



1.3 КОНТЕКСТНО СВОБОДНИ ЕЗИЦИ

Пример

Аритметични изрази: $G = (\{E, T, F\}, \{a, +, *, (,)\}, P, E)$ с
 $P = \{E \rightarrow T, E \rightarrow E + T, T \rightarrow F, T \rightarrow T * F, F \rightarrow a, F \rightarrow (E)\}$

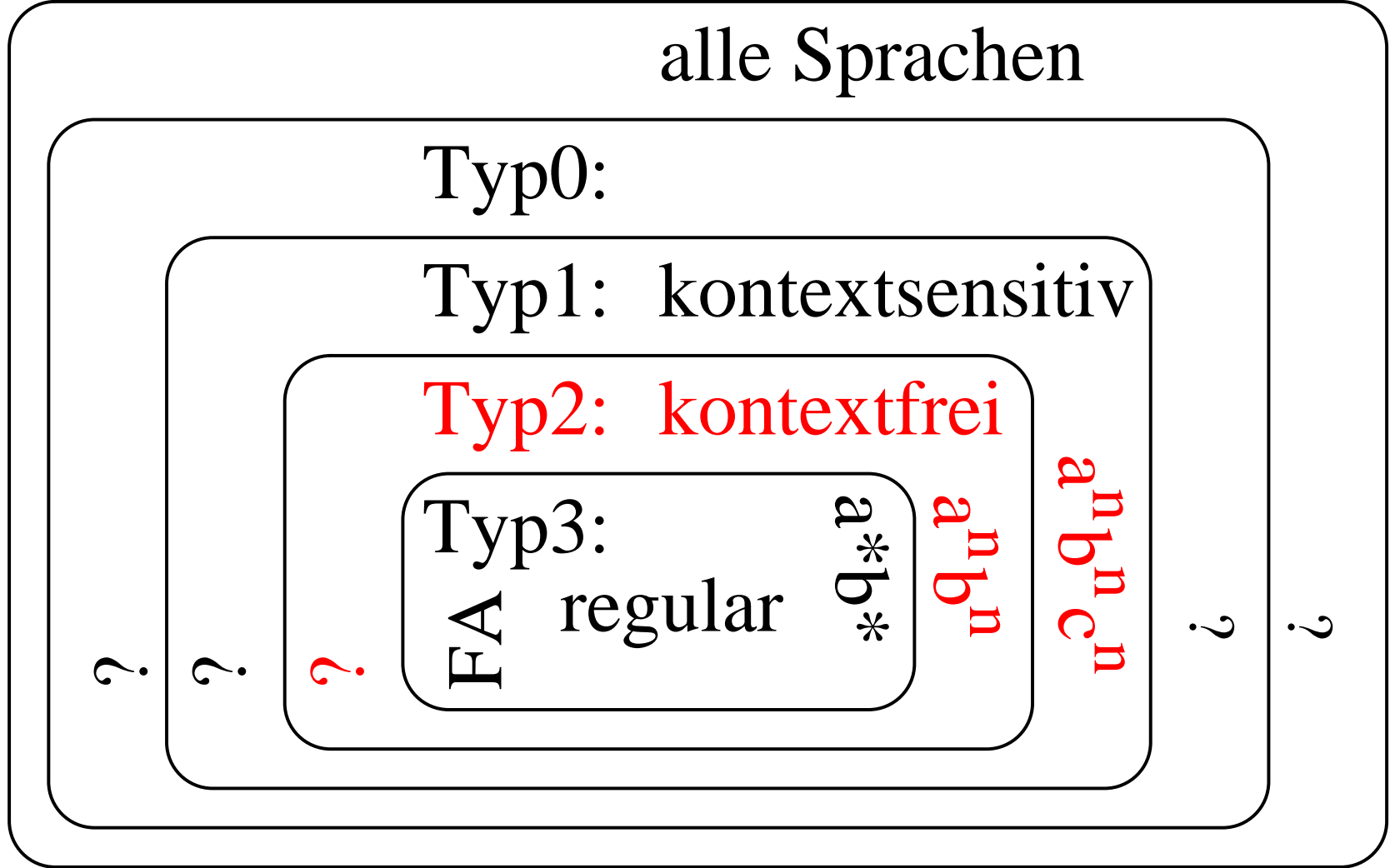
$\{a^n b^n : n \geq 1\}$: $G = (\{S\}, \{a, b\}, \{S \rightarrow ab, S \rightarrow aSb\}, S)$

Синтаксис на програмните езици: Pascal, C, ...



