Теория 2ри семестър

Теория 1ва част

Лекция 1

Уводни дефиниции

Транзитивно затваряне

Нека A е множество и $R \leq A^2$ е релация в A. Рефлексивно и транзитивно затваряне R* на R се определя(индуктивно):

А) Ако $(a,b) \in R$, то $(a,b) \in R^*$;

Б) Ако $a \in A$ то $(a,a) \in R^*$;

В) Ако $(a,b) \in R^*$ и $(b,c) \in R^*$, то $(a,c) \in R^*$.

Затваряне относно функция

Нека $F{:}A^n \text{->} A$ и $B \subseteq A$. Затваряне B* на B относно функцията F се определя (индуктивно):

А) Ако $a \in B$, то $a \in B^*$;

Б) Ако $a_1 \ldots a_n \in B^*$ и $F(a_1 \ldots a_n)=b$, то $b \in B^*$.

Затваряне относно операции

Нека A е множество, $F_1 \ldots F_k$ са функции, F_i: A^n_i -> A, i=1…k, а $B \subseteq A$. Затваряне B* на B относно операциите $F_1 \ldots F_k$ се определя (индуктивно):

А) Ако $a \in B$, то $a \in B^*$;

Б) Ако a_1…a_$n_i \in B^*$ и $F_i(a_1$…a_$n_i)=b$, то $b \in B^*$.

Затваряне относно релация

Нека $R \leq A^{n+1}$, A е множество и $B \subseteq A$. Затваряне B* на B относно релацията R се определя (индуктивно):

А) Ако $a \in B$, то $a \in B^*$;

Б) Ако $a_1 \ldots a_n \in B^*$ и $(a_1 \ldots a_n , b) \in R$, то $b \in B^*$.

Затваряне относно релации

Нека A е множество, $R_i \leq A\text{^}(r_i+1)$, $i=1\ldots n$, и $B \subseteq A$. Затваряне B^* на B относно релациите $R_1\ldots R_n$ се определя (индуктивно):

А) Ако $a \in B$, то $a \in B^*$;

Б) Ако $a_1\ldots a\_r_i \in B^*$ и $(a_1\ldots a\_r_i, b) \in R_i$, то $b \in B^*$.

Азбуки, думи, езици. Операции върху думи и езициΣ

Азбука

Азбука наричаме всяко крайно множество от символи.

Примери: българската, латинската, обединение на част от българската с част от латинската, или всяко множество от букви, цифри или символи.

Дума

Нека Σ е азбука. Думата в азбуката Σ се нарича всяко изображение $w:\{1\ldots n\}\rightarrow\Sigma$.

Вместо $w(1)\ldots w(n)$ ние ще записваме още $w_1\ldots w_n$, а самата дума ще пишем $w=w_1w_2\ldots w_n$, т.е. редицата от букви в Σ, както е стандартното разбиране за дума. Дължината на думата w се означава с $|w|$ и $|w|=n$, т.е. броя на буквите в редицата $w_1w_2\ldots w_n$.

Празната дума:

Има една специална дума, която наричаме празната дума и тя се означава с ε. $\varepsilon: \emptyset\rightarrow\Sigma$, т.е. дължината на ε е 0, т.е. $|\varepsilon|=0$. Така $w:I_n\rightarrow\Sigma$ е дума, като при n=0, $I_n=0$ и $|w|=0$, т.е. ε=w.

Операции върху думи

1. Конкатенация

Нека u и v са думи от азбуката Σ. Тогава конкатенацията на u и v се означава u.v и ако w=u.v то w е дума с дължина $|w|=|u|+|v|$ и $w:I_{|u|+|v|}\rightarrow\Sigma$ като $w(i)=u(i)$ за $1\leq i\leq|u|$ и $w(j)=v(j-|u|)$ за $1\leq j\leq|u|+|v|$, т.е. ако $u=u_1\ldots u_n$ и $v=v_1\ldots v_m$, то $w= u_1\ldots u_n\ v_1\ldots v_m$. Операцията е асоциативна, т.е. $(u\cdot v)\cdot w=u\cdot(v\cdot w)$, но не е комутативна в общия случай.

Проста конкатенация на думите u и v ще означаваме с uv.

Начало на дума

Казваме че думата u е начало на думата v ако съществува думата x, такава че $u \cdot x = v$.

Край на дума

Казваме че думата u е край на думата v ако съществува думата x, такава че $x \cdot u = v$.

Поддума

Казваме че думата u е поддума на v ако съществува думите x и y, такава че $x \cdot u \cdot y = v$.

2. Степенуване на думи

Нека w е дума, и n е естествено число. Ще определим индуктивно w^n както следва:

1) $w^0 = \varepsilon$;
2) $w^{n+1} = w^n \cdot w$

Иначе казано $w^n = w.w.w\ldots w$ (n пъти).

3. Обратна дума (reversal) на дадена дума w

Означаваме с w^R и дефинираме индуктивно както следва:

1) $|w|=0$, то $w=\varepsilon$ и $w^R=\varepsilon$
2) $|w|=n+1$, $w=u \cdot a$, $a \in \Sigma$, то $w^R = a \cdot u^R$

Свойство на обратните думи при конкатенация: $(u.v)^R = v^R.u^R$

С Σ^* означаваме всички думи в азбуката Σ.

Език

Нека Σ е азбука. Език в азбуката Σ се нарича всяко множество от думи в азбуката Σ, т.е. всеки език L е подмножество на Σ^*.

Примери: Езици са $\emptyset$, Σ и Σ^*.

Буква се среща в дума

Казваме че буквата a, $a \in \Sigma$ се среща в думата w, ако a е поддума на w.

Операции върху езици

1. Теоретично-множествени операции

Тъй като всеки език е множество върху него могат да се изпълняват теоретично-множествени операции (като $\cup$,$\cap$ и /).

2. Конкатенация

Нека L_1 и L_2 са езици. Конкатенация на L_1 и L_2 се означава с $L_1 \cdot L_2$ и $L_1 \cdot L_2 = \{w | w = w_1.w_2$ за някои $w_1 \in L_1$ и $w_2 \in L_2\}$.

3. Степенуване

Нека L е език и n е ест. число. Тогава L^n се определя индуктивно:

1) $L^0 = \{\varepsilon\}$
2) $L^{n+1} = L^n.L$

!!! $\emptyset^0 = \{\varepsilon\}$.

4. Звезда на Клини за език

Нека L е произволен език L* може да се определи по два начина:

1) $L^* = \bigcup L^n$, $n \in [0, +\infty)$;
2) $L^* = \{w | w \in \Sigma$ и $w = w_1 \dots w_k$ за $w_1, \dots, w_k \in L$ и $k \in \mathbb{N}\}$ това вкл. и $k=0$;

Така $\emptyset^* = \{\varepsilon\}$.

Използваме се още и L^+ като $L^+ = \bigcup L^n$, $n \in [1, +\infty)$.

!!! $\varepsilon \in L^*$ винаги но $\varepsilon \in L^+$ само ако $\varepsilon \in L$. Освен това означението Σ^* се съгласува със звездата на Клини на Σ.

Регулярни езици

Регулярен израз в (над) език

Нека Σ е азбука. Регулярен израз над (в) Σ се определя индуктивно както следва:

1) $\emptyset$ и всяка буква от Σ е регулярен израз;
2) Ако α и β са регулярни изрази то $(\alpha \,.\, \beta)$, $(\alpha \cup \beta)$ и α^* са регулярни изрази;

Регулярен език определен от регулярен израз (индуктивна деф.)

С помощта на регулярен израз α определяме регулярен език $L[\alpha]$.

