-ашық немесе P-да жатпайтын енгізу жолдарына деген шектеуді сақтау, бұл жерде қарастырылмайды. Мысалы, егер P-ның барлық элементінде және сол енгізулерде тоқтайтын, $A\cap P$ -ден дәлдікпен элемент қабылдайтын Тьюринг машинасы табылса, онда (A,P) уәде проблемасы шешіледі. (Рекурсивті жиын B табылып, $A\cap P=B\cap P$ орындалады деп айтылған тұжырым мен бұл бірдей емес, соны естеріңізге саламыз.) Егер әрбір $\mathcal{C}$-дағы B A-ға жалпы рекурсиялы функция арқылы $\leq_m$-келтірілсе, ал ол уәдені орындаса; яғни барлық x үшін $\sigma(x)\in P$ болса, онда $\mathcal{C}$ үшін (A,P) уәде проблемасы $\leq_m$-күрделі болады.

Біздің жағдайда уәде проблемасы

$$\begin{aligned} A &= \{M \mid M\text{-де транзитивті бинарлы қатынасқа қабылдау-ашық}\} \\ &= \{M \mid \forall x\ \exists y\ \forall z\ (x,y) \in L(M) \wedge (y,z) \in L(M) \\ &\qquad \rightarrow (x,z) \in L(M)\}, \\ P &= \{M \mid M \text{ жалпы }\} \end{aligned}$$

тұрады. Бұл Π_1^0-де жатады, өйткені тек $\forall$-тармақты IND программа бар, ол P-дегі енгізулер үшін әрқашан тоқтайды және $A \cap P$ дәлдікпен қабылдау-ашық.

Π_1^0-күрделілікті көрсету үшін біз тоқтаудың комплемент проблемасын A-ға келтіре аламыз. Біз берілген $N\#x$-тен бинарлық қатынасқа қабылдау-ашық жалпы M машинаны тиімді құрастыруымыз қажет, ол егер тек егер транзитивті болса, онда x-те N тоқтамайды. Егер тек егер енгізу (s,t)-ға M-де қабылдау ашық болса, онда не (i) $s \neq t$, немесе (ii) $s = t$ және t қадамда x-те N тоқтамайды орындалсын. Сөз жоқ, M-ді жалпы жасауға болады. Егер x-те N тоқтамаса, онда $L(M)$ барлық жұпты қамтиды, сондықтан ол транзитивті қатынас болады. Егер t қадамда x-те N тоқтаса, онда $(t,t+1),(t+1,t) \in L(M)$ болады, дегенмен $(t,t) \notin L(M)$, сондықтан $L(M)$ транзитивті емес.

Қолданылған әдебиеттер

[1] L. Adleman and M. Huang, Recognizing primes in random polynomial time, in *Proc. 19th Symp. Theory of Computing*, New York: ACM, 1987, pp. 462-469.
[2] M. Agrawal and S. Biswas, Primality and identity testing via Chinese remaindering, in *Proc. 40th Symp. Foundations of Computer Science*, Los Alamitos, CA: IEEE, 1999, pp. 202-208.
[3] M. Agrawal, N. Kayal, and N. Saxena, PRIMES is in P, *Ann.Math.*, 160 (2004), pp. 781-793.
[4] A.V. Aho, J.E. Hopcroft, and J.D. Ullman, *The Design and Analysis of Computer Algorithms*, Reading, MA: Addison-Wesley, 1975.
[5] M. Ajtai, Σ_1^1 formulae on finite structures, *Ann. Pure Appl. Logic*, 24 (1983), pp. 1-48.
[6] S. Arora, C. Lund, R. Motwani, M. Sudan, and M. Szegedy, Proof verication and hardness of approximation problems, *J. Assoc. Comput. Mach.*, 45 (1998), pp. 501-555.
[7] S. Arora and S. Safra, Probabilistic checking of proofs: A new characterization of *NP*, *J. Assoc. Comput. Mach.*, 45 (1998), pp. 780-112.
[8] L. Babai, Trading group theory for randomness, in *Proc. 17th Symp. Theory of Computing*, New York: ACM, April 1985, pp. 421-429.
[9] L. Babai, L. Fortnow, and C. Lund, Nondeterministic exponential time has two-prover interactive protocols, *Comput. Complex.*, 1 (1991), pp. 3-40.
[10] T. Baker, J. Gill, and R. Solovay, Relativizations of the $P \overset{?}{=} \mathrm{NP}$ question, *SIAM J. Comput.*, 4 (1975), pp. 431-442.
[11] M. Bellare, D. Coppersmith, J. H˚astad, M. Kiwi, and M. Sudan,

Linearity testing in characteristic two, *IEEE Trans. Inf. Theory*, 42 (1996), pp. 1781-1795.