Иерархия на Чомски

Maschinenmodelle



Sprachbeispiele



План

1. Нормална форма
2. Негативни резултати с помощта на Pumping-лемата
3. Свойства на затвореност
4. Проблемът за принадлежност на дума
5. Стекови автомати



1.3.1 Нормална форма на Чомски

Една граматика $G' = (V, \Sigma, P, S)$ е в **нормална форма на Чомски**, ако

$$P \subseteq V \times \Sigma \cup V \times VV.$$

- ϵ -елиминиране
- елиминиране на (единичните) правила от вида $A \rightarrow B$
- елиминиране на правилата с дълга дясна част



\mathcal{E} -елиминация

$\forall G = (V, \Sigma, P, S)$ с $P \subseteq V \times (V \cup \Sigma)^*$:

$\exists G' : L(G) = L(G'), G'$ е от тип 2



ε елиминация за $G = (V, \Sigma, P, S)$

$V_\varepsilon := \{\}$

while $\exists X \rightarrow \alpha \in P : X \notin V_\varepsilon \wedge \alpha \in V_\varepsilon^*$ do $V_\varepsilon := V_\varepsilon \cup \{X\}$

assert $V_\varepsilon = \{X \in V : X \xrightarrow{*} \varepsilon\}$

while $\exists X \rightarrow \alpha Y \beta \in P : Y \in V_\varepsilon \wedge X \rightarrow \alpha \beta \notin P$ do

$P := P \cup X \rightarrow \alpha \beta$ // invariant : $L(G)$

$P := P \setminus (V \times \{\varepsilon\})$ // invariant : $L(G) \setminus \{\varepsilon\}$

if $S \in V_\varepsilon$ then return $(V \cup \{S'\}, \Sigma, P \cup \{S' \rightarrow \varepsilon, S' \rightarrow S\}, S')$

else return (V, Σ, P, S)

Упражнение: Линейно време за откриване на V_ε .

$(\mathcal{O}(|V| + \sum_{X \rightarrow r \in P} |r|))$



Упражнение a^*b^*

$$G = (\{S, A, B\}, \{a, b\}, P, S),$$

$$P = \{S \rightarrow AB, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon\}$$

while $\exists X \rightarrow \alpha \in P : X \notin V_\varepsilon \wedge \alpha \in V_\varepsilon^*$ do $V_\varepsilon := V_\varepsilon \cup \{X\}$

$$V_\varepsilon : \{\} \rightsquigarrow \{A\} \rightsquigarrow \{A, B\} \rightsquigarrow \{A, B, S\}$$



Упражнение a^*b^*

$$G = (\{S, A, B\}, \{a, b\}, P, S),$$

$$P = \{S \rightarrow AB, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon\}$$

$$V_\varepsilon = \{A, B, S\}$$

while $\exists X \rightarrow \alpha Y \beta \in P : Y \in V_\varepsilon \wedge X \rightarrow \alpha\beta \notin P$ do $P := P \cup X \rightarrow \alpha\beta$

$$A \rightarrow aA, A \in V_\varepsilon \rightsquigarrow A \rightarrow a$$

$$B \rightarrow bB, B \in V_\varepsilon \rightsquigarrow B \rightarrow b$$

$$S \rightarrow AB, A \in V_\varepsilon \rightsquigarrow S \rightarrow B$$

$$S \rightarrow AB, B \in V_\varepsilon \rightsquigarrow S \rightarrow A$$

$$S \rightarrow A, A \in V_\varepsilon \rightsquigarrow S \rightarrow \varepsilon$$

$$S \rightarrow B, B \in V_\varepsilon \text{ but } S \rightarrow \varepsilon \in P$$

$$P = \{S \rightarrow AB, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon,$$

$$A \rightarrow a, B \rightarrow b, S \rightarrow B, S \rightarrow A, S \rightarrow \varepsilon\}$$



Упражнение a^*b^*

$$G = (\{S, A, B\}, \{a, b\}, P, S),$$

$$P = \{S \rightarrow AB, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon\}$$

$$V_\varepsilon = \{A, B, S\}$$

$$P := \{S \rightarrow AB, A \rightarrow aA, A \rightarrow \varepsilon, B \rightarrow bB, B \rightarrow \varepsilon, \\ A \rightarrow a, B \rightarrow b, S \rightarrow B, S \rightarrow A, S \rightarrow \varepsilon\}$$

$$P := P \setminus (V \times \{\varepsilon\})$$

$$P := \{S \rightarrow AB, A \rightarrow aA, B \rightarrow bB, \\ A \rightarrow a, B \rightarrow b, S \rightarrow B, S \rightarrow A\}$$

$$\text{Return } (\{S', S, A, B\}, \{a, b\}, P, S'),$$

$$P := \{S' \rightarrow \varepsilon, S' \rightarrow S, S \rightarrow AB, A \rightarrow aA, B \rightarrow bB, \\ A \rightarrow a, B \rightarrow b, S \rightarrow B, S \rightarrow A\}$$



Доказателство за коректност

- $X \in V_\varepsilon \longrightarrow X \xRightarrow{*} \varepsilon$: даден извод
- $X \xRightarrow{*} \varepsilon \longrightarrow X \in V_\varepsilon$: индукция по дължината на извода
- Инварианта на цикъла
- **Завършване !**
- Елиминация на ε -правилата не променя $L(G) \setminus \{\varepsilon\}$:
Индукция по дължината на извода. Заменяме
правилото

$$\gamma X \delta \xRightarrow{X \rightarrow \alpha Y \beta} \gamma \alpha Y \beta \delta \xRightarrow{Y \rightarrow \varepsilon} \gamma \alpha \beta \delta \text{ с}$$

$$\gamma X \delta \xRightarrow{X \rightarrow \alpha \beta} \gamma \alpha \beta \delta.$$