Индуктивната дефиниция на $L[\alpha]$ с индукция относно построението на α:

1) Ако $\alpha = \emptyset$ то $L[\alpha] = \emptyset$;

2) Ако $\alpha=a\in\Sigma$ то $L[\alpha]=\{a\}$;

3) Ако $\alpha=(\alpha_1\alpha_2)$ то $L[\alpha]=L[\alpha_1].L[\alpha_2]$;

4) Ако $\alpha=(\alpha_1\cup\alpha 2)$ то $L[\alpha]= L[\alpha_1]\cup L[\alpha_2]$;

5) Ако $\alpha=\alpha_1{}^*$ то $L[\alpha]=(L[\alpha_1])^*$;

Регулярен език

Казваме че един език L е регулярен, ако съществува регулярен израз α, такъв че $L[\alpha]=L$.

С други думи, регулярните езици се генерират (пораждат) от регулярни изрази.

Крайни детерминирани автомати

Краен детерминиран автомат (КДА)

Краен детерминиран автомат (КДА) се нарича петицата $<K,\Sigma,\delta,s,F>$ такива че:

K- крайно множество от състояния;

Σ- основна азбука чиито думи ще разпознаваме;

$s\in K$, s – начални състояния;

$F\subseteq K$, F – множество от заключителни състояния;

δ: $K x\Sigma \rightarrow K$, δ- функция на преходите.

Конфигурация за КДА

Нека $M=<K,\Sigma,\delta,s,F>$ е КДА. Конфигурация за M се нарича всяка двойка на $Kx\Sigma^*$, т.е. конфигурация е всяка двойка (q,w) където $q\in K$ и $w\in\Sigma^*$.

За конфигурация на M трябва да си мислим като състоянието q в което се намира „устройството за контрол“, и за w като непрочетената част от думата. КДА са прости устройства които нямат памет и главата никога не се връща назад да прочете отново прочетената буква.

Релация между конфигурации

Нека (q,w) и (q_1,w_1) са две конфигурации за M. Казваме че конфигурацията (q,w) се преработва за една стъпка от M в конфигурацията (q_1,w_1) и пишем $(q,w) \vdash_M (q_1,w_1)$ т.т.к съществува $a \in \Sigma$ такова че $w=a.w_1$ и $\delta(q,a)=q_1$.

Когато стигнем до конфигурация (q,ε) значи сме прочели окончателно входа (входната дума).

$\vdash_M^*$ - озн. рефлективното итранзитивното затваряна на релацията $\vdash_M$;

$(q,w) \vdash_M^* (q_1,w_1)$ четем от (q,w) се извежда (q_1,w_1) (за много стъпки);

$(q,w) \vdash_M^* (q,w)$ казваме че от (q,w) се извежда (q,w) за 0 стъпки;

$(q_0,w_0) \vdash_M \ldots \vdash_M (q_n,w_n)$ казваме че от (q_0,w_0) се извежда (q_n,w_n) за n стъпки;

КДА приема дума

Нека $M=\langle K,\Sigma,\delta,s,F\rangle$ е КДА. Казваме че думата $w \in \Sigma^*$ се приема (разпознава) от M т.т.к. $(s,w) \vdash_M^* (f,\varepsilon)$ и $f \in F$.

Лекция 2

Недетерминирани крайни автомати

Недетерминиран краен автомат (НКА)

Недетерминиран краен автомат (НКА) се нарича петицата $M=<K,\Sigma,\Delta,s,F>$, където:

K- крайно множество от състояния;

Σ- основна азбука;

$s\in K$, s – начални състояния;

$F\subseteq K$, F – множество от заключителни състояния;

Δ- релация на преходите, $\Delta\subseteq K\times(\Sigma\cup\{\varepsilon\})\times K$. Всяка тройка $(q,u,p)\in\Delta$ се нарича правило за преход.

Конфигурация за НКА

Нека $M=(K,\Sigma,\Delta,s,F)$ е НКА. Конфигурация за M се нарича всеки елемент на $K\times\Sigma^*$.

Релация между конфигурации

Нека (q,w) и (q_1,w_1) са две конфигурации за $M=(K,\Sigma,\Delta,s,F)$. Казваме че конфигурацията (q,w) се преработва за една стъпка се извежда (q_1,w_1) с помощта на M и пишем $(q,w) \vdash_M (q_1,w_1)$ т.т.к съществува $u\in\Sigma\cup\{\varepsilon\}$такова че $w=u.w_1$ и $(q,u,q_1)\in\Delta$.

$\vdash_M^*$ - озн. рефлективното итранзитивното затваряна на релацията $\vdash_M$;

НКА приема дума

Нека $M=(K,\Sigma,\Delta,s,F)$ е НКА. Казваме че думата $w\in\Sigma^*$ се приема (разпознава) от M т.т.к. $(s,w) \vdash_M^*(f,\varepsilon)$ и $f\in F$.

НКА разпознава език

$L(M)=\{w|\ w\in\Sigma^*$ и w се разпознава от $M\}$. L(M) се нарича език който се разпознава от M.

Еквивалентни автомати

Казваме че два автомата M_1 и M_2 (независимо дали са КДА или НКА) са еквивалентни т.т.к. $L(M_1)=L(M_2)$.

Теорема за еквивалентни автомати

За всеки недетерминиран краен автомат (НКА) съществува еквивалентен на него краен детерминиран автомат (КДА).

Дефиниция допълнителна за затваряне

За всяко $q\in K$ с $E(q)=\{p|\ (q,\varepsilon)\ |\text{-}_M{}^* (p,\varepsilon)\}$. $E(q)$ можем да го дефинираме като затваряне на мн. $\{q\}$ относно релацията $\{(p,r)|\ (p,\varepsilon,r)\in\Delta\}$

Помощно твърдение за дума и състояние

М- НКА; М1- КДА

За всяка дума $w\in\Sigma^*$ и за произволни състояния $p,q\in K$ е изпълнена еквивалентността: $(q,w)\ |\text{-}_M{}^* (p,\varepsilon) \Leftrightarrow (E(q),w)\ |\text{-}_{M1}{}^* (P,\varepsilon)$ за някое P, $p\in P$.

Теорема

Нека $w\in\Sigma^*$. Тогава $w\in L(M) \Leftrightarrow (s,w)\ |\text{-}_M (f,\varepsilon)$ за някое $f\in F \Leftrightarrow (E(s),w)|\text{-}_M(Q,\varepsilon)$ за някое Q такова че $f\in Q \Leftrightarrow w\in L(M_1)$ защото $Q\in F_1$ т.т.к. $Q\cap F\neq\emptyset$.

Крайни автомати и регулярни езици

Затворено множество (двоични функции)

Едно множество F от двоични функции се нарича затворено, ако F=[F] т.е. суперпозицията на произволна функция от F принадлежи на F

Затворено множество относно функция

Нека $F:A^n\rightarrow A$ и $B\subseteq A$. Казваме че B е затворено относно F ако за всички $a_1\ldots a_n\in B$ е изпълнено че $F(a_1\ldots a_n)\in B$.

Твърдение за затвореност на класа на автоматните езици

Класът на автоматните езици е затворен относно операциите:

1) Обединение;
2) Конкатенация;

3) Звезда на Клини;

4) Допълнение;

5) Сечение.

Теорема за регулярен език и автоматен езици

Един език е регулярен т.т.к. той е автоматен.

Лекция 3

Лема за разрастването

Нека L е регулярен език. Тогава съществува такова естествено число $n \geq 1$, че за всяка дума $w \in L$, такава че $|w| \geq n$, съществуват думи x, y и z, такива че $w = x . y . z$, $y \neq \varepsilon$ и $|x.y| \leq n$ и за всяко $i \in N$ е изпълнено че $x . y^i . z \in L$.