[12] C. Bennett and R. Gill, Relative to a random oracle $P \neq NP \neq$ co-*NP* with probability 1, *SIAM J. Comput.*, 10 (1981), pp. 96-103.

[13] E.R. Berlekamp, Factoring polynomials over large finite fields, *Math. Comp.*, 24 (1970), pp. 713-735.

[14] L. Berman, The complexity of logical theories, *Theor. Comput. Sci.*, 11 (1980), pp. 71-77.

[15] P. Berman, Relationship between the density and deterministic complexity of *NP*-complete languages, in *Proc. 5th Int. Colloq. Automata, Languages, and Programming*, vol. 62 of *Lect. Notes in Comput. Sci.*, New York: Springer-Verlag, 1978, pp. 63-71.

[16] M. Blum, A machine-independent theory of the complexity of recursive functions, *J. Assoc. Comput. Mach.*, 14 (1967), pp. 322-336.

[17] ______ , On effective procedures for speeding up algorithms, *J. Assoc. Comput. Mach.*, 18 (1971), pp. 290–305.

[18] M. Blum and D. Kozen, On the power of the compass, in *Proc. 19th Symp. Found. Comput. Sci.*, IEEE, October 1978, pp. 132-142.

[19] M. Blum, M. Luby, and R. Rubinfeld, Self-testing/correcting with applications to numerical problems, *J. Comput. Syst. Sci.*, 47 (1993), pp. 549-595.

[20] R.B. Boppana and M. Sipser, The complexity of finite functions, in *Handbook of Theoretical Computer Science (vol. A): Algorithms and Complexity*, Cambridge, MA: MIT Press, 1990, pp. 757-804.

[21] A.B. Borodin, Computational complexity and the existence of complexity gaps, *J. Assoc. Comput. Mach.*, 19 (1972), pp. 158-174.

[22] _______, On relating time and space to size and depth, *SIAM J. Comput.*, 6 (1977), pp. 733–744.

[23] J.R. Büchi, Weak second order arithmetic and finite automata, *Zeitschrift für Math. Logik und Grundlagen Math.*, 6 (1960), pp. 66-92.

[24] _______, On a decision method in restricted second order arithmetics, in *Proc. Int. Congr. Logic, Methodology and Philosophy of Science*, Stanford, CA: Stanford University Press, 1962, pp. 1–12.

[25] J.Y. Cai and M. Ogihara, Sparse hard sets, in *Complexity Theory Retrospective*, L.A. Hemaspaandra and A.L. Selman, eds., New York: Springer-Verlag, 1997, pp. 53-80.

