1. Контекстно-свободни граматики. 2. Дърветa за синтактичен анализ. 3. Нормална форма на Чомски. 4. Стекови автомати. 5. Връзка между стековите автомати и контекстно-свободните граматики (доказателство в едната посока по избор). 6. Свойства на затвореност. 7. Лема за покачването (xyzvw) (с доказателство). 8. Примери за езици, които не са контекстно-свободни.

4. Контекстно-свободни граматики и езици. Стекови автомати.

(1) **D** Контекстно-свободна граматика G наричаме четворката (V, Σ, R, S), където:
- V е крайна азбука,
- Σ (множество от терминали) е подмножество на V,
- R (множество от правила) е крайно подмножество на $(V-\Sigma) \times V^*$, и
- S (начален символ) е елемент на $V-\Sigma$.

Елементите на $V-\Sigma$ се наричат нетерминали. За всяко $A \in V-\Sigma$ и $u \in V^*$ казваме, че $A \to_G u$ е правило, когато $(A,u) \in R$. За всеки две думи $u, v \in V^*$ пишем $u \Rightarrow_G v$ т.с.т.к. има думи $x, y \in V^*$ и $A \in V-\Sigma$ такива че $u = xAy$, $v = xv'y$ и $A \to_G v'$. Релацията $\Rightarrow_G^*$ е рефлексивното, транзитивно затваряне на $\Rightarrow_G$. И накрая, $L(G)$, езикът генериран от G е $\{w \in \Sigma^* : S \Rightarrow_G^* w\}$, казваме още, че G генерира всяка дума от $L(G)$. Един език се нарича контекстно-свободен ако $L = L(G)$, за някоя КСГ G.

Когато граматиката се подразбира пишем $A \to w$ и $u \Rightarrow v$, вместо $A \to_G w$ и $u \Rightarrow_G v$.

Нарича се всяка последователност от вида:
$$w_0 \Rightarrow_G w_1 \Rightarrow_G w_2 \Rightarrow_G \dots \Rightarrow_G w_n$$
извод в G на w_n от w_0. Тук $w_0, \dots, w_n$ са произволни думи от V^*. За n наричаме дължина на извода, $n \in \mathbb{N}$ и $n \geq 0$. Казваме още, че изводът има n стъпки.

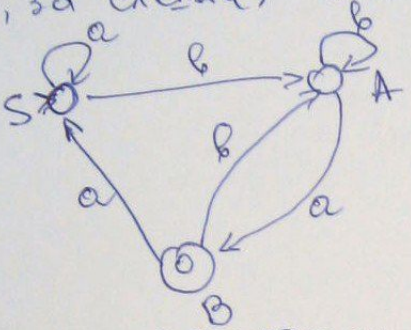
Пример! Следната граматика генерира всички изрази с правилно вложени скоби: всяка лява скоба трябва да има съответстваща дясна, и всяка затваряща трябва да има предшестваща съответстваща отваряща скоба. Още повече, низовете между всяка такава двойка скоби трябва да отговарят на същото условие. Нека $G = (V, \Sigma, R, S)$, където:
$V = \{S, (,)\}$,
$\Sigma = \{(,)\}$,
$R = \{S \to e,$
$S \to SS,$
$S \to (S)\}$.

Два извода от тази граматика са:
$$S \Rightarrow SS \Rightarrow S(S) \Rightarrow S((S)) \Rightarrow S(()) \Rightarrow ()(())$$
и
$$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()(())$$
Следователно една дума може да има няколко различни извода в една граматика.
$L(G)$ е пример за език, който е контекстно-свободен, но не е регулярен.

Очевидно има примери за езици, които са контекстно-свободни, но не са регулярни. Всички регулярни езици обаче са контекстно-свободни. Затова ще покажем няколко чрез директно построяване:
Нека разгледаме регулярния език, който се разпознава от крайния детерминиран автомат $M = (K, \Sigma, \delta, s, F)$. Същият език се генерира от граматиката $G(M) = (V, \Sigma, R, S)$, където $V = K \cup \Sigma$, $S = s$ и R се състои от тези правила:

$$R = \{q \to ap : \delta(q,a) = p\} \cup \{q \to e : q \in F\}$$

Тоест: нетерминалите са състоянията на автомата; за всеки преход от q към p при вход a имаме в R правило $q \to ap$.
Например, за следния автомат:



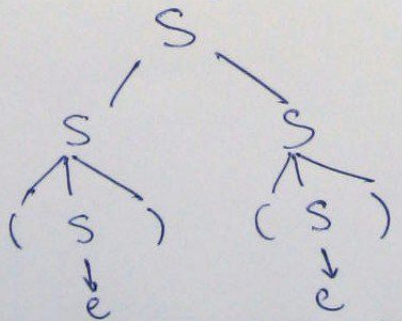
ще построим граматиката:
$S \to aS$, $S \to bA$, $A \to aB$, $A \to bA$, $B \to aS$, $B \to bA$, $B \to e$.