Релацията на еквивалентност $\approx_L$ за даден език L

Нека L е език в азбуката Σ и $x,u \in \Sigma^*$. Казваме че x и y са еквивалентни относно L (пишем $x \approx_L y$) ако за всяка дума $z \in \Sigma^*$ е изпълнена еквивалентността: $x . z \in L \Leftrightarrow y . z \in L$.

Релацията $\approx_L$ е релация на еквивалентност в Σ^*:

1) $x \approx_L x$ (рефлективност) ;

2) ако $x \approx_L y$ то $y \approx_L x$ (симетричност);

3) Ако $x \approx_L y$ и $y \approx_L z$ то за всяко $s \in \Sigma^*$ е изпълнено че
$x.s \in L \Leftrightarrow y.s \in L \Leftrightarrow z.s \in L$ т.е. $x \approx_L z$.

Релация на еквивалентност $\sim_M$ за даден автомат M

Нека $M=(K,\Sigma,\delta,s,F)$ е КДА. Казваме че думите x и y са еквивалентни относно M (пишем $x \sim_M y$) т.т.к. съществува състояние $q \in K$, такова че $(s, x) \vdash_M^* (q, \varepsilon)$ и $(s, y) \vdash_M^* (q, \varepsilon)$.

Класа на еквивалентност е $E_q=\{x | (s,x) \vdash_M^* (q,\varepsilon)\}$.

Твърдение за еквивалентностите спрямо автомат M и автоматен език на M : L(M)

За всеки КДА M=(K,Σ,δ,s,F) и за произволни думи x, y∈Σ е изпълнено ако x $\sim_M$ y то x $\approx_{L(M)}$ y.

Теорема на Майхил-Нероуд

Нека L е регулярен език в Σ. Тогава съществува краен детерминиран автомат M, който разпознава L с точно толкова състояния, колкото са класовете на еквивалентност относно релацията $\approx_L$.

Следствие от теоремата на Майхил-Нероуд

Един език L е регулярен т.т.к. релацията на еквивалентност $\approx_L$ има крайно много класове на еквивалентност т.е. т.т.к. $\approx_L$ има краен индекс.

Лекция 4

Еквивалентност на състояния

Нека p,q∈K. Казваме че p и q са еквивалентни (пишем p≡q) за КДА M=(K,Σ,δ,s,F) т.т.к за всички думи z∈Σ* е изпълнена еквивалентността: $(q,z)\in A_M \Leftrightarrow (p,z)\in A_M$

Две състояния p и q са еквивалентни т.т.к. E_q и E_p се съдържат в един и същ клас на еквивалентност относно $\sim_M$

Лема за еквивалентност на състояния

За всеки две състояния p,q∈K и за всяко n≥1. $p\equiv_n q$ т.т.к са изпълнени следните две условия:

1) $p \equiv_{n-1} q$;
2) За всички a∈Σ, $\delta(q,a) \equiv_{n-1} \delta(p,a)$.

Скорост на растеж на функции и анализ на сложността на алгоритми

Ред на функция

Нека f:ℕ →ℕ. Ред на функцията f се озн. с O(f) и е множеството на всички функции g: ℕ →ℕ такива че съществува естествени числа c,d (c,d>0) такива че за всички естествени числа n е изпълнено че $g(n) \leq c.f(n)+d$.

Всъщност достатъчно е да е изпълнено за всички естествени числа от известно място нататък т.е. за $n \geq n_0$ за някое фиксирано n_0.

Ако $g \in O(f)$ то също казваме че степента на растеж на g не надвишава степента на растежа на f.(при M→+∞) понякога пишем още $\preccurlyeq$.

Ако за две функции f ,g: $\mathbb{N} \to \mathbb{N}$ е изпълнено че $f \in O(g)$ и $g \in O(f)$ то ще пишем $f \asymp g$ и ще казваме че f и g са от един и същ ред (порядък) или че имат една и съща скорост на растеж. Релацията $\asymp$ е релация на еквивалентност.

Твърдение за ред на функция

Нека f ,g: $\mathbb{N} \to \mathbb{N}$. Тогава, $\prec$ ако:

1) $\lim\limits_{n\to+\infty} \frac{f(n)}{g(n)} = k > 0$, то $f \asymp g$;
2) $\lim\limits_{n\to+\infty} \frac{f(n)}{g(n)} = 0$, то $f \preccurlyeq g$, но $f \not\asymp g$ и в такъв случай пишем f g.

Твърдение

За всяко j=0,1,…n след изпълнението на най-външния j-ти цикъл R* съдържа всички двойки (a_i,a_k) такива че съществува път с ранг ρ, за $\rho \leq j$.

Алгоритъма е експоненциален.

Експоненциална сложност

Един алгоритъм има експоненциална сложност ако той има ред $O(2^n)$, $O(3^n)$…$O(p^k)$ за фиксирано $p \in \mathbb{N}$.

Полиномиална сложност

Когато един алгоритъм има ред O(P) където P е полином, то казваме че алгоритъма има полинимиална сложност.

Затваряне относно релация

Нека A е множество с n елемента, $B \subseteq A$ и $R \subseteq A^{k+1}$. Затварянето B* на B относно R се определя както следва:

1) Ако $a \in B$ то $a \in B^*$;

2) Ако $a_1 \dots a_к \in B^*$ и $(a_1 \dots a_к, b) \in R$ то $b \in B^*$.

Сложността на алгоритъма е $O(n^{k+2})$.

Алгоритъма: $B^* := B$; while: съществува $(a_1 \dots a_к, b) \in R$ и $a_1 \dots a_к \in B^*$, но $b \notin B^*$;
do: добави b към B*

Сложност на изучени алгоритми (допълнително)

1) **Построяване на съответен регулярен израз по краен автомат** – експоненциален $(\sigma\ (3^{|k|}))$;

2) **Детерминизация на недетерминиран автомат** – експоненциален $(\sigma(2^{|k|}|k|^2|\Sigma||\Delta||k|^3))$;

3) **Проверка дали два крайни недетерминирани автомата са еквивалентни или не** – експоненциален;

4) **Проверка дали L (α1) = L (α2) по дадени два регулярни израза α1 и α2** – експоненциален.

5) **Минимизация на краен детерминиран автомат** – полиномиален $((\sigma\ (|K|^3|\Sigma|))$;

6) **Съответен краен недетерминиран автомат по регулярен израз** – полиномиален $(\sigma\ (2|\alpha| + 1))$;

7) **Проверка дали два крайни детерминирани автомата са еквивалентни или не** – полиномиален.

Лекция 5

Сложност на алгоритми за крайни автомати

Теорема за сложност на алгоритми

1) Съществува **експоненциален** алгоритъм който по даден НКА дава еквивалентен КДА;
2) Съществува **полиномиален** алгоритъм който по даден регулярен израз α дава НКА M такъв че $L(M)=L[\alpha]$;
3) Съществува **експоненциален** алгоритъм който по даден НКА M дава регулярен израз α такъв че $L(M)=L[\alpha]$;
4) Съществува **полиномиален** алгоритъм който по даден КДА дава минимален КДА еквивалентен на първия;
5) Съществува **полиномиален** алгоритъм който по дадени два КДА дава отговор дали са еквивалентни или не;
6) Съществува **експоненциален** алгоритъм който по дадени два НКА дава отговор дали са еквивалентни или не;

Изоморфни автомати

Казваме че два $M_1=(K_1,\Sigma,\delta_1,s_1,F_1)$ и $M_2=(K_2,\Sigma,\delta_2,s_2,F_2)$ са изоморфни (неразличими) ако съществува биекция $f:K_1\rightarrow K_2$ такава че $f(s_1)=s_2$ $f[F_1]=F_2$ и $f(\delta_1(q,a))= \delta_1(f(q),a)$.

Крайни автомати като алгоритми

Твърдение

Ако $M=(K,\Sigma,\Delta,s,F)$ е НКА, то съществува алгоритъм който по дадена дума $w\in\Sigma^*$ проверява дали $w\in L(M)$ или не със сложност: $O(|w|.|k|^2)$.