[26] A. Chandra, D. Kozen, and L. Stockmeyer, Alternation, *J. Assoc.*

Comput. Mach., 28 (1981), pp. 114–133.
[27] R. Chang, B. Chor, O. Goldreich, J. Hartmanis, J. Håstad, D. Ranjan, and P. Rohatgi, The random oracle hypothesis is false, *J. Comput. Syst. Sci.*, 49 (1994), pp. 24–39.
[28] A. Church, A set of postulates for the foundation of logic, *Ann. Math.*, 33-34 (1933), pp. 346-366, 839-864.
[29] _______, An unsolvable problem of elementary number theory, *Amer. J. Math.*, 58 (1936), pp. 345–363.
[30] A. Cobham, The intrinsic computational difficulty of functions, in *Proc. 1964 Cong. for Logic, Methodology and Philosophy of Science II*, Y. Bar-Hillel, ed., Amsterdam: North-Holland, 1964, pp. 24-30.
[31] S.A. Cook, The complexity of theorem proving procedures, in *Proc. 3rd Symp. Theory of Computing*, New York: ACM, 1971, pp. 151-158.
[32] _______, Deterministic CFL's are accepted simultaneously in polynomial time and log squared space, in *Proc. 11th Symp. Theory of Computing*, New York: ACM, April 1979, pp. 338-345.
[33] D.C. Cooper, Theorem proving in arithmetic without multiplication, *Mach. Intell.*, 7 (1972), pp. 91-100.
[34] L. Csanky, Fast parallel matrix inversion algorithms, *SIAM J. Comput.*, 5 (1976), pp. 618–623.
[35] H.B. Curry, An analysis of logical substitution, *Amer. J. Math.*, 51 (1929), pp. 363-384.
[36] R.A. DeMillo and R.J. Lipton, A probabilistic remark on algebraic program testing, *Inf. Proc. Lett.*, 7 (1978), pp. 193-195.
[37] J. Edmonds, Paths, trees, and flowers, *Canadian J. Math.*, 17 (1965), pp. 449-467.
[38] U. Feige, S. Goldwasser, L. Lovasz, S. Safra, and M. Szegedy, Interactive proofs and the hardness of approximating cliques, *J. Assoc. Comput. Mach.*, 43 (1996), pp. 268-292.
[39] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, New York: Wiley, 1950.
[40] J. Ferrante and C. Rackoff, A decision procedure for the first order theory of real addition with order, *SIAM J. Comput.*, 4 (1975), pp. 69-76.
[41] _______, *The Computational Complexity of Logical Theories*, vol. 718 of *Lecture Notes in Mathematics*, New York: Springer-Verlag, 1979.
[42] M.J. Fischer and M.O. Rabin, Superexponential complexity of Presburger arithmetic, in *Complexity of Computation, SIAM–AMS*

Proceedings, vol. 7, Amer. Math. Soc., 1974, pp. 27-41.
[43] S. Fortune, A note on sparse complete sets, *SIAM J. Comput.*, 8 (1979), pp. 431-433.
[44] N. Francez, *Fairness*, New York: Springer-Verlag, 1986.
[45] R.M. Friedberg, Two recursively enumerable sets of incomparable degrees of unsolvability, *Proc. Nat. Acad. Sci.*, 43 (1957), pp. 236-238.
[46] M. Furst, J. Saxe, and M. Sipser, Parity, circuits, and the polynomial time hierarchy, *Math. Syst. Theory*, 17 (1984), pp. 13–27.
[47] K. Gödel, On undecidable propositions of formal mathematical systems, in *The Undecidable*, M. Davis, ed., Hewlitt, NY: Raven Press, 1965, pp. 5–38.
[48] O. Goldreich, S. Micali, and A. Wigderson, Proofs that yield nothing but their validity, and a methodology of cryptographic protocol design, in *Proc. 27th Symp. Foundations of Computer Science*, IEEE, October 1986, pp. 174–187.
[49] S. Goldwasser, S. Micali, and C. Rackoff, The knowledge complexity of interactive proof systems, *SIAM J. Comput.*, 18(1989), pp. 186–208.
[50] S. Goldwasser and M. Sipser, Private coins vs. public coins in interactive proof systems, in *Proc. 18th Symp. Theory of Computing*, ACM, May 1986, pp. 59–68.
[51] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, 5th ed., Oxford, UK: Oxford University, 1979.
[52] D. Harel, Effective transformations on infinite trees, with applications to high undecidability, dominoes, and fairness, *J. Assoc. Comput. Mach.*, 33 (1986), pp. 224–248.
[53] D. Harel and D. Kozen, A programming language for the inductive sets, and applications, *Inf. Control*, 63 (1984), pp. 118–139.
[54] D. Harel, D. Kozen, and J. Tiuryn, *Dynamic Logic*, Cambridge, MA: MIT Press, 2000.
[55] J. Hartmanis and R.E. Stearns, On the computational complexity of algorithms, *Trans. Amer. Math. Soc.*, 117 (1965), pp. 285–306.
[56] J. Håstad, Almost optimal lower bounds for small depth circuits, in *Proc. 18th Symp. Theory of Computing*, New York: ACM, 1986, pp. 6–20.
[57] _______, Clique is hard to approximate within $n1-\varepsilon$, *Acta Math.*, 182 (1999), pp. 105-142.