Доказателство за коректност — завършване

$$\text{assert } V_\varepsilon = \left\{ X \in V : X \xrightarrow{*} \varepsilon \right\}$$

while $\exists X \rightarrow \alpha Y \beta \in P : Y \in V_\varepsilon \wedge X \rightarrow \alpha \beta \notin P$ do

$$P := P \cup \{X \rightarrow \alpha \beta\}$$

Нека $k := \max \{|r| : X \rightarrow r \in P\}$.

Наблюдение: Има **нови** правила $X \rightarrow w \in P$, $|w| < k$.

Но са само краен брой правила с ограничена дължина.



Елиминация на **цикличните** единични правила

$G = (V, \Sigma, P, S)$ контекстно-свободна без ϵ -правила

Да разгледаме **графа** $U = (V, P \cap V \times V)$ на единичните правила.

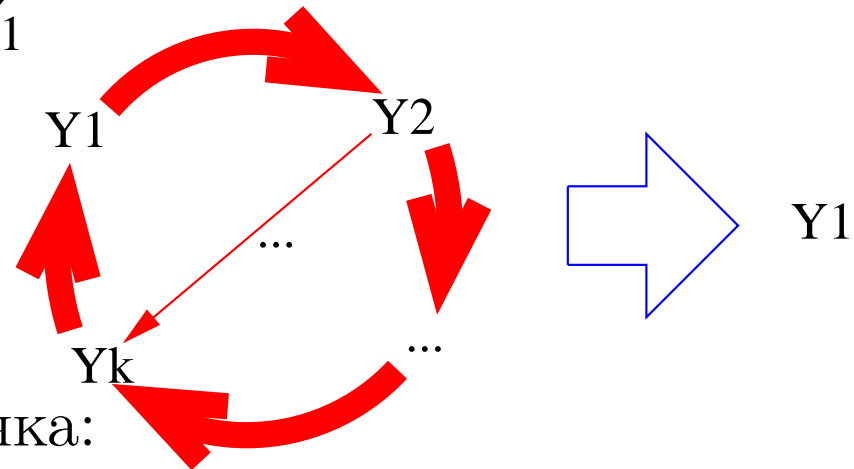
while \exists **cycle** $\{Y_1, \dots, Y_k\}$ in U , $k \geq 1$ do

invariant : $L(G)$

replace Y_2, \dots, Y_k in G с Y_1

$P := P \setminus \{Y_1 \rightarrow Y_1\}$

assert U is cycle-free



Граф-теоретична гледна точка:

контракция (редуциране) на **силно свързаните**

КОМПОНЕНТИ



Елиминация на нецикличните единични правила

invariant : Графът на единичните правила U няма цикли

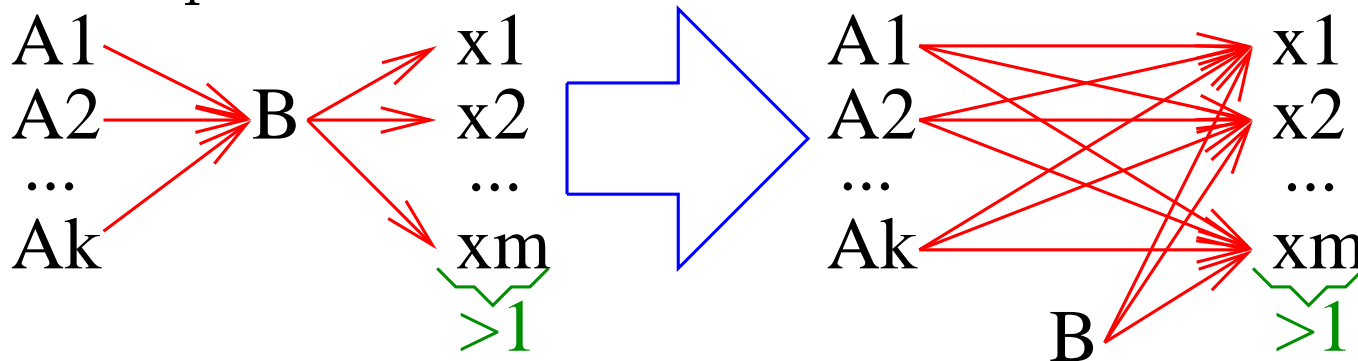
while $\exists X \rightarrow Y \in P \cap V \times V : P \cap \{Y\} \times V = \emptyset$ do

invariant : $L(G)$

$P := P \cup \{X \rightarrow x : X \rightarrow Y \in P \wedge Y \rightarrow x \in P\}$

$P := P \setminus V \times \{Y\}$

assert U е празен



Граф-теоретична гледна точка: Нарездаме ги в обратна на **топологична сортировка** ред.