Сложност: полиномиална $O(|\Sigma|.|w|)$

Контекстно-свободни езици

Контекстно-свободна граматика

КСГ е четворка $G=(V,\Sigma,R,S)$, където:

V е фиксирана крайна азбука от символи;

$\Sigma\subseteq V$ – елементите на Σ се наричат терминални символи или терминали;

Нетерминали- елементите на $V\setminus\Sigma$

R- множеството на правила $R\subseteq(V\setminus\Sigma)\mathrm{x}V^*$

S-начален символ $S\in V\setminus\Sigma$.

Правило

Ако $(A,u)\in R$, то A е нетерминал, а $u\in V^*$ и правилото (A,u) ще го озн. $A\rightarrow_G u$ (правило от G).

$=>_G$ Правило в граматика G

Нека $G=(V,\Sigma,R,S)$ е КСГ. Тогава пишем $u =>_G v$, т.т.к. $u,v\in V^*$ и съществуват $x,y\in V^*$ и $A\in V\setminus\Sigma$ и правилото $A\rightarrow v'$ такава че $u=xAy$, $v=xv'y$

Рефлективно и транзитивно затваряне на $=>_G$

$=>_G^*$ се определя като рефлективно и транзитивно затваряне на $=>_G$

L(G) за контекстно-свободна граматика G

Нека $G=(V,\Sigma,R,S)$ е КСГ. Тогава с L(G) означаваме множеството $L(G)=\{w|\ w\in\Sigma^* \text{ и } S=>_G^* w\}$ и се нарича език породен от КСГ G.

Контекстно свободен език

Един език L се нарича КСЕ т.т.к. $L=L(G)$ за някоя КСГ $G=(V,\Sigma,R,S)$.

Извод

Всяка редица $w_0=>_G w_1=>_G\ldots=>_G w_n$ се нарича извод в КСГ G на думата w_n от думата w_0, а n се нарича дължина на извода.

Твърдение за КДА и КСГ

Нека $M=(K,\Sigma,\delta,s,F)$ е КДА. Тогава съществува КСГ $G=(V,\Sigma,R,S)$ такава че $L(M)=L(G)$.

Теория за 2ра част

Лекция 6

Граматични дървета

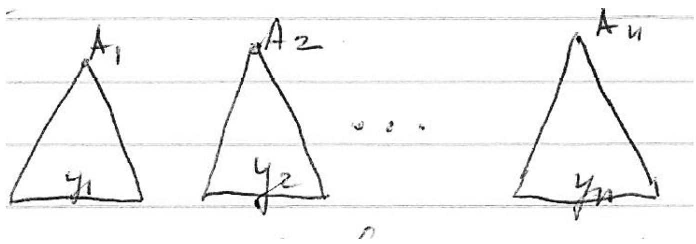
Граматично дърво

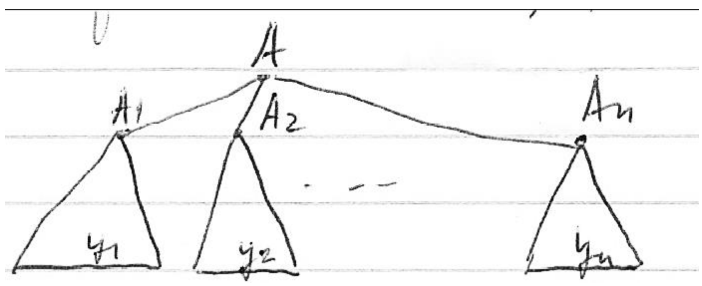
Нека G=(V,Σ,R,S) е произволна КЦГ. Индуктивно определяме граматично дърво:

1) За всяко $a \in \Sigma$,  a е граматично дърво с корен a и листо a (тривиално дърво);

2) Ако $A \to \varepsilon$ е правило е G където $A \in V \setminus \Sigma$ то 

е граматично дърво с корен A и листо ε.

3) Ако 

са граматични дървета с корени $A_1, A_2 \ldots A_n$ съответно и $y_1 \ldots y_n$ са думи от азбуката Σ получени съответно от дърветата с корени $A_1, \ldots A_n$ съответно и $A \to A_1 A_2 \ldots A_n$ е правило в граматиката G то 

е граматично дърво.

Извод предхожда извод

Нека G=(V,Σ,R,S) е КСГ и нека $D = x_1 \Rightarrow x_2 \ldots \Rightarrow x_n$ $D' = x'_1 \Rightarrow x'_2 \ldots \Rightarrow x'_n$ са два извода в G, където $x_i, x'_i \in V^*$ $i = 1 \ldots n$, $x_1, x'_1 \in V \setminus \Sigma$ и $x_n, x'_n \in \Sigma^*$. Казваме че D предхожда D′ и пишем D D′ ако n>2 и $\prec$ съществува естествено число k , 1<k<n такова че са изпълнени следните 5 условия:

1) За всяко i, $1 \le i \le n$, $i \ne k$ имаме $x_i = x'_I$;

2) $x_{k-1} = x'_{k-1} = uAvBw$ където $u,v,w \in V^*$ и $A,B \in V \setminus \Sigma$;

3) $x_k = uyvBw$, където $A \to y$ е правило от G;

4) $x'_k = uAvzw$, където $B \to z$ е правило от G;

5) $x_{k+1} = x'_{k+1} = uyvzw$;

С други думи един извод предхожда друг ако в извода те са различават само по едно място и в този извод който предхожда другия по-напред заместваме нетерминала който е по-вляво.

Подобни изводи (релация на еквивалентност на изводи)

Нека D и D′ са два извода на една и съща дума в граматиката G. Казваме че те са подобни ако (D, D′) принадлежи на рефлексивното, симетричното и транзитивно затваряне на релацията $\prec$.

Следователно релацията за подобни изрази е релация на еквивалентност.

Твърдение за граматично дърво

Нека $G=(V,\Sigma,R,S)$ е КСГ и $A \in V \setminus \Sigma$, а $w \in \Sigma^*$. Тогава следните твърдения са еквивалентни:

1) $A \Rightarrow^* w$;

2) Съществува граматично дърво с корен A и извод w.

3) Съществува най-ляв извод $A \overset{L}{\Rightarrow}{}^* w$;

4) Съществува най-десен извод $A \overset{R}{\Rightarrow}{}^* w$;

Стекови автомати

Стеков автомат

Стеков автомат се нарича шестицата $M=(K,\Sigma,\Gamma,\Delta,s,F)$, където:

K- крайна азбука от състояния;
Σ-крайна входна (основна) азбука;
Γ-крайно множество от стекови символи (стекова азбука);
$F \subseteq K$,F- множество от заключителните състояния;
s-начален символ, $s \in K$;
$\Delta \subseteq (K \times (\Sigma \cup \{\varepsilon\}) \times \Gamma^*) \times (K \times \Gamma^*)$- релация на преходите като всеки елемент от Δ се нарича правило за преход от Δ(от M).

Конфигурация на стеков автомат

Конфигурация се нарича всеки елемент на $K \times \Sigma^* \times F^*$. Ако $(p,x,\alpha) \in K \times \Sigma^* \times F^*$ то $p \in K$, $x \in \Sigma^*$ и $\alpha \in F^*$.

Всичко това се отнася винаги за конкретен стеков автомат $M=(K,\Sigma,\Gamma,\Delta,s,F)$ като p ще си мислим като състоянието в което се намира M, x е непрочетена част от думата с която се запълва лентата от самото начало, а α е думата, с която е запълнен стека.