② Дървета на извода / Дървета за синтактичен анализ

Нека G е КСГ. Както показахме, думата $w \in L(G)$ може да има много изводи в G. Например в разглежданата по-горе граматика за правилно вложени скоби, думата $()()$ може да бъде получена от S чрез поне два различни извода:

$S \Rightarrow SS \Rightarrow (S)S \Rightarrow ()S \Rightarrow ()(S) \Rightarrow ()()$, и
$S \Rightarrow SS \Rightarrow S(S) \Rightarrow (S)(S) \Rightarrow (S)() \Rightarrow ()()$

Ако разгледаме тези два извода ще забележим, че те са еднакви - различават се единствено по реда на прилагане на правилата. Интуитивно и двете могат да бъдат означени графично така:



Диаграми от горния тип нарисуваме дървета на извода. Точките нарисуваме върхове; всеки връх е означен със символ от V. Най-горният връх се нарича корен, а най-долните - листа. Всички листа са означени или с терминални символи, или е празната дума e. Ако следим листата на дървото ще получим думата, която се нарича резултат на дървото на извода.

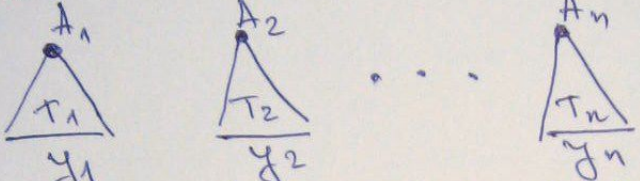
Или казано по-формално, за произволна КСГ $G=(V,\Sigma,R,S)$ дефинираме дърво на извода, корен, листа, ~~yield~~ резултат, така:

(1) $\overset{a}{\circ}$ Това е дървото на извода за всяко $a \in \Sigma$. Единственият връх е и корен, и лист. ~~Yield~~ Резултатът е "a".

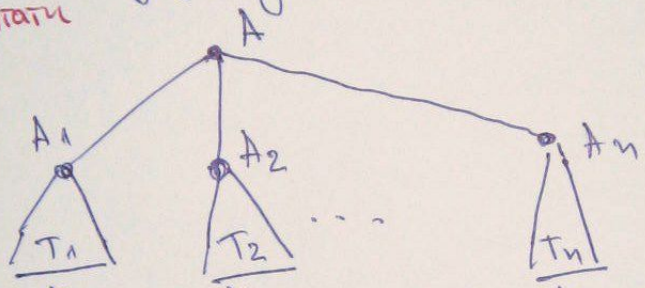
(2) Ако $A \to e$ е правило в R, то

е дърво на извод. Коренът му е означен с A, единственото му листо е e, ~~yield~~ резултатът е e.

(3) Ако



са дървета на извод, където $n \geq 1$, с корени обозначени с $A_1, \ldots, A_n$ и ~~yields~~ резултати $y_1, \ldots, y_n$ и $A \to A_1 \ldots A_n$ е правило от R, то



е дърво на извода. Коренът му е новият връх A, листата му са листата на дърветата $A_1 \ldots A_n$, yield е $y_1 \ldots y_n$.

(4) Няма други дървета на извода.

Интуитивно, дърветата на извода представят изводите по такъв начин, че различните дъщерни се на реда на прилагане на правилата, се елиминират. Казано иначе, дърветата на извода дефинират класове на еквивалентности на изводите. По формално: Нека $G=(V,\Sigma,R,S)$ е КСГ, $D = x_1 \Rightarrow x_2 \Rightarrow \ldots \Rightarrow x_n$ и $D' = x'_1 \Rightarrow \ldots \Rightarrow x'_n$ са два извода в G, където $x_i, x'_i \in V^*$ за $i = 1 \ldots n$, $x_1, x'_1 \in V-\Sigma$ и $x_n, x'_n \in \Sigma^*$. Тоест, и двата са изводи на терминални символни от един нетерминал. Казваме, че D предхожда D' (записваме $D \lhd D'$), ако $n > 2$ и съществува $k \in \mathbb{N}$, $1 < k < n$, такова че:

(1) За всяко $i \neq k$ $x_i = x'_i$

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

(3) $x_k = uywBw$, където $A \to y \in R$

(4) $x'_k = uAvzw$, където $B \to z \in R$

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

С други думи, двата извода са идентични, с изключение на две последователни стъпки, през които едни и същи два нетерминала са заменени с едни и същи думи, но в противоположен ред. Изводите, в които по-лявото нетерминал е заменено първо предхожда другия.