Нормална форма на Чомски

Една граматика $G' = (V, \Sigma, P, S)$ е в **нормална форма на Чомски**, ако

$$P \subseteq V \times \Sigma \cup V \times VV.$$

Твърдение: За всяка контекстно-свободна граматика G с $\epsilon \notin L(G)$, $\exists G'$ в нормална форма на Чомски, такава че $L(G) = L(G')$.

Д-во на твърдението: Стъпка по стъпка променяме граматиката G .

Инвариант: $L(G)$ остава непроменен.

1. ϵ елиминирани
2. елиминирани на единичните правила.



Елиминация на смесените десни страни

foreach $a \in \Sigma$ do

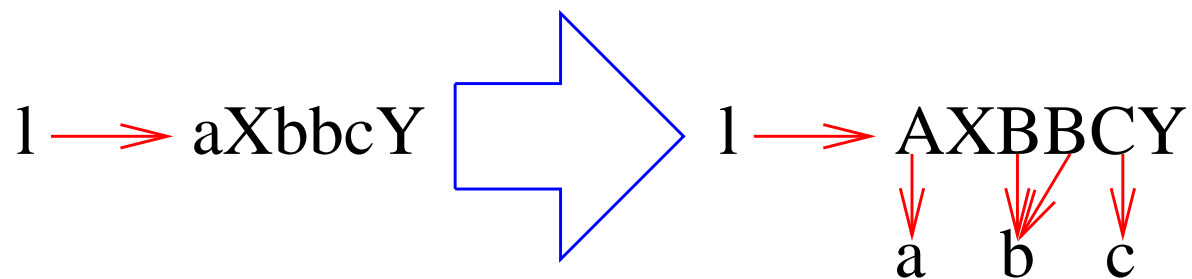
$V_a :=$ new variable

$P := P \cup \{V_a \rightarrow a\}$

foreach $\ell \rightarrow r \in P$ do

if $a \in r \wedge |r| \geq 2$ then **replace a с V_a** in $V \rightarrow r$

assert $P \subseteq V \times \Sigma \cup V \times V^*$





Елиминация на дългите десни страни

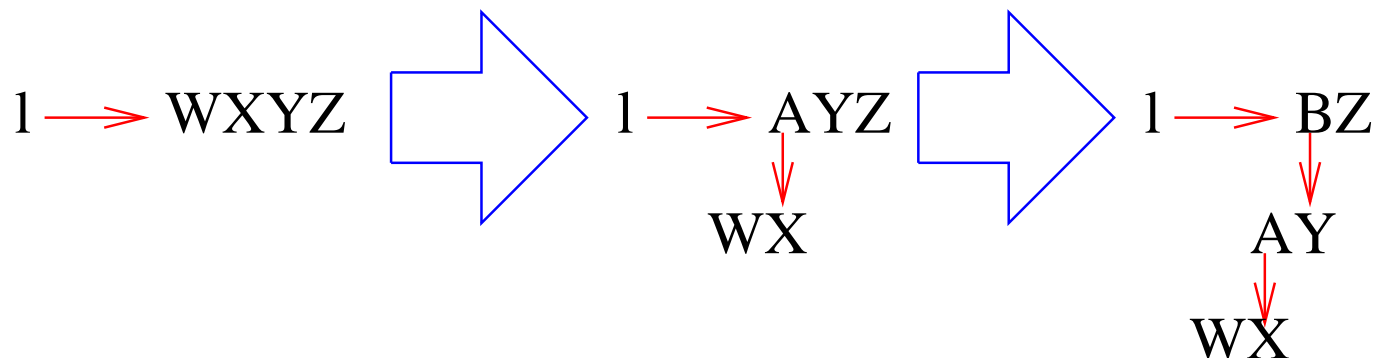
while $\exists X \rightarrow Y_1 Y_2 Y_3 \cdots Y_k \in P$ с $k \geq 3$ do

$C :=$ new variable

$P := P \cup \{C \rightarrow Y_1 Y_2, X \rightarrow C Y_3 \cdots Y_k\} \setminus \{X \rightarrow Y_1 Y_2 \cdots Y_k\}$

Цикълът завършва, тъй като $\sum_{l \rightarrow r \in P} \max(0, |r| - 2)$

намалява.