$|-_M$ за стеков автомат M (едностъпково преобразувание)

Нека $M=(K,\Sigma,\Gamma,\Delta,s,F)$ е стеков автомат и (p,x,α), $(q,y,\beta) \in K \times \Sigma^* \times F^*$ т.е. те са конфигурации. Казваме че от (p,x,α) за една стъпка се извежда (q,y,β) и означаваме $(p,x,\alpha) |-_M (q,y,\beta)$ точно тогава когато съществува правило $((p,x,\alpha), (q,y,\beta)) \in \Delta$, такова че $x=ay$, $\alpha=\lambda\eta$ и $\beta=\mu\eta$ за някое $\eta \in \Gamma^*$.

С $|-_M{}^*$ озн. рефлексивното и транзитивно затваряне на релацията $|-_M$ в множеството на конфигурациите.

Стеков автомат разпознава дума

Казваме че стековият автомат $M=(K,\Sigma,\Gamma,\Delta,s,F)$ разпознава думата $w \in \Sigma^*$ т.т.к.$(s,w,\varepsilon) |-_M{}^*(p,\varepsilon,\varepsilon)$ за някое $p \in F$. Всички конфигурации от вида $(p,\varepsilon,\varepsilon)$ такива че $p \in F$ се наричат още заключителни.

M разпознава думата $w \in \Sigma^*$ т.т.к. съществува редицата от конфигурации $C_0,C_1,\ldots,C_n$ $(n>0)$ такава че $C_0 |-_M C_1 |-_M \ldots |-_M C_n$ където $C_0=(s,w,\varepsilon)$ и C_n е заключителна конфигурация.

Теорема за класът на езиците на стеков автомат

Класът на езиците, разпознавани от стеков автомат съзвпада с класър на КСЕ.

Език разпознаван от стеков автомат.

Един език L се разпознава от стеков автомат ако съществува стеков автомат M такъв че $L=L(M)$.

Твърдение 1 за КСЕ и стеков автомат

Всеки контекстно свободен език се разпознава от стеков автомат.

Лема за най-ляв извод в G

Нека $w \in \Sigma^*$ и $\alpha \in ((V\setminus\Sigma)V^*) \cup \{\varepsilon\}$. Тогава $s \ {}^{L}\!\!=>_G^* \ w\alpha$ т.т.к.
$(q,w,s) |\!-_M^* (q,\varepsilon,\alpha)$.

${}^{L}\!\!=>_G^*$ озн. най-ляв извод в G.

Лекция 7

Твърдение 2 за КСЕ и стеков автомат

Ако един език се разпознава от стеков автомат то този език е КСЕ.

Прост стеков автомат

Нека $M=(K,\Sigma,\Gamma,\Delta,s,F)$ е стеков автомат. Казваме че M е прост ако за всяко правило от $\Delta((q,a,\beta),(p,\gamma))$ такова че $q \neq s$ е изпълнено че $\beta \in \Gamma$ и $|\gamma| \leq 2$.

Лема за прост стеков автомат

Ако един език се разпознава от стеков автомат то той се разпознава и от прост стеков автомат.

Помощно твърдение

За произволни $p,q \in K'$ $A \in \Gamma \cup \{\varepsilon\}$ е изпълнена еквивалентността:
$<q,A,p> =>_G^* x$, т.т.к. $(q,x,A) |\!-_M^* (p,\varepsilon,\varepsilon)$.

Свойства на КСЕ. Лема за разрастването за КСЕ

Твърдение за КСЕ и операциите: обединение, конкатенация и звезда на Клини.

КСЕ са затворени относно обединение, конкатенация и звезда на Клини.

Твърдение за КСЕ и регулярните езици

Сечението на КСЕ и регулярните езици е КСЕ.

Най-голям брой букви на дясната част на правило

Нека $G=(V,\Sigma,R,S)$ е КСГ. С $\Phi(G)$ ще означаваме най-големият брой букви на дясната част на правило от граматиката G.

С други думи разглеждаме всички правила от вида $A\rightarrow x$, където A е нетерминал а $x \in V^*$. Намираме максимума на дължините $|x|$ на думите от дясната част на правилата . Това определя $\Phi(G)$.

Лема за разрастването на граматични дървета

Нека $G=(V,\Sigma,R,S)$ е КСГ. Тогава за всяка дума $w\in L(G)$ чиято дължина $|w|$ е по-голяма от $\Phi(G)^{|V\setminus\Sigma|}$ може да се представи във вида $w=uvxyz$ така че $vy\neq\varepsilon$ и $u.v^n.x.y^n.z\in L(G)$ за всяко естествено число n.

Лекция 8

Нека $L\subseteq\Sigma^*$, L е КСЕ. Тогава съществува ест. число n, такова че за всяка дума $w\in L$, $|w|\geq n$ съществуват думи u,v,x,y,z, такива че $w=uvxyz$ и $|vxy|\leq n$, $vy\neq\varepsilon$ и за всички $i\in\mathbb{N}$ е изпълнено $uv^ixy^iz\in L$.

Твърдение за КСЕ и операциите сечение и допълнение

Множеството на КСЕ не е затворено относно операциите сечение и допълнение.

КСГ в нормална форма на Чомски

Казваме че една КСГ $G=(V,\Sigma,R,S)$ се намира в нормална форма на Чомски ако $R\subseteq(V\setminus\Sigma)xV^2$, т.е. всички правила са от вида $A\rightarrow BC$ където $A\in V\setminus\Sigma$ а $B,C\in V$.

Теорема за нормална форма на Чомски

За всяка КСГ G съществува КСГ G′ в нормална форма на Чомски таквава че $L(G')=L(G)\setminus(\Sigma\cup\{\varepsilon\})$.

Теорема за алгоритми за КСГ

Един стеков автомат M е еквивалентен на КСГ G, т.т.к. $L(M)=L(G)$.

1) Съществува ***полиномиален*** алгоритъм който по дадена контексно свободна граматика дава стеков автомат еквивалентен на граматиката.($8|G|^2$)

2) Съществува **полиномиален** алгоритъм който по даден стеков автомат дава КСГ еквивалентна на стековия автомат.($O(|M|^3)$)

3) Съществува **полиномиален** алгоритъм който по дадена КСГ G и дума x дава отговор дали $x\in L(G)$. ($O(|x|^3)$)

Лекция 9

Машина на Тюринг

Машина на Тюринг

Машина на Тюринг се нарича петицата $M=(K,\Sigma,\delta,s,H)$ където:

K- крайна азбука на състоянията;

Σ- основна азбука, съдържаща символа $\sqcup$ за празната клетка и символа $\rhd$ за ляв ограничител, но не съдържа символите $\rightarrow$ и $\leftarrow$;

$s\in K$, s- начално състояние (символ);

$H\subseteq K$, H- множество на стоп-състоянията;

δ- функция на преходите, $\delta:(K\setminus H)\times\Sigma\rightarrow K\times(\Sigma\cup\{\rightarrow,\leftarrow\})$, така че следните условия:

1) За всички $q\in(K\setminus H)$ ако $\delta(q,\rhd)=(p,b)$ то $b=\rightarrow$;
2) За всички $q\in(K\setminus H)$ и $a\in\Sigma\setminus\{\rhd\}$ ако $\delta(q, a)=(p,b)$ то $b\neq\rhd$.

Конфигурация на машина на Тюринг

Конфигурация на машина на Тюринг $M=(K,\Sigma,\delta,s,H)$ се нарича всеки елемент на $K\times(\Sigma\setminus\{\rhd\})^*\times((\Sigma\setminus\{\rhd\})^*\ (\Sigma\setminus\{\rhd,\sqcup\})\cup\{\varepsilon\})$ т.е. конфигурация е тройката $(p, \rhd u, v)$ такава че $p\in K$ и думите $u,v\in(\Sigma\setminus\{\rhd\})^*$, като v не може да завършва с $\sqcup$.