[58] _______, Some optimal inapproximability results, *J. Assoc. Comput. Mach.*, 48 (2001), pp. 798-859.
[59] F.C. Hennie, One-tape off-line Turing machine computations, *Inf. Control*, 8 (1965), pp. 553-578.
[60] F.C. Hennie and R.E. Stearns, Two-tape simulation of multitape turing machines, *J. Assoc. Comput. Mach.*, 13 (1966), pp. 533-546.
[61] J. Hopcroft, R. Motwani, and J. Ullman, *Introduction to AutomataTheory, Languages, and Computation*, 2nd ed., Reading, MA: Addison-Wesley, 2001.
[62] J.E. Hopcroft and R.M. Karp, An *n*5/2 algorithm for maximum matching in bipartite graphs, *SIAM J. Comput.*, 2 (1973), pp. 225-231.
[63] J.E. Hopcroft and J.D. Ullman, *Introduction to Automata Theory, Languages and Computation*, Reading, MA: Addison-Wesley, 1979.
[64] D. Ierardi and D. Kozen, Parallel resultant computation, in *Synthesis of Parallel Algorithms*, J. Reif, ed., San Francisco: Morgan Kaufmann, 1993, pp. 679-720.
[65] N. Immerman, Nondeterministic space is closed under complement, *SIAM J. Comput.*, 17 (1988), pp. 935-938.
[66] D. Johnson, Approximation algorithms for combinatorial problems, *J. Comput. Syst. Sci.*, 9 (1974), pp. 256-278.
[67] N.D. Jones, Space-bounded reducibility among combinatorial problems, *J. Comput. Syst. Sci.*, 11 (1975), pp. 68-85.
[68] N.D. Jones, E. Lien, and W. Laaser, New problems complete for nondeterministic logspace, *Math. Syst. Theory*, 10 (1976), pp. 1-17.
[69] H. Karloff and U. Zwick, A 7/8 approximation algorithm for MAX3SAT?, in *Proc. 38th Symp. Foundations of Computer Science*, Los Alamitos, CA: IEEE, 1997, pp. 406-415.
[70] R.M. Karp, Reducibility among combinatorial problems, in *Complexity of Computer Computations*, R.E. Miller and J.W. Thatcher, eds., New York: Plenum, 1972, pp. 85-103.
[71] S.C. Kleene, A theory of positive integers in formal logic, *Amer. J. Math.*, 57 (1935), pp. 153-173, 219-244.
[72] _______, On notation for ordinal numbers, *J. Symbolic Logic*, 42 (1938), pp. 150-155.
[73] _______, *Introduction to Metamathematics*, Princeton, NJ: D. van Nostrand, 1952.
[74] _______, On the forms of the predicates in the theory of constructive

ordinals (second paper), *Amer. J. Math.*, 77 (1955), pp. 405-428.
[75] D. Kozen, *The Design and Analysis of Algorithms*, New York:Springer-Verlag, 1991.
[76] _______ , *Automata and Computability*, New York: Springer-Verlag, 1997.
[77] _______, Computational inductive definability, *Ann. Pure Appl. Logic*, 126 (2004), pp. 139-148.
[78] R.A. Ladner, The circuit value problem is logspace complete for P, *SIGACT News*, 7 (1975), pp. 18-20.
[79] L.A. Levin, Universal sorting problems, *Prob. Inf. Transmission*, 9 (1973), pp. 265–266.
[80] L. Lovász, On determinants, matchings, and random algorithms, in *Proc. Symp. on Fundamentals of Computing Theory*, L. Budach, ed., Berlin: Akademia-Verlag, 1979, pp. 565-574.
[81] C. Lund, L. Fortnow, H. Karloff, and N. Nisan, Algebraic methods for interactive proof systems, *J. Assoc. Comput. Mach.*, 39 (1992), pp. 859-868.
[82] S.R. Mahaney, Sparse complete sets for *NP*: solution of a conjecture by Berman and Hartmanis, *J. Comput. Syst. Sci.*, 25 (1982), pp. 130-143.
[83] E.M. McCreight and A.R. Meyer, Classes of computable functions defined by bounds on computation: Preliminary report, in *Proc. 1st ACM Symp. Theory of Computing (STOC'69)*, New York: ACM, 1969, pp. 79-88.
[84] R. McNaughton, Testing and generating infinite sequences by a finite automaton, *Inf. Control*, 9 (1966), pp. 521-530.
[85] A.R. Meyer and D.M. Ritchie, The complexity of loop programs, in *Proc. ACM Natl. Meeting*, 1967, pp. 465-469.
[86] G.L. Miller, Riemann's hypothesis and tests for primality, *J. Comput. Syst. Sci.*, 13 (1976), pp. 300-317.
[87] Y.N. Moschovakis, *Elementary Induction on Abstract Structures*, Amsterdam: North-Holland, 1974.
[88] A.A. Muchnik, On the unsolvability of the problem of reducibility in the theory of algorithms, *Doklady Akademii Nauk SSSR*, 108 (1956), pp. 194-197.
[89] D. Muller, Infinite sequences and finite machines, in *Proc. 4th Symp. Switching Circuit Theory and Logical Design*, Los Alamitos,