Упражнение

$$\{E \rightarrow E + E, E \rightarrow E * E, E \rightarrow (E), E \rightarrow a\}$$

\rightsquigarrow

$$\{E \rightarrow EV_+E, E \rightarrow EV_*E, E \rightarrow V_-(EV), E \rightarrow a, \\ V_+ \rightarrow +, V_* \rightarrow *, V_-(\rightarrow (, V) \rightarrow)\}$$



~→

$$\{C \rightarrow EV_+, E \rightarrow CE,$$

$$D \rightarrow EV_*, E \rightarrow DE,$$

$$F \rightarrow V_*(E, E \rightarrow FV_),$$

$$E \rightarrow a,$$

$$V_+ \rightarrow +, V_* \rightarrow *, V_((\rightarrow (, V_)) \rightarrow) \}$$



1.3.2 Pumping Лема

L е контекстно-свободен

$$\rightarrow \exists n \in \mathbb{N} : \forall z \in L : |z| > n$$

$$\rightarrow \exists u, v, w, x, y : z = uvwxу \wedge |vx| \geq 1 \wedge |vwx| \leq n \wedge$$

$$\forall i \in \mathbb{N}_0 : uv^iwx^iy \in L$$

С думи:

Достатъчно дългите думи в един **контекстно-свободен** език остават в него с итериране на една **или две** нетривиални поддуми "pumping".



Доказателство на Pumping лемата

Нека $G = (V, \Sigma, P, S)$ е граматика в **нормална форма на Чомски** за $L = \{\epsilon\}$.

Нека $k = |V|$, $n = 2^k$,

$z \in L$ с $|z| = m \geq n$ произволна.

Да разгледаме **синтактично дърво** за z .

Макс. **степен** ≤ 2 , има $\geq n = 2^k$ листа

$\rightarrow \exists$ път P с дължина $|P| > k$.

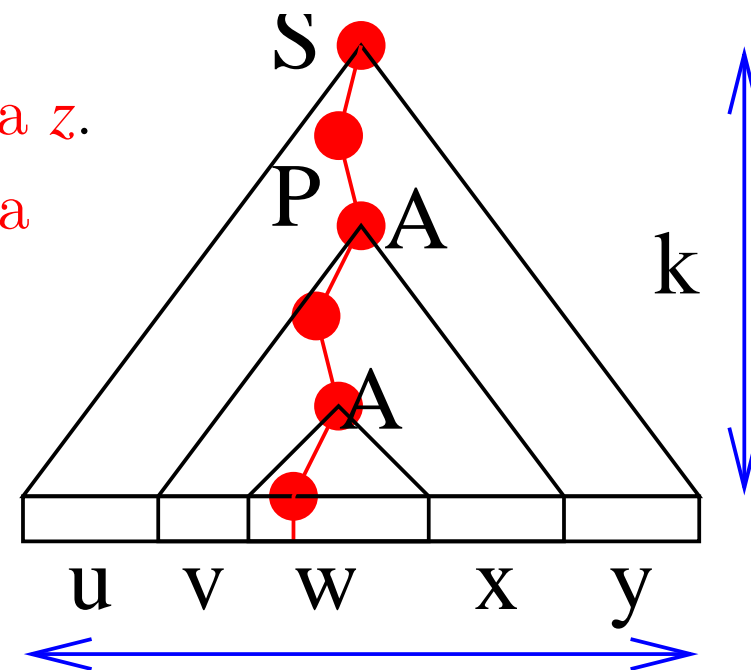
$\rightarrow \geq k + 1$ променливи в P .

$\rightarrow \exists$ променлива $A \in P$:

A се появява ≥ 2 пъти ($|vx| \geq 1$).