$\vdash_M$ за Машина на Тюринг M

Нека $M=(K,\Sigma,\delta,s,H)$ е машина на Тюринг и да разгледаме конфигурациите $(q_1,w_1a_1u_1)$ и $(q_2,w_2a_2u_2)$ където $a_1,a_2\in\Sigma$. Тогава казваме че машината M преобразува $(q_1,w_1a_1u_1)$ в $(q_2,w_2a_2u_2)$ и пишем

$(q_1,w_1a_1u_1) \vdash_M (q_2,w_2a_2u_2)$ т.т.к за някое $b\in \Sigma\cup\{\rightarrow,\leftarrow\}\delta(q_1,q_2)=(q_2,b)$ и, или:

1) $b\in\Sigma\setminus\{\rhd\}$, $w_1=w_2$, $u_1=u_2$ и $a_2=b$ (заместваме a_1 с b) или
2) $b=\leftarrow$, $w_1=w_2.a_2$ и, или
   a. $u_2=a_1u_1$ ако $a_1\neq\sqcup$ или $u_1\neq\varepsilon$, или
   b. $u_2=\varepsilon$ ако $a_1=\sqcup$ или $u_1=\varepsilon$ (преместваме главата наляво), или

3) $b=\rightarrow$, $w_2=w_1.a_1$ и или

   a. $u_1=a_2u_2$ или

   b. $u_1=u_2=\varepsilon$ и $a_2=\sqcup$ (преместваме главата надясно).

Рефлексивно и транзитивно затваряне на релацията $|\text{-}_M$

Нека M е машина на Тюринг. С $|\text{-}_M{}^*$ озн. рефлексивно и транзитивно затваряне на релацията $|\text{-}_M$ между конфигурациите на M.

Базови машини на Тюринг

1. Пишат някакъв символ от азбуката и спират

Нека $a\in\Sigma\setminus\{\rhd\}$. С Ma озн. машината $Ma=(\{s,h\},\Sigma,\delta,s,\{h\})$ където за всяко $b\in\Sigma\setminus\{\rhd\}$, $\delta(s,b)=(h,a)$ т.е. машината замества съдържанието на клетката която чете в момента и спира. По подразбиране $\delta(s,\rhd)=(s,\rightarrow)$

2. Машина M← и M→

$M_{\rightarrow}=(\{s,h\},\Sigma,\delta,s,\{h\})$ където за всяко $b\in\Sigma$ $\delta(s,b)=(h,\rightarrow)$ и $M_{\leftarrow}=(\{s,h\},\Sigma,\delta,s,\{h\})$ където за всяко $b\in\Sigma\setminus\{\rhd\}$ $\delta(s,b)=(h,\leftarrow)$ и $\delta(s,\rhd)=(s,\rightarrow)$.

Операции между машини на Тюринг

1. Композиция на Машина на Тюринг

Нека $M_1=(K_1,\Sigma,\delta_1,s_1,H_1)$ и $M_2=(K_2,\Sigma,\delta_2,s_2,H_2)$ като $K_1\cap K_2=\emptyset$. Композиция на M_1 и M_2 ще означаваме с M_1M_2 и $M_1M_2=(K_1\cup K_2,\Sigma,\delta,s_1,H_2)$ като $\delta(q,r)=\delta_1(q,r)$ за $q\in K_1\setminus H_1$, $\delta(q,r)=(s_2,r)$ за $q\in H_1$ и $r\in\Sigma$, $\delta(q,r)=\delta_2(q,r)$ за $q\in K_2\setminus H_2$ и $r\in\Sigma$.

Пускаме M_1 върху тази конфигурация и ако стигнем до стоп-конфигурация заместваме стоп-състоянието с s_2 и пускаме получената конфигурация да се изпълни от M_2.

2. R-машина

Машина която каквото и да прочете прави стъпка в дясно и спира.

3. L-машина

Машина която каквото и да прочете освен знака за най-ляв символ ($\rhd$) прави стъпка в ляво и спира.

4. R⊔

$R^{\circlearrowleft a \neq \sqcup} \Leftrightarrow R^{\circlearrowleft \sqcup}$ - върви надясно, докато срещнеш ⊔ (празната дума)

$\delta(p,\sigma) = h,\ \sigma \in \Sigma\setminus$

$R_\sqcup^2 = R_\sqcup R_\sqcup$

Машини

1. **Какво и в какво преобразува машината (Копи-машината) C на Тюринг?**

   C трансформира $\rhd \underline{\sqcup} w$ в $\rhd \underline{\sqcup} w \sqcup w$, за $w \in (\Sigma \setminus \{\rhd,\sqcup\})^*$

2. **Какво и в какво преобразува машината (Шифт-машината) S_→ на Тюринг?**

   $S_{\rightarrow}$трансформира $\rhd \underline{\sqcup} w$ в $\rhd \sqcup \underline{\sqcup} w$, за w∈(Σ\{⊳,⊔})*

3. **Какво и в какво преобразува простата машина на Тюринг L_⊔?**

   $L_\sqcup$ трансформира $\rhd \sqcup abba \underline{\sqcup}$ в $\rhd \underline{\sqcup} abba$

4. **Какво и в какво преобразува простата машина на Тюринг R_⊔?**

   $R_\sqcup$ трансформира $\rhd \underline{\sqcup} abba$ в $\rhd \sqcup abba \underline{\sqcup}$

Лекция 10

Разпознаване и изчисляване с помощта на машина на Тюринг

Машина на Тюринг разпознаване на дума

Нека M=(K,Σ,δ,s,H) е машина на Тюринг където H={y,n}. Казваме че думата w∈(Σ\{⊳,⊔})* се приема от M ако $(s, \rhd \underline{\sqcup} w) \vdash_M^* (y, u\underline{a}v)$ и че w се отхвърля от M ако $(s, \rhd \underline{\sqcup} w) \vdash_M^* (u, u\underline{a}v)$ за всяко Σ_0 такова че $\Sigma_0 \subseteq \Sigma\setminus\{\rhd,\mathbf{w}\}$. Σ_0 се нарича входна азбука.

Машина на Тюринг разпознава език

Нека $M = \langle K, \Sigma, \delta, s, H\rangle$ е машина на Тюринг, където H={y,n}, и $\Sigma_0 \subseteq \Sigma \setminus \{\rhd,\sqcup\}$, Σ_0 – входна азбука. Казваме, че езикът $L \subseteq \Sigma_0{}^*$ се разпознава от M, ако за всяко $w \in \Sigma_0{}^*$ са изпълнени следните две условия:

1) Ако $w \in L$, то w се разпознава от M

2) Ако $w \notin L$, то w се отхвърля от M

Рекурсивен език

Един език L се нарича рекурсивен(разширим) ако съществува машина на Тюринг M, която го разширява.

M изчислява функция f

Нека $M = \langle K, \Sigma, \delta, s, H \rangle$ е машина на Тюринг и $\Sigma_0 \subseteq \Sigma \setminus \{\rhd, \sqcup\}$. Казваме че функцията $f:\Sigma_0^* \rightarrow \Sigma_0^*$ се изчислява с помощта на M (M изчислява f) ако за произволно $w \in \Sigma_0^*$ са изпълнени следните условия:

1) Ако $f(w)=y$ за $y \in \Sigma_0^*$, то $(s, \rhd \underline{\sqcup} w) \vdash_M^* (h, \rhd \underline{\sqcup} y)$;

2) Ако $(s, \rhd \underline{\sqcup} w) \vdash_M^* (h, \rhd \underline{\sqcup} y)$ за $y \in \Sigma_0^*$, то $f(w)=y$.

