ДИСКРЕТНИ СТРУКТУРИ II
ТЕОРИЯ II

КОНТЕКСТНО-СВОБОДНА ГРАМАТИКА

17. Контекстно-свободна граматика (КСГ).
    КСГ наричаме наредената четворка $G = \langle \Sigma, V, S, R \rangle$, където
    - Σ е крайно множество от терминални символи (азбука на граматиката);
    - V е крайно множество от нетерминални символи (променливи) и $V \cap \Sigma = \emptyset$;
    - $S \in V$ е начален нетерминал;
    - $R \subseteq V \times (\Sigma \times V)^*$ е крайно множество от правила, които преобразуват нетерминален символ в последователност от други символи.

18. Релацията $\Rightarrow_G$ за дадена контекстно-свободна граматика (КСГ) G.
    Нека $G = \langle \Sigma, V, S, R \rangle$ е КСГ. За всело две думи $\alpha_1, \alpha_2 \in (\Sigma \cup V)^*$ ще казваме, че са в релацията $\Rightarrow_G$ и ще бележим с $\alpha_1 \Rightarrow_G \alpha_2$ тогава и само тогава, когато съществуват други две думи $\beta_1, \beta_2 \in (\Sigma, V)^*$ и нетерминал $T \in V$, за които е изпълнено $\alpha_1 = \beta_1 T \beta_2$, $\alpha_2 = \beta_1 \gamma \beta_2$ и $T \to \gamma$. Тоест от думата α_1 може да преобразуваме до думата α_2 чрез една стъпка (чрез прилагане точно веднъж на едно правило).

19. Релацията $\Rightarrow_G^*$ за дадена контекстно-свободна граматика (КСГ) G.
    Релацията $\Rightarrow_G^*$ е рефлексивното и транзитивното затваряне на релацията $\Rightarrow_G$. Тоест, ако $\alpha_1 \Rightarrow_G^* \alpha_2$, то от думата α_1 може да отидем в думата α_2 чрез няколко (може и 0) стъпки.

20. Език на граматика.
    Езикът на граматиката $G = \langle \Sigma, V, S, R \rangle$ е $\mathscr{L}(G) = \{\omega \in \Sigma^* \,|\, S \Rightarrow_G^* \omega\}$ или аналогично $\mathscr{L}(G) = \{\omega \in \Sigma^* \,|\, S \overset{l}{\Rightarrow} \omega$, за някое $l \in \mathbb{N}_0\}$.

21. Кога една дума $\omega \in \Sigma^*$ се приема от КСГ $G = \langle \Sigma, V, S, R \rangle$.
    Казваме, че $\omega \in \mathscr{L}(G)$ тогава и само тогава, когато от началния символ S, за краен брой стъпки се извежда ω: $S \overset{l}{\Rightarrow}_G \omega, l \in \mathbb{N}$.

22. Граматика в нормална форма на Чомски.
    КСГ $G = \langle \Sigma, V, S, R \rangle$ е в нормална форма на Чомски (НФЧ), ако всеички правила R са от вида $A \to BC$ или $A \to a$. Казано с други думи, G не може да породи дума с дължина по-малка от 2.

    Така дефинираната граматика няма да е способна да възпроизведе празната дума ε. Следователно констекстно-свободните езици, които съдържат тези думи не могат да се генерират от КСГ в НФЧ. Но това е единствената загуба, която идва с преминаването на дадена КСГ в НФЧ.

23. Теорема за КСГ в НФЧ.
    За всяка КСГ G, съществува КСГ G_C в НФЧ, за която $\mathscr{L}(G_C) = \mathscr{L}(G) \backslash \{\varepsilon\}$. Освен това, консгтруирането на G_C може да се направи с полиномиална сложност по време по отношение на размера на G. Под размер на КСГ G разбираме дължината на думата образувана от конкатенацията на всички десни части на правилата R в G.

24. Лемата за разрастването на граматични дървета (лемата за разрастването $xyzuv$).
Нека $G = \langle \Sigma, V, S, R \rangle$ е КСГ с безкраен език $\mathscr{L}(G) = L$.
СЪЩЕСТВУВА естествено число n, такова че
ЗА ВСЯКА дума $\omega \in L$ с дължина $|\omega| \geq n$,
СЪЩЕСТВУВАТ думи x, y, z, u, v, за които $\omega = xyzuv, yu \neq \varepsilon, |yzu| \leq n$ и
ЗА ВСЯКО естествено число i, думата xy^izu^iv е от L.