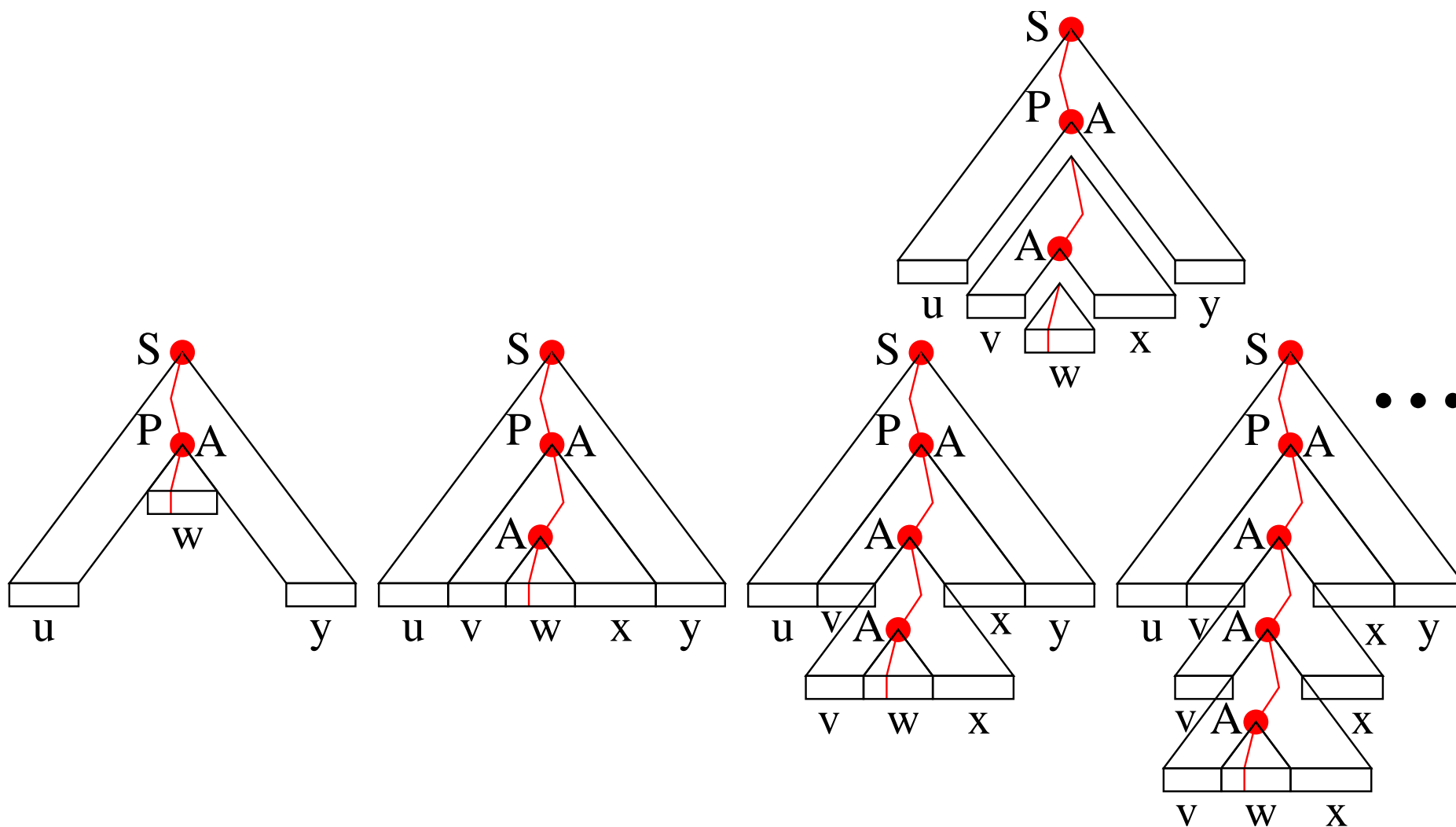
Второто появяване отдолу нагоре е на **разстояние** $\leq k$.

\rightarrow корен на поддърво за поддумата vw^kx с дължина $\leq n$





Конструкция на итерациите:





Лема: Едно двоично дърво (възлите със степен 0 или 2) с $\geq 2^k$ листа съдържа път P с дължина $\geq k$.

Д-во: Индукция по k .

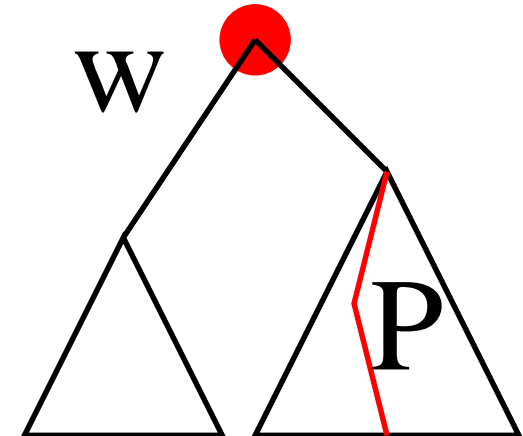
Случай $k = 0$: 1 листо, дължина на път 0.

Случай $k \rightsquigarrow k + 1$:

$\geq 2^{k+1}$ листа. Коренът w има степен 2.

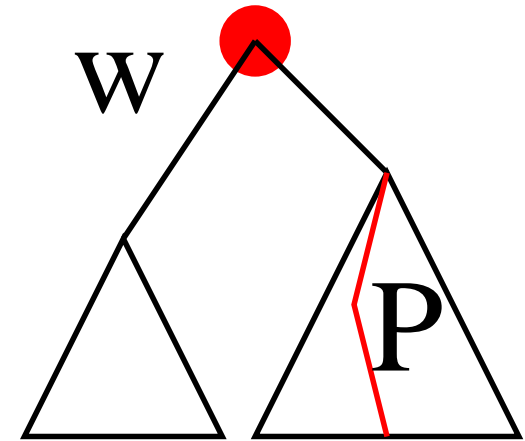
Поне едно поддърво има $\geq 2^k$ листа и съдържа път P с дължина $\geq k$.

Така wP има дължина $\geq k + 1$.





Следствие: Ако в едно двоично дърво (възлите със степен 0 или 2) всеки път е с дължина $\leq k$, то има $\leq 2^k$ листа.





$L = \{a^m b^m c^m : m \geq 1\}$ не е контекстно-свободен

Допускаме, че L е контекстно-свободен.

Нека n е числото от Pumping лемата.

Да разгледаме $z = a^n b^n c^n$ и представянето

$z = uvwxu$ от Pumping лемата като

$|vwx| \leq n$, $|vx| \geq 1$, $uv^0wx^0u = uvu \in L$.

vx не може да съдържа a -тата, b -тата и c -тата.

→ балансът на a -тата, b -тата и c -тата в uvu е нарушен.

→ $uvu \notin L$

Противоречие.