Пример | Да разгледаме следните 3 извода D_1, D_2 и D_3 може без примера

$D_1 = S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow (())S \Rightarrow (())(S) \Rightarrow (())()$

$D_2 = S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow ((S))(S) \Rightarrow (())(S) \Rightarrow (())()$

$D_3 = S \Rightarrow SS \Rightarrow (S)S \Rightarrow ((S))S \Rightarrow ((S))(S) \Rightarrow ((S))() \Rightarrow (())()$

Имаме, че $D_1 \prec D_2$ и $D_2 \prec D_3$, но $D_1 \prec D_3$ НЕ е изпълнено, тъй като D_1 и D_3 се различават в повече от една междинна дума. И трите имат едно и също дърво на извода.

Казваме, че два извода D и D' са подобни, ако двойката (D, D') принадлежи на рефлексивното, симетрично, транзитивно затваряне на $\prec$ (следователно подобието е релация на еквивалентност).

Всеки такъв клас на еквивалентност (т.е. всяко дърво на извода), съдържа извод, който е максимален спрямо $\prec$ (не предхожда никой друг извод) – нарича се най-ляв извод и се получава като от корена A последователно заменяме най-левия нетерминал в думата според правилото, означено в дървото. Подобно се дефинира най-десен извод – извод, който не предхожда никой друг извод и се получава чрез последователно разширяване на най-десния нетерминал. Всяко дърво на извода определя точно един най-ляв и точно един най-десен извод.

За да обобщим, ще формулираме следната теорема:

Т | Нека $G = (V, \Sigma, R, S)$ е КСГ и $A \in V - \Sigma$ и $w \in \Sigma^*$. Следните твърдения са еквивалентни:

(a) $A \Rightarrow^* w$

(b) Съществува дърво на извода с корен A и резултат w.

(c) Съществува най-ляв извод $A \overset{L}{\Rightarrow}^* w$.

(d) Съществува най-десен извод $A \overset{R}{\Rightarrow}^* w$.

④ Стекови Автомати

Не всеки контекстно-свободен език може да бъде разпознат от краен автомат, понеже не всеки КСЕ е регулярен. За да получим автомати, които да разпознават контекстно-свободните езици ще трябва да добавим нови възможности към крайните автомати. Оказва се, че ключовата особеност на тези по-сложни автомати е добавянето на допълнителна памет, под формата на стек.

D| Ще дефинираме стеков автомат като наредената шесторка $M = (K, \Sigma, \Gamma, \Delta, s, F)$, където:
- K е крайно множество от състояния
- Σ е крайна азбука (входни символи)
- Γ е крайна азбука (символи на стека)
- $s \in K$ е началното състояние
- $F \subseteq K$ е множество от крайни състояния, и
- Δ, релация на прехода, е крайно подмножество на: $(K \times (\Sigma \cup \{e\}) \times \Gamma^*) \times (K \times \Gamma^*)$.

Интуитивно, ако $((p, a, \beta), (q, \gamma)) \in \Delta$, тогава M, винаги когато е в състояние p с β на върха на стека, може да прочете a от входа (или нищо, ако $a = e$), да замени β с γ на върха на стека, и да премине към състояние q. Така двойката $((p, a, \beta), (q, \gamma))$ се нарича преход в M. Тъй като в даден момент може да има няколко възможни прехода, стековите автомати по принцип са недетерминистични.

Дефинираме две операции: push (добавяне на символ на върха на стека), и pop (премахване на символ от върха на стека).

Например преходът $((p, a, e), (q, a))$ демонстрира push, а $((p, a, a), (q, e))$ – pop.

Конфигурация на стековия автомат се дефинира като елемент на $K \times \Sigma^* \times \Gamma^*$: първият компонент е текущото състояние, вторият е частта от лентата, която още не е била прочетена, а третият – съдържанието на стека, прочетено отгоре надолу.

Ако (p, x, α) и (q, y, β) са две конфигурации на M, казваме че (p, x, α) води след една стъпка до (q, y, β) ($(p, x, \alpha) \vdash_M (q, y, \beta)$) ако съществува преход $((p, a, \beta), (q, \gamma)) \in \Delta$, такъв че $x = ay$ и $\alpha = \delta\eta$ и $\beta = \gamma\eta$ за някое $\eta \in \Gamma^*$. Бележим рефлексивното и транзитивно затваряне на $\vdash_M$ с $\vdash_M^*$. Казваме, че M разпознава думата $w \in \Sigma^*$ т.с.т.к. $(s, w, e) \vdash_M^* (p, e, e)$ за някое $p \in F$. Казано иначе M разпознава w т.с.т.к. $\exists$ последователности от конфигурации $C_0, C_1, \ldots, C_n$ ($n > 0$) такава че $C_0 \vdash_M C_1 \vdash_M \ldots \vdash_M C_n$, $C_0 = (s, w, e)$, $C_n = (p, e, e)$ за някое $p \in F$. Всяка последователност от конфигурации $C_0, C_1, \ldots, C_n$, такава че $C_i \vdash_M C_{i+1}$, $i = 0, \ldots, n-1$ се нарича изчисление от M; дължината ѝ е n, или има и стойност. Езикът, разпознат от M, бележим с $L(M)$, е множеството от всички думи, разпознавани от M.