Рекурсивни и рекурсивно наредени езици

Полурезрешен език

Нека $M = \langle K, \Sigma, \delta, s, H \rangle$ е машина на Тюринг и $\Sigma_0 \subseteq \Sigma \setminus \{\rhd, \sqcup\}$. Казваме че M полуразрешава езика L, $L \subseteq \Sigma_0^*$, т.т.к за всяко $w \in \Sigma_0^*$ следната еквивалентност е изпълнена:

$w \in L \Leftrightarrow$ M спира работа върху w, т.е. $M(w) \searrow$.

Рекурсивно номеруем език

Един език L се нарича рекурсивно номеруем(полуразрешим), т.т.к. съществува машина на Тюринг M такава че M полуразрешава L.

Твърдения за разрешимите и полуразрешимите езици

1) Всеки рекурсивен език е рекурсивно номеруем.

2) Ако L е рекурсивен език, то $\overline{L}$ също е рекурсивен.

3) Съществува рекурсивно номеруем език, който не е рекурсивен.

Примитивни рекурсивни функции(ПРФ). Примитивна рекурсивност на някои функции

Суперпозиция и примитивна рекурсия

Нека f: $\mathbb{N}^{\boldsymbol{k}} \nrightarrow \mathbb{N}$ и g_i: $\mathbb{N}^{\boldsymbol{k}} \nrightarrow \mathbb{N}$ i=1,…,n .Казваме че функцията h се получава от $f,g_1,\ldots g_n$ с помощта на операцията суперпозиция ако h: $\mathbb{N}^{\boldsymbol{k}} \nrightarrow \mathbb{N}$ и за всички естествени числа $x_1,\ldots,x_n$ е изпълнено равенството $h(x_1,\ldots,x_n)=f(g_1(x_1,\ldots,x_k),\ldots, g_n(x_1,\ldots,x_k))$.

Операцията примитивна рекурсия:

Нека $f : \mathbb{N} \nrightarrow \mathbb{N}$ и $g : \mathbb{N}^3 \nrightarrow \mathbb{N}$. Казваме, че функцията $h: \mathbb{N}^2 \nrightarrow \mathbb{N}$ се получава от f и g с помощта на операцията примитивна рекурсия, ако за произволни $x, y \in \mathbb{N}$ е изпълнена системата

$$\begin{cases} h(x,0) = f(x) \\ h(x,y+1) = g\big(x,y,h(x,y)\big) \end{cases}.$$

Кои функции са ПРФ?

1) Всички основни функции $O,s,I_i^{\,n}$, $1 \leq i \leq n$ са ПРФ;
2) Ако $f,g_1,\ldots g_n$ са ПРФ то и тяхната суперпозиция която стандартно означаваме с $h= f(g_1,\ldots g_n)$ също е ПРФ;
3) Ако f,g са ПРФ то и функцията h която се получава от тях с помощта на ПР също е ПРФ.

Твърдение за ПРФ

Следните функции са ПРФ:

1) Всяка константа $Ca(x)=a$ за $x \in \mathbb{N}$;
2) $x+y$
3) xy
4) x^y
5) $P(x) \begin{cases} x-1, \text{ ако } x > 0 \\ 0, \text{ ако } x = 0 \end{cases}$
6) $\Pi(x,y)=2^x(2y+1)-1$.

Проста схема на ПР

Нека $g:\mathbb{N}^2 \nrightarrow \mathbb{N}$ и $a \in \mathbb{N}$. Казваме че $h:\mathbb{N} \to \mathbb{N}$ се получава от a, g с помощта на проста схема на ПР, ако h(0)=a и h(x+1)=g(K,h(x)) за $x \in \mathbb{N}$.

Лема за проста схема на ПР

Ако $g:\mathbb{N}^2 \to \mathbb{N}$ и $a \in \mathbb{N}$ е ПРФ то и функцията h която се получава от a,g с помощта на проста схема на ПР също е ПРФ.

Кодът на $<x_0,\ldots,x_k>$

Нека $<x_0,\ldots,x_k>$ е крайна редица от естествени числа. Кодът на $<x_0,\ldots,x_k>$ също ще озн. с $\tau(< x_0, \ldots, x_k >)$ и $q(<x_0,\ldots,x_k>)=\Pi(k,\Pi_{k+1}(x_0,\ldots,x_k))$.

Изчислимост на ПРФ

Кога една машина на Тюринг изчислява една функция $\boldsymbol{F} : \mathbb{N}^{\boldsymbol{k}} \to \mathbb{N}$ на $\boldsymbol{k}$ променливи?

Нека $F: \mathbb{N}^k \to \mathbb{N}$. Казваме, че машината на Тюринг $M = \langle K, \Sigma, \delta, s, H \rangle$ изчислява F, ако за произволни естествени числа $n_1, -, n_k$ са изпълнени условията:

1) Ако $F(n_1, -, n_k) = k$, то $M(1^{n_1} \sqcup 1^{n_2} \sqcup - \sqcup 1^{n_k}) \searrow$ и $M(1^{n_1} \sqcup 1^{n_2} \sqcup - \sqcup 1^{n_k}) = 1^n$.
2) Ако $M(1^{n_1} \sqcup 1^{n_2} \sqcup - \sqcup 1^{n_k}) \searrow$, то $F(n_1, -, n_k)$ е дефинирана.

Твърдение за машина на Тюринг и ПРФ

1) Всички основни ПРФ са изчислими с машина на Тюринг.

Лекция 11

2) Операцията суперпозиция запазва изчислимостта с помощта на машини на Тюринг.

3) Операцията примитивна рекурсия запазва изчислимостта с помощта на машини на Тюринг.

Теорема за ПРФ и машини на Тюринг

Всички примитивни рекурсивни функции са изчислими с помощта на машини на Тюринг.

Частично рекурсивна функция

Операцията минимизация

Нека $f : \mathbb{N}^{k+1} \to \mathbb{N}$ е частично определена функция. Казваме, че функцията $g: \mathbb{N}^k \to \mathbb{N}$ се получава от f с помощта на операцията минимизация (още μ-операция), ако за произволни $x_1, -, x_k, y \in \mathbb{N}$ е изпълнена еквивалентността:

$g(x_1, -, x_k) = y \Leftrightarrow$ са изпълнени следните две условия:

1. $f(x_1, -, x_k, y) = 0$
2. $\forall z < y$ е дефинирана $f(x_1, -, x_k, z)$ и $f(x_1, -, x_k, z) > 0$.

Когато g се получава от f с помощта на μ-операция(минимилизазия) това се записва така:

$g(x_1,\ldots,x_n)=\mu y[f(x_1,\ldots,x_n,y)=0]$.

Индуктивна дефиниция за частична рекурсивна функция(ЧРГ)

1) Всички основни ПРФ- $O,S,I_i^{\,n}$ са ЧРФ;
2) Ако $f,g_1,\ldots g_n$ са ЧРФ то и тяхната суперпозиция също е ЧРФ;
3) Ако f,g са ЧРФ то и функцията която се получава от тях с помощта на ПР също е ЧРФ.
4) Ако f е ЧРФ то и функцията g, която се получава от тях с помощта на μ-операция, също е ЧРФ.

Твърдение за операцията минимизация и машини на Тюринг

Операцията минимизация запазва изчислимостта с помощта на машини на Тюринг.

Теорема за ЧРФ и машини на Тюринг

Всички частично рекурсивни функции са изчислими с помощта на машини на Тюринг.

Различни разширения на машини на Тюринг

Теорема за машини на Тюринг

Всеки език който е рекурсивен, полуразрешим и всяка функция която е изчислима с различни видове споменати машини на Тюринг, е рекурсивен, полурарешим и съответно изчислима функция с помощта на стандартна машина на Тюринг.