CA: IEEE, 1963, pp. 3-16.
[90] J. Myhill, Creative sets, *Zeitschrift für mathematische Logik und Grundlagen der Mathematik*, 1 (1955), pp. 97-108.
[91] D.C. Oppen, An upper bound on the complexity of Presburger arithmetic, *J. Comput. Syst. Sci.*, 16 (1978), pp. 323-332.
[92] C.H. Papadimitriou, *NP*-completeness: A retrospective, in *24th Int. Colloq. Automata, Languages and Programming (ICALP'97)*, P. Degano, R. Gorrieri, and A.Marchetti-Spaccamela, eds., vol. 1256 of *Lecture Notes in Computer Science*, Bologna, Italy, New York: Springer-Verlag, July 1997, pp. 2-6.
[93] C.H. Papadimitriou and S. Zachos, Two remarks on the power of counting, in *Proc. 6th GI Conf. Theoretical Computer Science*, New York: Springer-Verlag, 1982, pp. 269-276.
[94] E.L. Post, Finite combinatory processes-formulation, I, *J. Symbolic Logic*, 1 (1936), pp. 103-105.
[95] _______, Formal reductions of the general combinatorial decision problem, *Amer. J. Math.*, 65 (1943), pp. 197–215.
[96] _______, Recursively enumerable sets of positive integers and their decision problems, *Bull. Amer. Math. Soc.*, 50 (1944), pp. 284–316.
[97] V.R. Pratt, Every prime has a succinct certificate, *SIAM J. Comput.*, 4 (1975), pp. 214-220.
[98] M. Presburger, Über die Vollständigkeit eines gewissen Systems der Arithmetik ganzer Zahlen, in welchen die Addition als einzige Operation hervortritt, *Comptes Rendus, I. Congrès des Math. des Pays Slaves*, (1929), pp. 192-201.
[99] M.O. Rabin, Decidability of second order theories and automata on infinite trees, *Trans. Amer. Math. Soc.*, 141 (1969), pp. 1-35.
[100] _______, Probabilistic algorithms for testing primality, *J. Num. Theory*, 12 (1980), pp. 128–138.
[101] O. Reingold, Undirected ST-connectivity in log-space, in *Proc.37th Symp. Theory of Computing*, New York: ACM, May 2005, pp. 376-385.
[102] H.G. Rice, Classes of recursively enumerable sets and their decision problems, *Trans. Amer. Math. Soc.*, 89 (1953), pp. 25-59.
[103] _______, On completely recursively enumerable classes and their key arrays, *J. Symbolic Logic*, 21 (1956), pp. 304-341.
[104] H. Rogers, *Theory of Recursive Functions and Effective Computability*, New York: McGraw-Hill, 1967.