Когато не може да настъпи объркване пишем $\vdash$ и $\vdash^*$ вместо $\vdash_M$ и $\vdash_M^*$.

⑤

T| Класът на езиците, разпознавани от стекови автомати, съвпада с класа на контекстно свободните езици.

Доказателство: Ще разделим доказателството на две части

=>)

Лема Всеки контекстно-свободен език се разпознава от някой стеков автомат.

Доказателство:

Нека $G = (V, \Sigma, R, S)$ е КСГ; трябва да построим стеков автомат M, така че $L(M) = L(G)$. Ще построим автомат с само две състояния, p и q, който стои винаги в състоянието q след първия си преход. M използва V, множеството от терминали и нетерминали, за азбуката на стека. Нека $M = (\{p, q\}, \Sigma, V, \Delta, p, \{q\})$, където Δ съдържа следните преходи:

(1) $((p, e, e), (q, S))$

(2) $((q, e, A), (q, x))$ за всяко правило $A \to x$ в R (преход тип 1)

(3) $((q, a, a), (q, e))$ за всяко $a \in \Sigma$

Стековият автомат първо слага S на върха на стека, и след това последователно или

- замества най-горния символ A в стека, ако е нетерминал, с дясната страна x на някое правило $A \to x$ (преход тип 2), или
- премахва символа на върха на стека ако е терминал и съвпада със следващия символ на лентата (преход тип 3).

По този начин M симулира най-ляв извод на входа.

За да докажем, че $L(M) = L(G)$, ще използваме следното твърдение:

Твърдение Нека $w \in \Sigma^*$ и $\alpha \in (V - \Sigma)V^* \cup \{e\}$. Тогава $S \overset{L}{\Rightarrow}^* w\alpha$ т.с.т.к. $(q, w, S) \vdash_M^* (q, e, \alpha)$

Това твърдение ще ни помогне да докажем лемата, тъй като ако $\alpha = e$ ще следва, че $S \overset{L}{\Rightarrow}^* w$ т.с.т.к. $(q, e, S) \vdash_M^* (q, e, e)$, т.е. $w \in L(G)$ т.с.т.к. $w \in L(M)$.

$(\Rightarrow)$ Нека $S \overset{L}{\Rightarrow}^* w\alpha$, където $w \in \Sigma^*$ и $\alpha \in (V - \Sigma)V^* \cup \{e\}$. Ще покажем по индукция по дължината на най-левия извод на w от S, че $(q, w, S) \vdash_M^* (q, e, \alpha)$.

1. Ако дължината на извода е 0, тогава $w = e$, $\alpha = S$ и накрая $(q, w, S) \vdash_M^* (q, e, \alpha)$
2. Допускаме, че ако $S \overset{L}{\Rightarrow}^* w\alpha$ при извод с дължина n или по-малко, $n \geq 0$, то $(q, w, S) \vdash_M^* (q, e, \alpha)$.
3. Нека $S = u_0 \overset{L}{\Rightarrow} u_1 \overset{L}{\Rightarrow} \dots \overset{L}{\Rightarrow} u_n \overset{L}{\Rightarrow} u_{n+1} = w\alpha$ да е най-ляв извод на $w\alpha$ от S. Нека A е най-левият нетерминал на u_n. Тогава $u_n = xA\beta$ и $u_{n+1} = x\gamma\beta$, където $x \in \Sigma^*$, $\beta, \gamma \in V^*$ и $A \to \gamma$ е правило в R. Тъй като съществува най-ляв извод с дължина n на $u_n = xA\beta$ от S, то по И.П.

$$(q, x, S) \vdash_M^* (q, e, A\beta) \qquad (2)$$

Тъй като $A \to \gamma$ е правило в R:

$$(q, e, A\beta) \vdash_M (q, e, \gamma\beta) \quad (3)$$

след преход от тип 2.

Тъй като u_{n+1} е $w\alpha$, но и $x\gamma\beta$, има дума $y \in \Sigma^*$, такава че $w = xy$ и $y\alpha = \gamma\beta$. Тогава може да приложим (2) и (3), така:

$$(q, w, S) \vdash_M^* (q, y, \gamma\beta) \quad (4)$$

Но $y\alpha = \gamma\beta \Rightarrow$

$$(q, y, \gamma\beta) \vdash_M^* (q, e, \alpha) \quad (5)$$

след последователност от $|y|$ на брой прехода от тип 3.

Обединяването на (4) и (5) приключва индукцията.

$\Leftarrow$) Сега да допуснем, че $(q, w, S) \vdash_M^* (q, e, \alpha)$ с $w \in \Sigma^*$ и $\alpha \in (V - \Sigma)^* \cup \{\varepsilon\}$. Ще покажем, че $S \overset{L}{\Rightarrow}^* w\alpha$. Отново ще използваме индукция, този път по броя преходи от тип 2.