СТЕКОВ АВТОМАТ

25. (Недетерминиран) стеков автомат (Push-down automaton PDA).
НСА наричаме наредената седморка $P = \langle Q, \Sigma, \Gamma, \#, q_{\text{start}}, q_{\text{accept}}, \Delta \rangle$, където
   1) Q е крайно множество от състояния;
   2) Σ е крайна входна азбука;
   3) Γ е крайна стекова азбука (тук няма изискване както при граматиките $\Sigma \cap \Gamma = \emptyset$);
   4) # е специален символ за дъно на стека, $\# \notin \Sigma \cup \Gamma$;
   5) q_{start} е началното състояние на автомата;
   6) q_{accept} е финалното състояние (единственото финално състояние не променя изразителната сила на стековия автомат);
   7) Δ е релацията на преходите, която е подмножество на

$$\Big(\underset{\text{текущо състояние}}{Q} \times (\overset{\text{буква за прочитане}}{\Sigma \times \{\varepsilon\}}) \times \underset{\text{текущ връх на стека}}{\Gamma}\Big) \times (\overset{\text{ново състояние}}{Q} \times \underset{\text{нов връх на стека}}{\Gamma^{\leq 2}})$$

Важни неща, които трябва да отбележим за сигнатурата на Δ: буквата за прочитане може да е буква от азбуката или празната буква, но задължително трябва да имаме буква в стека, която да прочетем, за да може да извършим преход.
$\Gamma^{\leq 2} = \Gamma^0 \cup \Gamma^1 \cup \Gamma^2$; $\Gamma^0 = \{\varepsilon\}$ – изтрива върха на стека; $\Gamma^1 = \{a\}$ – заменя върха на стека; $\Gamma^2 = \{ab\}$ – добавя над върха на стека.

26. Релацията $\vdash_P$ за стеков автомат P.
Елементите на $Q \times \Sigma^* \times \Gamma^*$ ще наричаме конфигурация на P, където $P = \langle Q, \Sigma, \Gamma, \#, q_{\text{start}}, q_{\text{accept}}, \Delta \rangle$. Нека (p, u, α) и (q, v, γ) са две конфигурации на P.
Дефинираме релацията $\vdash_P$ по следния начин:
$(p, u, \alpha) \vdash_P (q, v, \gamma) \overset{\text{def.}}{\Leftrightarrow} \exists \big((p, \alpha, \beta), (q, \delta)\big) \in \Delta$, такова че $u = a \cdot v$, $\alpha = \beta \cdot \eta$, $\gamma = \delta \cdot \eta$, за някое $\eta \in \Gamma^*$.

27. Релацията $\vdash_P^*$ за стеков автомат P.
Релацията $\vdash_P^*$ за стеков автомат P е рефлексивното (включително 0 стъпки) и транзитивното затваряне на релацията $\vdash_P$. Тази релация ни казва до каква конфигурация е доведен автомата след краен брой стъпки от дадена начална конфигурация.

28. Език на стеков автомат.
Език на стеков автомат $P = \langle Q, \Sigma, \Gamma, \#, q_{\text{start}}, q_{\text{accept}}, \Delta \rangle$ дефинираме по следния начин: $\mathscr{L}(P) = \{\omega \in \Sigma^* \,|\, (q_{\text{start}}, \omega, \#) \vdash_p^* (q_{\text{accept}}, \varepsilon, \varepsilon)\}$.

29. Кога една дума се приема от стеков автомат.
Стековият автомат $P = \langle Q, \Sigma, \Gamma, \#, q_{\text{start}}, q_{\text{accept}}, \Delta \rangle$ приема думата $\omega \in \Sigma^*$, ако е изпълнено $(q_{\text{start}}, \omega, \#) \vdash_p^* (q_{\text{accept}}, \varepsilon, \varepsilon)$.

30. Прост стеков автомат.
Казваме, че стековият автомат $P = \langle Q, \Sigma, \Gamma, \#, q_{\text{start}}, q_{\text{accept}}, \Delta \rangle$ е прост, ако за всяко правило $\big((q, a, \beta), (p, \gamma)\big) \in \Delta$, такова че $q \neq s$ е изпълнено, че $\beta \in \Gamma$ и $|\gamma| \leq 2$.