[105] R. Rubinfeld and M. Sudan, Robust characterizations of polynomials with applications to program testing, *SIAM J. Comput.*, 25 (1996), pp. 252-271.
[106] W. Ruzzo, On uniform circuit complexity, *J. Comput. Syst. Sci.*, 22 (1981), pp. 365-383.
[107] S. Safra, On the complexity of ω-automata, in *Proc. 29th Symp. Foundations of Comput. Sci*, Los Alamitos, CA: IEEE, October 1988, pp. 319-327.
[108] W. Savitch, Relationship between nondeterministic and deterministic tape complexities, *J. Comput. Syst. Sci.*, 4 (1970), pp. 177-192.
[109] M. Schönfinkel, Über die Bausteine der mathematischen Logik, *Math. Annalen*, 92 (1924), pp. 305-316.
[110] J.T. Schwartz, Fast probabilistic algorithms for verification of polynomial identities, *J. Assoc. Comput. Mach.*, 27 (1980), pp. 701-717.
[111] A. Shamir, *IP = PSPACE*, in *Proc. 31st Symp. Foundations of Computer Science*, Los Alamitos, CA: IEEE, 1990, pp. 11-15.
[112] M. Sipser, A complexity theoretic approach to randomness, in *Proc.15th Symp. Theory of Computing*, New York: ACM, 1983, pp. 330-335.
[113] _______, *Introduction to the Theory of Computation*, Pacific Grove, CA: Brooks Cole, 1996.
[114] R.I. Soare, *Recursively Enumerable Sets and Degrees*, Berlin: Springer- Verlag, 1987.
[115] R. Stearns, J. Hartmanis, and R. Lewis, Hierarchies of memory limited computations, in *Proc. IEEE Conf. Switching Circuit Theory and Logical Design*, 1965, pp. 179-190.
[116] L. Stockmeyer and A. Chandra, Provably difficult combinatorial games, *SIAM J. Comput.*, 8 (1979), pp. 151-174.
[117] L.J. Stockmeyer, The polynomial-time hierarchy, *Theor. Comput. Sci.*, 3 (1976), pp. 1-22.
[118] L.J. Stockmeyer and A.R. Meyer, Word problems requiring exponential time, in *Proc. 5th Symp. Theory of Computing*, NewYork: ACM, 1973, pp. 1-9.
[119] R. Szelepcsényi, The method of forcing for nondeterministic automata, *Bull. EATCS*, 33 (1987), pp. 96-100.

[120] S. Toda, On the computational power of *PP* and $\oplus$ *P*, in *Proc. 30th Symp. Foundations of Computer Science (FOCS'89)*, Los Alamitos, CA: IEEE, 1989, pp. 514-519.
[121] _______, *PP* is as hard as the polynomial-time hierarchy, *SIAM J. Comput.*, 20 (1991), pp. 865-877.
[122] B.A. Trakhtenbrot, Turing computations with logarithmic delay, *Algebra i Logika*, 3 (1964), pp. 33-48.
[123] A.M. Turing, On computable numbers with an application to the Entscheidungsproblem, *Proc. London Math. Soc.*, 42 (1936), pp. 230-265. Erratum: Ibid., 43 (1937), pp. 544-546.
[124] L.G. Valiant, The complexity of computing the permanent, *Theor. Comput. Sci.*, 8 (1979), pp. 189-201.
[125] L.G. Valiant and V.V. Vazirani, *NP* is as easy as detecting unique solutions, in *Proc. 17th ACM Symp. Theory of Computing (STOC'85)*, New York: ACM, 1985, pp. 458-463.
[126] A.C. Yao, Separating the polynomial-time hierarchy by oracles, in *Proc. 26th Symp. Foundations of Computer Science (FOCS'85)*, Los Alamitos, CA: IEEE, 1985, pp. 1-10.
[127] R.E. Zippel, Probabilistic algorithms for sparse polynomials, in *Proc. EUROSAM 79*, Ng, ed., vol. 72 of *Lect. Notes in Comput.Sci.*, New York: Springer-Verlag, 1979, pp. 216-226.

Белгілеулер және Аббревиатуралар

белгі	Аталуы	бөлім, бет
TM	Тьюринг машинасы..	1, 15
Σ	Шектеулі автомат.....................................	1, 16
$\lvert x \rvert$	x жолдың ұзындығы.....................................	1, 16
$\lvert A \rvert$	A жиынның қуаты..	1, 16
ε	Бос жол..	1, 16
Σ^*	Σ алфавиттегі ұзындығы шектеулі жолдар..	1, 16
$\vdash$	Сол жақ шеткі маркер.....................................	1, 16
Γ	Таспа алфавиті...	1, 16
$\sqcup$	Бос символ..	1, 16
$L(M)$	M Тьюринг машинасында қабылдау-ашық жолдар жиыны................................	1, 17
δ	Ауысу функциясы..	1, 17
PAL	Палиндромдар	1, 20
$b_k(i)$	i -дің k -битті бинарлы өрнектелуі.............	1, 22
$T(n)$	Уақытпен шектелу..	1, 24