1. Тъй като първият преход винаги е от тип 2, ако $(q, w, S) \vdash_M^* (q, e, \alpha)$ без преходи от тип 2, то $w = e$ и $\alpha = S \Rightarrow$ вярно

2. Ако $(q, w, S) \vdash_M^* (q, e, \alpha)$ след n стъпки от тип 2 (или по-малко), $n \geq 0$, то $S \overset{L}{\Rightarrow}^* w\alpha$

3. Да предположим, че $(q, w, S) \vdash_M^* (q, e, \alpha)$ с $n+1$ прехода от тип 2, и да разгледаме предпоследния такъв преход:

$$(q, w, S) \vdash_M^* (q, y, A\beta) \vdash_M (q, y, \gamma\beta) \vdash_M^* (q, e, \alpha)$$

където $w = xy$ за някои $x, y \in \Sigma^*$ и $A \to \gamma$ е правило от граматиката. Според И.Х. имаме че $S \overset{L}{\Rightarrow}^* xA\beta$ и $\Rightarrow S \overset{L}{\Rightarrow}^* x\gamma\beta$. Но понеже $(q, y, \gamma\beta) \vdash_M^* (q, e, \alpha)$, евентуално след преход от тип 3, следва че $y\alpha = \gamma\beta$ и следователно $S \overset{L}{\Rightarrow}^* xy\alpha = w\alpha$. С това лемата е доказана, както и първата половина на теоремата.

Нека пристъпим към втората половина на Теоремата:

Лема 1 (<=) Ако един език се разпознава от стеков автомат, то той е контекстно свободен.

Доказателство:

Удобно ще е да ограничим някак стековия автомат, който разпознава. Ще наричаме стековия автомат *прост* ако:

Винаги когато $((q, a, \beta), (p, \gamma))$ е преход в стековия автомат и q *не е* началното състояние, то $\beta \in \Gamma$ и $|\gamma| \leq 2$.

с други думи вместо се разглежда само символа на върха на стека и се заменя или с e, или с 1 или с 2 стекови символа.

Твърдим, че ако един език се разпознава от обикновен стеков автомат, то той се разпознава и от прост стеков автомат. За дадена дека $M = (K, \Sigma, \Gamma, \Delta, s, F)$ е произволен стеков автомат. Ще построим $M' = (K', \Sigma, \Gamma \cup \{Z\}, \Delta', s', \{f'\})$ – прост, който също разпознава $L(M)$. Тук s' и f' са нови състояния, непринадлежащи в K, а Z е нов стеков символ, различен дъното на стека (също непринадлежащ в Γ).

Започваме като към Δ^* добавяме преходите:

1) $((s', e, e), (s, Z))$
2) $((f, e, Z), (f', e))$ за $\forall f \in F$.

Така Δ' съдържа новите 2 прехода, както и всички преходи от Δ. Следващата ни стъпка ще е да заменим в Δ' всички преходи, които не удовлетворяват условието за простота, с еквивалентни преходи, които го удовлетворяват. Това ще стане на 3 етапа:

- Преходи с $|\beta| \geq 2$
- Преходи с $|\gamma| > 2$ (без да добавяме преходи с $|\beta| \geq 2$)
- Преходи с $\beta = e$ (без да нарушаваме предните 2).

1) За всички преходи от типа $((q, a, \beta), (p, \gamma)) \in \Delta'$ и $|\beta| > 1$, $\beta = B_1 \dots B_n$, $n > 1$, добавяме към Δ':

$((q, e, B_1), (q_{B_1}, e))$,
$((q_{B_1}, e, B_2), (q_{B_1 B_2}, e))$,
$\dots$
$((q_{B_1 \dots B_{n-1}}, e, B_n), (p, \gamma))$;

където за $i = 1 \dots n-1$, $q_{B_1 B_2 \dots B_i}$ е ново състояние със значение "състояние q след прочитане на символите $B_1 \dots B_i$ от стека". Очевидно новият автомат е еквивалентен на оригиналния.

2) По подобен начин заменяме преходите $((q, a, \beta), (p, \gamma))$, $\gamma = c_1 \dots c_m$, $m \geq 2$ с

$((q, a, \beta), (r_1, c_m))$
$((r_1, e, e), (r_2, c_{m-1}))$
$\vdots$
$((r_{m-2}, e, e), (r_{m-1}, c_2))$
$((r_{m-1}, e, e), (p, c_1))$

$r_1 \dots r_{m-1}$ са нови състояния. Нещо повече, сега $\forall$ преходи $((q, a, \beta), (p, \gamma)) \in \Delta'$ изпълняват $|\gamma| \leq 1$. Ще го възстановим до $|\gamma| \leq 2$ на следващата стъпка. Освен това, не добавихме преходи с $|\beta| > 1$.