МАШИНА НА ТЮРИНГ

31. Машина на Тюринг (МТ).
МТ наричаме наредената петорка $M = \langle K, \Sigma, \delta, s, H \rangle$, където:
- K е крайно множество от състояния;
- Σ е азбука, която съдържа символ за празна клетка $\sqcup$ и символ за ляв ограничител $\rhd$, но не съдържа $\leftarrow$ и $\rightarrow$;
- $s \in K$ е началното състояние;
- $H \in K$ е множество от стоп състояния;
- δ е функцията на преходите: $(K \backslash H) \times \Sigma \rightarrow K \times \big(\Sigma \cup \{\leftarrow, \rightarrow\}\big)$, за която за всяко $q \in K \backslash H$:
  - ако $\delta(q, \rhd) = (p, b)$, то $b = \rightarrow$;
  - $\forall a \in \Sigma$, ако $\delta(q, a) = (p, b)$, то $b \neg \rhd$.

32. Кога една машина на Тюринг разпознава един език.
Езикът L се разпознава от машина на Тюринг $M = \langle K, \Sigma, \delta, s, H \rangle$ s $\underbrace{y, n \in H}_{\text{y=yes, n=no}}$, ако за всяка дума $\omega \in \Sigma_\circ^*$ (азбука с допълнителен символ) е изпълнено:
- ако $\omega \in L$, то M приема ω.
  $(s, \rhd \underline{\sqcup}\omega)$ спира на приемаща конфигурацуя (такава с y);
- ако $\omega \notin L$, то M отхвърля ω.
  $(s, \rhd \underline{\sqcup}\omega)$ спира на отхвърляща конфигурацуя (такава с n);

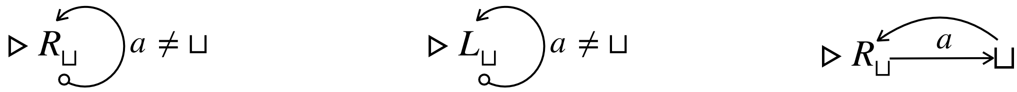
33. Какво и в какво преобразува простата машина на Тюринг $R_\sqcup$.
$\rhd \omega_1 \underline{\sqcup} \omega_2 \rightarrow_{R_\sqcup} \rhd \omega_1 \sqcup \omega_2 \underline{\sqcup}$; $\omega_1, \omega_2 \in \big(\Sigma \backslash \{\rhd, \sqcup\}\big)^*$ - обхожда (сканира) лентата надясно докато не намери символ за празната клетка.

34. Какво и в какво преобразува простата машина на Тютинг $L_\sqcup$
$\rhd \omega_1 \sqcup \omega_2 \underline{\sqcup} \rightarrow_{L_\sqcup} \rhd \omega_1 \underline{\sqcup} \omega_2$; $\omega_1, \omega_2 \in \big(\Sigma \backslash \{\rhd, \sqcup\}\big)^*$ - обхожда (сканира) лентата наляво докато не намери символ за празната клетка.

35. Какво и в какво преобразува машината C на Тюринг (копиращата (copy) машина).
$\sqcup \omega \underline{\sqcup} \xrightarrow{C} \sqcup \omega \sqcup \omega \underline{\sqcup}, \omega \in \big(\Sigma \backslash \{\rhd, \sqcup\}\big)^*$.

36. Какво и в какво преобразува машината $S_{\rightarrow}$ на Тюринг.
$\sqcup\,\omega\,\underline{\sqcup} \xrightarrow{R} \sqcup\sqcup\,\omega\,\underline{\sqcup}\,, \omega \in \left(\Sigma\backslash\{\,\triangleright\,,\sqcup\,\}\right)^*.$

37. Какво и в какво преобразува D (delete) машината на Тюринг.
Заменя непразните символи от лентата с празни (изтрива ги).



38. Твърденията за разрешимите (рекурсивните) и полуразрешимите (рекурсивно номеруемите) езици:
- всеки разрешим език е полуразрешим;
- ако $L = \left(\Sigma\backslash\{\,\triangleright\,,\sqcup\,\}\right)^*$ е разрешим език, то и допълненоието му $\overline{L}$ е разрешим език;
- съществува полуразрешим език, който не е разрешим.

39. Кога една функция $f : \Sigma_\circ^* \nrightarrow \Sigma_\circ^*$ се изчислява с помощта на машина на Тюринг $M = \langle K, \Sigma, \delta, s, H\rangle$, $\Sigma_\circ^* \subseteq \Sigma\backslash\{\,\triangleright\,,\sqcup\,\}$.
Тогава, когато за всяка дума $\omega \in \Sigma_\circ^*$ са изпълнени условията:
- $(s, \triangleright\,\underline{\sqcup}\omega) \vdash_M^* (h, \triangleright\,\underline{\sqcup}y)$, за $y \in \Sigma_\circ^* \Leftrightarrow f(\omega) = y$
- $f(\omega)$ е определена $\Leftrightarrow$ M спира работа върху $(s, \triangleright\,\underline{\sqcup}\omega)$, тоест $M(\omega) = y$.