Недетерминистична машина на Тюринг(НМТ)

Недетерминистична машина на Тюринг е петицата M=(K,Σ,Δ,s,H) където всичко останало без Δ е както при стандартна машина на Тюринг а вместо δ-функция на преходите при стандартна машина на Тюринг, имаме Δ която е релация на преходите $\Delta \subseteq ((K\backslash H)x\Sigma)x(Kx(\Sigma \cup \{\rightarrow,\leftarrow\}))$

НМТ приема дума

Нека M=(K,Σ,Δ,s,H) е НМТ. Казваме че M приема $w \in (\Sigma\backslash\{\rhd,\sqcup\})^*$ т.т.к

1) $(s,\rhd \underline{\sqcup}w) \vdash_M^* (h,u\underline{a}v)$ за някое $h \in H$ и $u,v \in \Sigma^*$.

НМТ полуразрешава език

Нека M=(K,Σ,Δ,s,H) е НМТ . Казваме че M полуразрешава езика $L \subseteq (\Sigma\backslash\{\rhd,\sqcup\})^*$ т.т.к. за всички $w \in (\Sigma\backslash\{\rhd,\sqcup\})^*$ е изпълнена еквивалентността:

$w \in L \Leftrightarrow$ M приема w.

Полуразрешим език

Един език L се нарича полуразрешим т.т.к. съществува НМТ M такава че M полуразрешава L.

НМТ разрешава език

Нека M=(K,Σ,Δ,s,H) е НМТ . Казваме че M разрешава езика $L \subseteq (\Sigma\backslash\{\rhd,\sqcup\})^*$ т.т.к. H={y,u} и за всички $w \in (\Sigma\backslash\{\rhd,\sqcup\})^*$ са изпълнени следните две условия:

1. Съществува ест. число n зависещо от M и w, такова че не съществува конфигурация C удовлетворяваща $(s,\rhd \underline{\sqcup}w) \vdash_M^n C$

2) $w \in L \Leftrightarrow (s,\rhd \underline{\sqcup}w) \vdash_M^* (y,u\underline{a}v)$ за някои $u,v \in \Sigma^*$ и $a \in \Sigma$.

Функция е изчислима с помощта на НМТ

Нека M=(K,Σ,Δ,s,H) е НМТ . Казваме че функцията f:(Σ\{⊳,⊔})*→(Σ\{⊳,⊔})* е изчислима с помощта на M, т.т.к. следните две условия са изпълнени за произволно w∈(Σ\{⊳,⊔})*:

1. Съществува ест. число n зависещо от M и н, такова че не съществува конфигурация C удовлетворяваща $(s,\rhd \underline{\sqcup}w) \vdash_M^n C$
2. $w\in L \Leftrightarrow (s,\rhd \underline{\sqcup}w) \vdash_M^* (y,u\underline{a}v)$ т.т.к $u\underline{a}=\rhd\underline{\sqcup}$ и $v=f(w)$.

Тезис на Чърг-Тюринг. Кодиране на машини на Тюринг. Универсална машина на Тюринг

Твърдение за H

H не е рекурсивно множество.

Твърдение за Класът на рекурсивно номеруемите множества

Класът на рекурсивно номеруемите множества не е затворен относно операцията допълнение.

Допълнителни свойства на рекурсивни и рекурсивно номеруеми езици.

Твърдение за рекурсивен език

Един език L е рекурсивен т.т.к. L и $\bar{L}$ са рекурсивно нмеруеми.

Машина на Тюринг номерира език

Казваме че една машина на Тюринг номерира (извежда) езика L т.т.к. за някое фиксирано q от M $L=\{w|(s,\rhd \underline{\sqcup}) \vdash_M^*(q,\rhd \underline{\sqcup}w)\}$.

Език е Тюринг-номеруем

Един език L е Тюринг-номеруем т.т.к. съществува машина на Тюринг която номерира L.

Твърдение за Тюринг-номеруем език

Един език L е рекурсивен т.т.к. L е Тюринг-номеруем.

M лексикографски номерира език

Нека M е машина на Тюринг. Казваме че M лексикографски номерира език L ако следни неща са в сила:

Съществува „специално“ състояние q такова че когато $(q,\rhd \underline{\sqcup}w) \vdash_M^* (q,\rhd \underline{\sqcup}w')$ то w' е следващия член на L след w.

Лексикографски Тюринг-номеруем език

Казваме че L Лексикографски Тюринг-номеруем език т.т.к. съществува машина на Тюринг M която лексикографски номерира език L.

Твърдение за лексикографски номеруем език

Един език L е рекурсивен т.т.к. L е лексикографски номеруем.

Лекция 13

Неразрешими проблеми за машини на Тюринг

Редукция на език към език

Нека $L_1, L_2 \in \Sigma^*$ са езици. Свеждане L_1 към L_2 се нарича рекурсивната функция $\tau : \Sigma^* \rightarrow \Sigma^*$ такава че за всяко $x \in \Sigma^*$ е изпълнена еквивалентността:

$x \in L_1 \Leftrightarrow \tau(x) \in L_2$'

Твърдение

Ако L_1 не е рекурсивен и има редукция от L_1 към L_2 то L_2 не е рекурсивен.

Теоремата за неразрешимите проблеми на машина на Тюринг

Следните проблеми за машини на Тюринг са неразрешими:

1) **Стоп-проблем:** По дадена машина на Тюринг M и вход w дали M спира върху входа w.
2) **Проблем за празния вход:** По дадена машина на Тюринг M, дали M спира върху празния вход.
3) **Проблема за съществуването на вход:** По дадена машина на Тюринг M дали M спира върху някой вход.
4) **Проблема за тотална машина на Тюринг:** По дадена машина на Тюринг M дали M спира върху всеки вход.
5) **Проблема за еквивалентния стоп:** По дадени машини на Тюринг M_1 и M_2 дали M_1 и M_2 спират върху едни и същи входове
6) **Вид на езика, който полуразпознава:** По дадена машина на Тюринг M дали езикът, който M полуразпознава, е регулярен/КСЕ/рекурсивен.
7) **Проблем с входа на универсална машина на Тюринг:** Съществува машина на Тюринг M(фиксирана) такава че по даден вход w не е разрешимо дали M ще спре работа върху w.

Теорема за неразрешими проблеми

Следните проблеми са неразрешими:

1) По дадена КСГ G дали $L(G)=\Sigma^*$;
2) По дадени 2 КСГ G_1 и G_2 дали $L(G_1)=L(G_2)$;
3) По дадени 2 стекови автомата M_1 и M_2 дали $L(M_1)=L(M_2)$;
4) По даден стеков автомат M да се намери стеков автомат еквивалентен на M който имам най-малко състояния.

Лекция 14

Класовете P и NP

Полиномиално ограничена машина на Тюринг

Нека $M=(K,\Sigma,\Delta,s,H)$ е машина на Тюринг. Казваме че M е полиномиално ограничена т.т.к. съществува полином $p(n)$ такъв че за всеки вход x не съществува конфигурация C такава че

$(s,\rhd\underline{\sqcup}x)\vdash_M^{p(|x|)+1} C$.

С други думи машината M върху входа x ще стигне до стоп-конфигурация най-много след $p(|x|)$ стъпки

Полиномиално разширим език

Един език L се нарича полиномиално разширим ако съществува полиномиално ограничена машина на Тюринг M която разпознава L.

P

С P озн. класът на всички езици които са полиномиално разширими:

$P=\{L|L$ е полиномиални разширим$\}$

Полиномиално ограничена НМТ

Нека $M=(K,\Sigma,\Delta,s,H)$ е НМТ. Казваме че M е полиномиално ограничена т.т.к. съществува полином $p(x)$ такъв че за всеки вход x не съществува конфигурация C за M такава че $(s,\rhd\underline{\sqcup}x)\vdash_M^{p(|x|)+1} C$.

NP

Класът NP се определя като класът на всички езици които се разпознават от полиномиално ограничена НМТ.

$NP=\{L|L$ се разпознава от полиномиално ограничени НМТ$\}$