Следствие: тип 2 ≠ тип 1



$L = \{ww : w \in \{a, b\}^*\}$ не е контекстно-свободен

Допускаме, че L е контекстно-свободен.

Нека n е числото от Pumping лемата.

Да разгледаме $z = a^n b^n a^n b^n$ и декомпозицията

$z = uvwxu$ от Pumping лемата с $|vwx| \leq n$, $|vx| \geq 1$,

$z' := uvw \in L$.

Случай $vx = a^k b^j$ е в лявата половина:

$$\longrightarrow z' = a^{n-k} b^{n-j} a^n b^n.$$

Противоречие.

Случай vx лежи в дясната половина: (аналогично)

Случай vx лежи в средата: $vx = b^k a^j$

$$\longrightarrow z' = a^n b^{n-k} a^{n-j} b^n.$$

Противоречие.



Правила за д-во с Pumping Лемата

1. Нека n е числото от Pumping лемата.
2. Да разгледаме $z = ???$ ($|z| \geq n$) и представянето $z = uvwxu$ от Pumping лемата с $|vwx| \leq n$, $|vx| \geq 1$
 - Всяка дума z с $|z| \geq n$. “Изобретателната” част !
 - Изборът на думата — д-вото да е просто
 - Тъй като $|vwx| \leq n$ съдържа блокове с дължина n имаме следните случаи
3. Случаи за **всички възможни декомпозиции** $z = uvwxu$. За всеки случай: Намираме $i \geq 0$, такава че $uv^iwx^i u \notin L(G)$.

Типични стойности: $i = 0$, $i = 2$.

Предизвикателство: Броят на случаите по-малък.



1.3.3 Свойства на затвореност

Контекстно свободните езици са затворени относно

- обединение
- конкатенация
- операцията звезда

\cup

\cdot

$*$

не са затворени относно

- сечение
- допълнение

\cap

$\bar{}$



Затвореност на CFG относно \cup

Да разгледаме

$$G_1 = (V_1, \Sigma, P_1, S_1),$$

$$G_2 = (V_2, \Sigma, P_2, S_2),$$

Нека $V_1 \cap V_2 = \emptyset$ и

$$G = (\{S\} \cup V_1 \cup V_2, \Sigma, \{S \rightarrow S_1, S \rightarrow S_2\} \cup P_1 \cup P_2).$$

Очевидно имаме

$$L(G) = L(G_1) \cup L(G_2).$$



Затвореност на CFG относно \cdot

Да разгледаме

$$G_1 = (V_1, \Sigma, P_1, S_1),$$

$$G_2 = (V_2, \Sigma, P_2, S_2),$$

Нека $V_1 \cap V_2 = \emptyset$ и

$$G = (\{S\} \cup V_1 \cup V_2, \Sigma, \{S \rightarrow S_1 S_2\} \cup P_1 \cup P_2).$$

Ясно е, че

$$L(G) = L(G_1) \cdot L(G_2).$$



Затвореност на CFG относно *

Да разгледаме

$$G_1 = (V_1, \Sigma, P_1, S_1)$$

и нека S_1 не участва в дясните страни на P . И

$$G = (\{S\} \cup V_1, \Sigma, \{S \rightarrow \varepsilon, S \rightarrow S_1, S_1 \rightarrow S_1 S_1\} \cup P_1 \setminus \{S_1 \rightarrow \varepsilon\}).$$

Тогава

$$L(G) = L(G_1)^*.$$



НЕзатвореност на CFG относно \cap

Да разгледаме контекстно свободните езици

$$L_1 = \{a^i b^j c^j : i, j > 0\}$$

$$L_2 = \{a^i b^i c^j : i, j > 0\}.$$

$$L_1 \cap L_2 = \{a^i b^i c^i : i > 0\} \text{ не е контекстно-свободен!}$$



НЕзатвореност на CFG относно $\bar{\cdot}$

Да допуснем:

затвореност относно \cup и $\bar{\cdot}$.

→

затвореност относно \cap .

$$L_1 \cap L_2 = \overline{\overline{L_1} \cup \overline{L_2}}$$

Противоречие



1.3.4 СҮК-алгоритъм

— Проблемът за принадлежност на дума за контекстно-свободни граматики

Дадено: Една граматика $G = (V, \Sigma, P, S)$,
и дума $x = x_1 \cdots x_n \in \Sigma^*$.

Въпрос: $x \in L(G)$?

Алгоритъм на Cocke, Younger и Kasami

Нека G е в нормална форма на Чомски:

Специалният случай $x = \varepsilon$ е лесен,

тъй като предварително е превирната в нормална форма на Чомски.