40. Кога една машина на Тюринг изчислява една функция $F : \mathbb{N}^k \nrightarrow \mathbb{N}$ на k променливи.
Нека $F : \mathbb{N}^k \nrightarrow \mathbb{N}$. Казваме, че машината на Тюринг $M = \langle K, \Sigma, \delta, s, H\rangle$ изчислява фунцкията F точно тогава, когато са изпълнени следните условия:
- $![F](n_1, \ldots, n_k)$ е дефинирана $\Leftrightarrow$ $M(1^{n_1} \sqcup \ldots \sqcup 1^{n_k}) \downarrow$ спира работа:
$(s, \triangleright\,\underline{\sqcup}1^{n_1} \sqcup \ldots \sqcup 1^{n_k}) \vdash_M^* \left(h, \triangleright\,\underline{\sqcup}1^{f(n_1,\ldots,n_k)}\right)$.
- ако $F(n_1, \ldots, n_k) = m$, то $M(1^{n_1} \sqcup \ldots \sqcup 1^{n_k}) = 1^m$.

41. Теоремата за неразрешимите проблеми на машина на Тюринг свързани с:
- празната дума
- съществуването на вход
- стоп-проблема
- всеки вход
- две машини на Тюринг
- регулярните езици

Следните проблеми на машината на Тюринг M са неразрешими:
- определяне на това дали M спира на празната дума.
- определяне на това дали M спира върху поне един вход. Тоест дали съществува дума ω, за която $M(\omega) \downarrow$ спира.
- определяне на това дали M спира при вход ω.
- определяне на това дали M спира за всеки вход.
- определяне на това давли две машини на Тюринг M_1 и M_2 спират върху един и същ вход.
- определяне на това дали $\mathscr{L}(M)$ е регулярен език.

ФОРМАЛИЗИРАНЕ НА ОПЕРАЦИИТЕ

42. Операцията минимизация.
Нека $f : \mathbb{N}^{n+1} \nrightarrow \mathbb{N}$. Казваме, че $g : \mathbb{N}^n \nrightarrow \mathbb{N}$ се получава от f с помощта на операцията

минимизация (μ-операция), ако за произволни $x_1, \ldots, x_n, y$ е изпълнена еквивалентността:
$g(x_1, \ldots, x_n) = y \Leftrightarrow f(x_1, \ldots, x_n, y) = 0 \wedge \forall z < y : f(x_1, \ldots, x_n, z)$ е дефинирана и $f(x_1, \ldots, x_n, z) > 0$.

43. Операцията примитивна рекурсия.
Нека $f : \mathbb{N} \nrightarrow \mathbb{N}$ и $g : \mathbb{N}^3 \nrightarrow \mathbb{N}$. Казваме, че $h : \mathbb{N}^2 \nrightarrow \mathbb{N}$ се определя с помощта на операцията примитивна рекирсия от f и g, ако за всяко $x, y \in \mathbb{N}$ е в сила:
$$\begin{cases} h(x,0) = f(x) \\ h(x, y+1) = g\big(x, y, h(x, y)\big) \end{cases}$$

44. Примитивно рекурсивна функция.
Индуктивна дефиниция:
1) Всички изходни ПРФ $\left(\{O, S, I_i^n\}\right)$ са ПРФ (винаги връщаща 0, инкрементираща с 1 (successor) и проектиращата функция).
2) Ако $f, g_1, \ldots, g_n$ са ПРФ, то и функцията h, която се получава от тях с помощта на операцията суперпозиция, също е ПРФ.
3) Ако f и g са ПРФ, то и функцията h, която се получава от f и g с помощта на операцията примитивна рекурсия, също е ПРФ.

45. Частично рекурсивна функция.
Индуктивна дефиниция:
1) Всички изходни ПРФ $\left(\{O, S, I_i^n\}\right)$ са ЧРФ (винаги връщаща 0, инкрементираща с 1 (successor) и проектиращата функция).
2) Ако $f, g_1, \ldots, g_n$ са ЧРФ, то и функцията h, която се получава от тях с помощта на операцията суперпозиция, също е ЧРФ.
3) Ако f и g са ЧРФ, то и функцията h, която се получава от f и g с помощта на операцията примитивна рекурсия, също е ЧРФ.
4) Ако f е ЧРФ, то и g, която се получава от f с помощта на μ-операция (минимизация), също е ЧРФ.