3) Последно, нека разгледаме преходи от вида $((q,a,c),(p,\delta))$ с $q \neq s'$ – само те все още нарушават условието за простота. Ще заменим тези преходи с $((q,a,A),(p,\gamma A))$ за $\forall A \in \Gamma \cup \{z\}$. Сега, ако автоматът може да се премества без да поглежда стека, то той може да се премества и ако погледне последния символ и веднага го върне.

Лесно се вижда, че така получаваме прост автомат M' и $L(M)=L(M')$.

За да докажем лемата ще покажем КСГ G, такава че $L(G)=L(M')$.

Нека $G=(V,\Sigma,R,S)$. V съдържа освен новия символ S и символите в Σ, нов символ $\langle q,A,p\rangle$ за $q,p \in K'$ и за всяко $A \in \Gamma \cup \{e,z\}$. За да се разбере ролята на нетерминалите $\langle q,A,p\rangle$ трябва да се има предвид, че G трябва да генерира всички думи, разпознавани от M' $\Rightarrow$ нетерминалите на G представляват разумни части на входните думи, които се разпознават от M'. В частност, ако $A \in \Gamma$, тогава нетерминалът $\langle q,A,p\rangle$ представлява някоя и да е част от входния низ, която може да бъде прочетена между моменти, в които M' е в състояние q със A на върха на стека, и момента, в който M' премахва A от върха на стека и преминава в състояние p. Ако $A=e$, то $\langle q,e,p\rangle$ представлява частта от входната дума, която може да се прочете между състояние q и p, без стекът да се прочита или променя.

Правилата в R са от 4 типа:

(1) Правилото $S \to \langle s,z,f'\rangle$, където s е началното състояние на M, а f' – новото крайно състояние.

(2) За всеки преход $((q,a,B),(r,C))$, където $q,r \in K'$, $a \in \Sigma \cup \{e\}$, $B,C \in \Gamma \cup \{e\}$ и за всяко $p \in K'$, добавяме правилото $\langle q,B,p\rangle \to a\langle r,C,p\rangle$.

(3) За всеки преход $((q,a,B),(r,C_1C_2))$, където $q,r \in K'$, $a \in \Sigma \cup \{e\}$, $B \in \Gamma \cup \{e\}$, и $C_1,C_2 \in \Gamma$ и за всяко $p,p' \in K'$, добавяме правилото $\langle q,B,p\rangle \to a\langle r,C_1,p'\rangle\langle p',C_2,p\rangle$.

(4) За $\forall q \in K'$, правилото $\langle q,e,q\rangle \to e$

Тъй като M' е прост, или тип 2 или тип 3 отговаря на всеки преход от M'.

Твърдение: За $\forall q,p \in K'$, $A \in \Gamma \cup \{e\}$ и $x \in \Sigma^*$,

$\langle q,A,p\rangle \Rightarrow_G^* x$ т.с.т.к. $(q,x,A) \vdash_{M'}^* (p,e,e)$.

Лемата, а с това и теоремата, следват от това твърдение
Поне $\langle s, e, f \rangle \Rightarrow_G^* x$ за някое $f \in F$ т.ст.к. $(s, x, e) \vdash_M^* (s', \varepsilon, e')$
т.е. $x \in L(G)$ т.ст.к $x \in L(M')$.
И двете посоки на твърдението могат да бъдат доказани по индукция, или по дължината на извода на G, или по дължината на изчислението на M!

(6)

Т| Контекстно-свободните езици са затворени спрямо обединение, слепване и звезда на Клини.

Доказателство:

Нека $G_1 = (V_1, \Sigma_1, R_1, S_1)$ и $G_2 = (V_2, \Sigma_2, R_2, S_2)$ са две КСГ. Без
можем да считаме, че $V_1 - \Sigma_1$ и $V_2 - \Sigma_2$ нямат общи елементи.

Обединение: Нека S е нов символ и нека $G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R, S)$, където $R = R_1 \cup R_2 \cup \{S \to S_1, S \to S_2\}$. Тогава твърдим, че
$L(G) = L(G_1) \cup L(G_2)$. Понеже единствените правила, в които участва
S, са $S \to S_1$ и $S \to S_2$, то $S \Rightarrow_G^* w$ т.с.т.к. или $S_1 \Rightarrow_G^* w$, или
$S_2 \Rightarrow_G^* w$; и т.к. G_1 и G_2 имат ~~нито~~ множества от нетерминали без
общи елементи, последното твърдение е еквивалентно на $w \in L(G_1) \cup L(G_2)$.