$\mathbb{N}$	Натурал сандар	1, 24
$S(n)$	Кеңістікті шектелу	1, 24
n	Енгізу жолының ұзындығы	1, 24
$DTIME(T(n))$	Детерминирлі уақыт класы	1, 25
$NTIME(T(n))$	Бейдетерминирлі уақыт класы	1, 25
$DSPACE(S(n))$	Детерминирлі кеңістік класы	1, 25
$NSPACE(S(n))$	Бейдетерминирлі	1, 25
$\mathrm{co}-\mathcal{A}$	$\mathcal{A}$ -ның жиындарының комплиментінің жиыны	1, 25
$LOGSPACE$	Детерминирлі $\mathrm{logspace}$	1, 26
$NLOGSPACE$	Бейдетерминирлі $\mathrm{logspace}$	1, 26
P	Детерминирлі полиномиалды уақыт	1, 26
NP	Бейдетерминирлі полиномиалды уақыт	1, 26
$PSPACE$	Детерминирлі полиномиалды кеңістік	1, 26
$NPSPACE$	Бейдетерминирлі полиномиалды кеңісті.	1, 27
$EXPTIME$	Детерминирлі экспоненттті уақыт	1, 27
$NEXPTIME$	Бейдетерминирлі экспоненттті уақыт	1, 27
$EXPSPACE$	Детерминирлі экспоненттті кеңістік	1, 27
$NEXPSPACE$	Бейдетерминирлі экспоненттті кеңістік	1, 27
$\xrightarrow{\leq k}$	Конфигурацияға қабылдау-ашық қатына сы	1, 31
$\#(x)$	Бинарлы сан x -пен кескінделген сан	1, 33
k - FA	k -бастиекті шектеулі автомат	1, 42

$\leq_{\mathrm{m}}^{\log}$	Көпмәнді logspace келтірімділік........... .....1, 43
$\leq_{\mathrm{m}}^{\mathrm{p}}$	Полиномиал уақыттағы көпмәнді келтірі мділік...1, 44
MAZE	Бағытталған графтың жете алушылығы.........1, 45
SAT	Логикалық орындалушылық...........................1, 48
CVP	Схеманың мәндер проблемасы.......................1, 49
CNF	Конъюктивті қалыпты форма..........................1, 54
$k-\mathrm{CNF}$	k -конъюктивті қалыпты форма.....................1, 54
3SAT	3CNF формула үшін орындалушылық проблемасы..1, 54
ω	Ең кіші шексіз ординал....................................1, 55
α, β, γ ...	Ординалдар 1, 56
Ord	Барлық ординалдар класы 1, 56
ZF	Цермало-Франкел жиындар теориясы...........1, 59
$\sup A$	A жиынының супремумы..............................1, 59
$x \vee y$	x және y -ті решеткада жалғау.....................1, 59
$\top$	Толық решетканың максималды элементі.....1, 59
$\perp$	Толық решетканың минималды элементі......1, 59
$\inf A$	A жиынының инфинимумы.........................1, 60
$\bigcup\mathcal{C}$	$\mathcal{C}$ жиындар кластарының бірігуі..................1, 60
$\bigcap\mathcal{C}$	$\mathcal{C}$ жиындар кластарының қиылысуы............1, 60
$PF_{\tau}(x)$	x -тегі τ -дың префикс нүктесі.....................1, 61
$\tau^{\dagger}(x)$	x -тегі τ -дың ең төменгі префикс нүкт есі.. .....1, 62
ι	Теңбе-тең қатынас.................................... .....1, 64

R^*	R сақинаның кері элементтері....................1, 126
$\varphi(n)$	Эйлердің φ функциясы................................1, 126
$\mathbb{GF}_q$	q элементті өріс...1, 127
$m \mid n$	n-ді m бөледі...1, 127
$\overline{\mathbb{F}}$	Өріс $\mathbb{F}$-тің алгебралық тұйықталуы.........1, 133
IP	Тиімді интерактивті дәлелдемелер..............1, 138
P	Дәлелдеуші 1, 138
V	Қабылдаушы 1, 138
$\mathbb{Z}$	Бүтін сандар сақинасы 1, 146
PCP	Ықтималды тексерілетін дәлелдемелер.......1, 157
$PCP(r(n), q(n))$	$O(r(n))$ кездейсоқ битті және $O(q(n))$ сұратымды PCP 1, 158
PTAS	Аппроксимацияның полиномиалды уақыт схемасы..1, 159
DNF	Дизъюнктивті қалыпты форма......................1, 163
$x \bullet y$	Ішкі көбейту 1, 166
$\otimes$	Тензорлық көбейтінді 1, 173
u^T	u-дың транспонирленген матрицасы 1, 175
rank C	C матрицаның рангы 1, 176
dim E	Сызықты кеңістік E-нің өлшемі 1, 176
$f^{\mathcal{A}}$	$\mathcal{A}$ структурада f символын түсіну 1, 179
0	Жалған 1, 179
1	Ақиқат 1, 179

$u[x/a]$	Жаңа u бағалауда x айнымалыны a -ның мәніне қайтадан байланыстыру... ...1, 181
$\models$	Қанағаттандыру 1, 182
$\mathrm{Th}(\mathcal{A})$	$\mathcal{A}$ структураның бірінші ретті теориясы...1, 185
$\mathrm{Th}(\mathbb{C})$	$\mathbb{C}$ структура класының бірінші ретті теориясы...1, 185
$\mathrm{Mod}(\Phi)$	Φ -дің модельдері 1, 185
$\mathbb{Q}$	Рационал сандар 1, 190
$\mathbb{R}$	Нақты сандар 1, 190
$\Vert f \Vert$	f -тің өлшемі 1, 196
$sign(x)$	x -тың белгісі 1, 197
S1S	Мұрагердің екінші ретті монадикалық теориясы 1, 213
SnS	n мұрагердің екінші ретті монадикалық теориясы 1, 213
s	Мұрагер функциясы 1, 214
$\mathrm{IO}(\sigma)$	σ жүргізудің IO жиыны 1, 216
PCP	Пост сәйкестену проблемасы 1, 220
$\sqsubseteq$	Аппроксимациялау 1, 229
$\Sigma^{\le k}$	Σ -дағы ұзындығы k -дан аспайтын жолдар 1, 244
USAT	Ерекше орындалушылық 2,11
$A^{\perp}$	Ортогонал толықтауыш 2,11
$< A >$	A -ның сызықты қабықшасы.................. 2,13
$\mathrm{codim}\, E$	Сызықты E кеңістігінің ко-өлшемі 2,13

φ^{-W}	Жойылған элементтері W -да болатын φ 2,45
б.ж.д	Барлық жағдайда дерлік.......................................2,59
ш.к	Шексіз көп 2,59
$f \circ g$	Функционалды композиция 2,59
$<\ ,\ >$	Жұп түзу функциясы 2,65
π_i^n	Проекция функциясы 2,65
κ_c	Мәні c-ға тең тұрақты функция 2,65
U	Универсал функция 2,66
s_n^m	s_n^m функция 2,66
comp	Композиция операторы 2,67
pair	Жұп құрастыру операторы 2,67
const	Тұрақты функция операторы 2,67
р.т	Рекурсивті тізбеленетін (р.т.)..............................2,72
$\overset{\infty}{\forall}$	Барлық, бірақ шектеулі көп 2,82
$\overset{\infty}{\exists}$	Шексіз көп табылады 2,82
$\leq_T$	Тьюринг келтірілімділігі 2,86
Σ_1^0	р.т. жиын 2,86
Δ_1^0	Рекурсивті жиындар 2,87
Π_1^0	ко-р.т. жиындар 2,87
Σ_n^0	Арифметикалық иерархиядағы кластар 2,87

ИНДЕКСТЕУ

ДЕКСТЕР С. КОЗЕН

ЕСЕПТЕУ ТЕОРИЯСЫ

2-бөлім

Оқулық

Басуға 06.11.2014 ж. қол қойылды. Қағазы офсеттік. Қаріп түрі «Times». Пішімі 60x90$^{1}/_{16}$. Офсеттік басылым. Баспа табағы 20
Таралымы: Мемлекеттік тапсырыс бойынша – 800 дана
Тапсырыс

Тапсырыс берушінің файлдарынан Қазақстан Республикасы «Полиграфкомбинат» ЖШС-де басылды.
050002, Алматы қаласы, М.Мақатаев көшесі, 41.