Слепване: Подходяме по подобен начин: $L(G_1)L(G_2)$ се генерира от граматиката:

$$G = (V_1 \cup V_2 \cup \{S\}, \Sigma_1 \cup \Sigma_2, R_1 \cup R_2 \cup \{S \to S_1 S_2\}, S)$$

Звезда на Клини: $L(G_1)^*$ се генерира от:

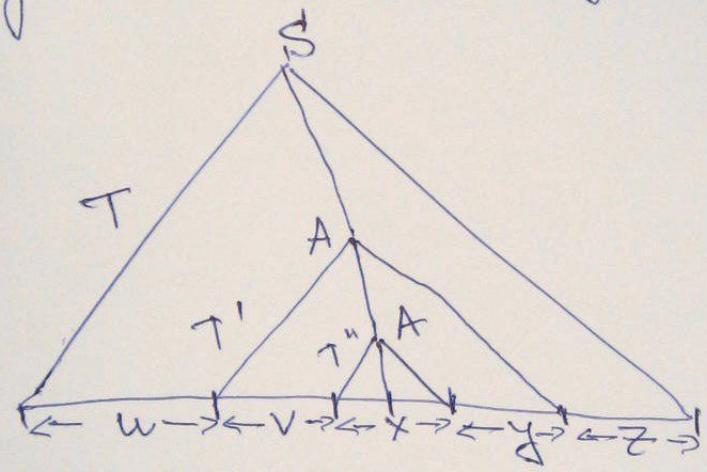
$$G = (V_1 \cup \{S\}, \Sigma_1, R_1 \cup \{S \to \varepsilon, S \to S S_1\}, S)$$

⑦

Т] Нека $G=(V,\Sigma,R,S)$ е КСГ. Тогава $\forall$ дума $w \in L(G)$ с дължина поне от $\phi(G)^{|V-\Sigma|}$ може да бъде представена във вида $w=uvxyz$, така че или v или y е непразна и $uv^nxy^nz \in L(G)$ за $\forall n \geq 0$.

Доказателство:

Нека w е такава дума и T е дървото на извода с корен S и с ~~yield~~ резултат, който има най-малък брой листа сред всички дървета на извод със същите корен и ~~yield~~ резултат. Тъй като ~~yield~~ резултат на T е по-дълъг от $\phi(G)^{|V-\Sigma|}$ следва, че в T има път с дължина поне $|V-\Sigma|+1$, т.е. с поне $|V-\Sigma|+2$ върха. Само един от тези върхове може да е означен с терминал $\Rightarrow$ останалите са означени с нетерминали. Тъй като по пътя има повече върхове отколкото нетерминали, два върха от пътя са означени със същия елемент A от $V-\Sigma$. Да разгледаме пътя в повече детайли:



X – ~~yield~~ резултат на T''
V – част от ~~yield~~ резултата на T' до началото на ~~yield~~ резултата на T''
U – част от ~~yield~~ резултата на T до началото на T'
Z – останалата част от ~~yield~~ резултата на T.

Оттук е ясно, че частта от T' изключвайки T'' може да бъде повторена произволен брой пъти (включително 0), за да се получат други дървета на извода с ~~yield~~ резултат uv^nxy^nz, $n \geq 0$. Остана само да докажем, че $vy \neq \varepsilon$. Но ако $vy=\varepsilon$, тогава има дърво с корен S и ~~yield~~ резултат w с по-малко листа от T, получено ако от T премахнем частта от T', която изключва T'' $\Rightarrow$ Противоречие.

Т] Контекстно-свободните езици не са затворени относно сечение или допълнение. **Доказателство: вижи допълнението.**

~~Алгоритъм за КСГ~~

③

Д КСГ $G=(V,\Sigma,R,S)$ е в Нормална форма на Чомски ако $R \subseteq (V-\Sigma) \times V^2$.

Т За всяка КСГ G $\exists$ КСГ G' в Нормална форма на Чомски, такава че $L(G') = L(G) - (\Sigma \cup \{\varepsilon\})$. Още повече, конструирането на G' може да бъде извършено за полиномиално относно големината на G време.

Доказателство:

Нека $G=(V,\Sigma,R,S)$. Съществуват 3 вида правила $A \to x$, които не отговарят на условията:

1) Дълги правила ($|x| \geq 3$).
Нека $A \to B_1 \dots B_n \in R$, $B_1, \dots B_n \in V$, $n \geq 3$. Заместваме с $n-1$ нови правила:

$$A \to B_1 A_1$$
$$A_1 \to B_2 A_2$$
$$A_2 \to B_3 A_3$$
$$\vdots$$
$$A_{n-2} \to B_{n-1} B_n$$

$A_1 \dots A_{n-2}$ - нови нетерминали. След прилагане за $\forall$ такова правило получаваме еквивалентна граматика.

2) ε-правила ($A \to \varepsilon$).
За целта определяме множеството:

$$\mathcal{E} = \{A \in V-\Sigma : A \Rightarrow^* \varepsilon\}$$

чрез следния алгоритъм:

$\mathcal{E} := \emptyset$
while there is rule $A \to \alpha$, $\alpha \in \mathcal{E}^*$ и $A \notin \mathcal{E}$ do
add A to $\mathcal{E}$

След това изтриваме от G всички ε-правила и: За $\forall$ правило от вида $A \to BC$ или $A \to CB$, където $B \in \mathcal{E}$ и $C \in V$ добавяме в граматиката $A \to C$. Единственият извод, който новата граматика би могла да изпуска, е $S \to \varepsilon$ - за щастие, формулировката на теоремата позволява това.

3) Хвси правила ($A \to a$ или $A \to B$)
За $\forall A \in V$ изчисляване множеството $D(A) = \{B \in V : A \Rightarrow^* B\}$:

$D(A) := \{A\}$
while there is a rule $B \to C$ with $B \in D[A]$ and $C \notin D(A)$ do
    add C to $D(A)$.

Ако a е терминал, то $D(a) = a$.
Премахване всички хвси правила и заменяне $\forall$ правила от вида $A \to B^* C^*$ с всички възможни правила от типа $A \to B'C'$, $B' \in D(B)$, $C' \in D(C)$. Накрая добавяне правилата $S \to BC$ за $\forall$ правило $A \to BC$ такова че $A \in D(S) - \{S\}$
Този нст е възможно да пропуснем правила от вида $S \to a$, но отново това е позволено от информацията.
Нещо повече, получеият алгоритъм е полиномиален по време относно $n = |R|$.
Стъпка 1) отнема $O(n)$ и остава граматика с големина $O(n)$.
Втората отнема $O(n^2)$, а третата - $O(n^3)$. С това теоремата е доказана.
Предимството на нормалната форма на Чомски е в това, че позволява използването на полиномиален алгоритъм за определяне дали дадена дума може да се генерира от КСГ:

for $i := 1$ to n do $N[i,i] := \{x_i\}$; All other $N[i,j]$ are empty
for $s := 1$ to n do
    for $i := 1$ to $n-s$ do
        for $k := i$ to $i+s-1$ do
            if there is a rule $A \to BC \in R$ with $B \in N[i,k] \& C \in N[k+1, i+s]$
                add A to $N[i, i+s]$

Accept x if $S \in N(1,n)$.

Допълнения:

(8) Примери за езици, които не са контекстно-свободни:

→ $L = \{a^n b^n c^n : n \geq 0\}$

Да допуснем, че $L = L(G)$ за някоя КСГ G. Нека $n > \phi(G)^{|V-\Sigma|}$. Тогава $w = a^n b^n c^n$ е от $L(G)$ и може да се представи в вид $uvxyz$ така че v или y е непразно и $uv^m xy^m z$ е от $L(G)$, за всяко $m = 0, 1, 2, \ldots$. Има два случая:

- vy съдържа и трите символа a, b, c, тогава поне едно от v, y трябва да съдържа поне два от тях (Дирихле). Но тогава uv^2xy^2z съдържа два срещания в "грешен" ред - или "b" преди "a", или "c" преди "a" или "b".
- vy съдържа някои, но не всичките три символа. Но тогава uv^2xy^2z съдържа неравен брой a, b, c.

→ $L = \{a^n : n \geq 1 \text{ е просто число}\}$

Нека $p = \phi(G)^{|V-\Sigma|}$, където G е КСГ, която поражда L. Тогава $w = a^p$ може да бъде представена във вида $w = uvxyz$, $vy \neq \varepsilon$. Нека $vy = a^q$ и $uxz = a^r$, $q, r \in \mathbb{N}$, $q > 0$. Тогава, според лемата, $r + nq$ е просто, $\forall n \geq 0$. Но това е невъзможно, защото ако например $n = r + 2q + 2 \Rightarrow$ $r + 2q^2 + qr + 2q = r(q+1) + 2q(q+1) = (2q+r)(q+1)$ е просто = ↯
$\geq 2 \quad \geq 2$

→ $L = \{w \in \{a, b, c\}^* : w$ има равен брой $a, b, c\}$.

Да допуснем, че L е КС $\Rightarrow$ по Th за сечение на КС и регулярен език, КС ще бъде и сечението на L с регулярния език $a^* b^* c^*$. Но ние току-що доказахме, че този език ($\{a^n b^n c^n : n \geq 0\}$) не е КС = ↯

(6) - допълнение:

Т Контекстно-свободните езици не са затворени спрямо сечение или допълнение.

Доказателство:

Ясно е, че $\{a^n b^n c^m : n, m \geq 0\}$ и $\{a^m b^n c^n : m, n \geq 0\}$ са КС. Сечението им е езикът $\{a^n b^n c^n : n \geq 0\}$, който показахме, че не е КС. И тъй като:

$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$ и понеже КСЕ са затворени спрямо $\cup$, $\cap$ и $\overline{\ }$, но КСЕ бяха затворени спрямо операциите, които бяха да бъдат затворени и спрямо сечение (а те не са).

⇒ КСЕ не са затворени спрямо $\cap$ и $\overline{\ }$.