СУК алгоритъм

Ние решаваме по-общ проблем:

За всяка поддума $x_i \cdots x_{i+j-1}$ на x с дължина j ,
от кои променливи е изводима $x_i \cdots x_{i+j-1}$?

$$T[i, j] := \left\{ A \in V : A \stackrel{*}{\Rightarrow} x_i \cdots x_{i+j-1} \right\}$$

Случай $j = 1$: $T[i, 1] = \{A \in V : A \rightarrow x_i \in P\}$

В противен случай:

$$T[i, j] := \left\{ A \in V : \exists A \rightarrow BC \in P : \exists k \in \{1, \dots, j-1\} : \right. \\ \left. B \in T[i, k] \wedge C \in T[i+k, j-k] \right\}$$

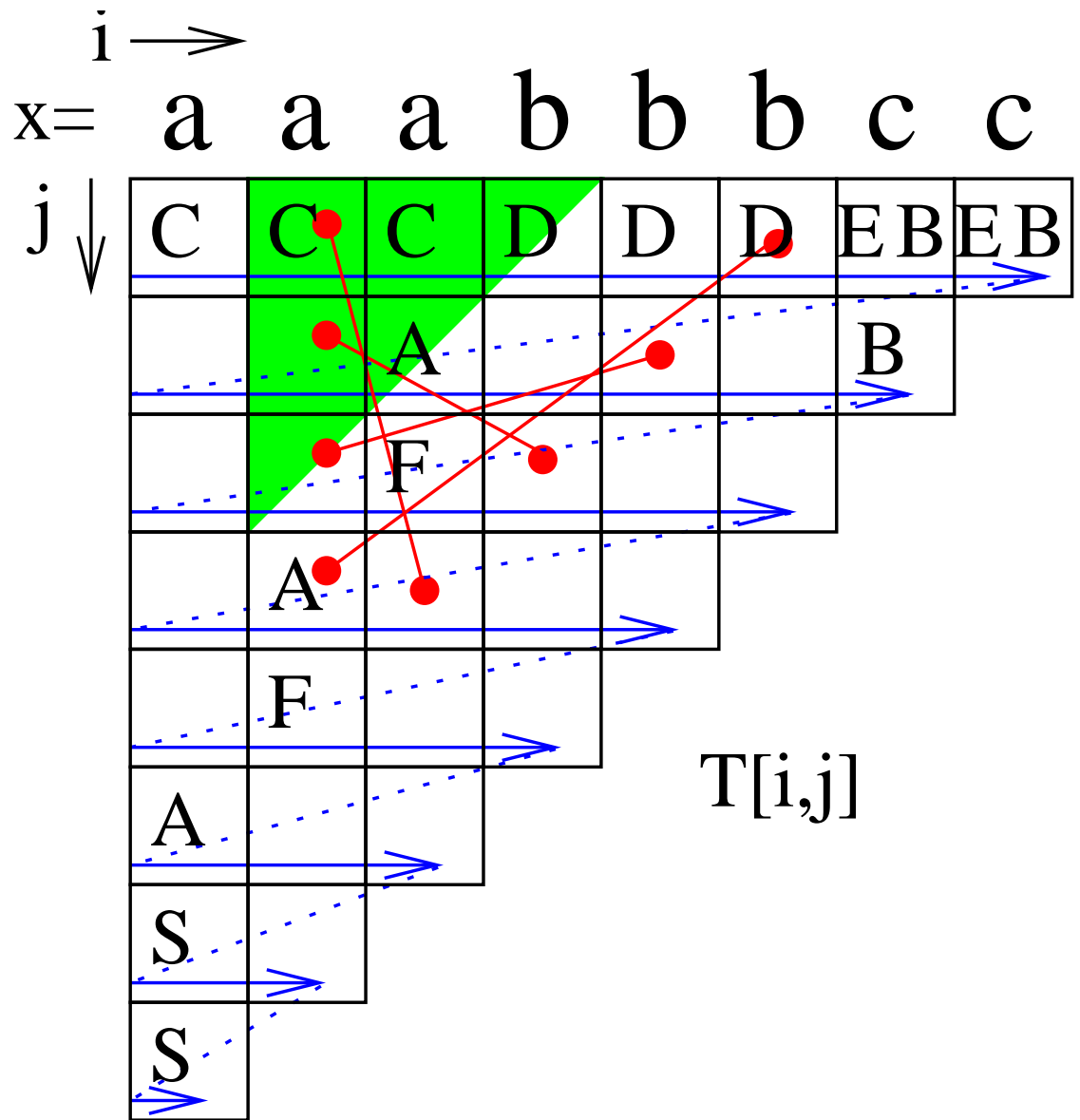
Накрая: $S \in T[1, n]$?



Упражнение

$G = (\{S, A, B, C, D, E, F\},$
 $\{a, b, c\}, P, S),$

- $P = \{$
- $S \rightarrow AB,$
- $A \rightarrow CD,$
- $A \rightarrow CF,$
- $B \rightarrow c,$
- $B \rightarrow EB,$
- $C \rightarrow a,$
- $D \rightarrow b,$
- $E \rightarrow c,$
- $F \rightarrow AD\}$





Приложение на динамичното програмиране

```

for  $i := 1$  to  $n$  do  $T[i, 1] := \{A \in V : A \rightarrow x_i \in P\}$ 
for  $j := 2$  to  $n$  do
  for  $i := 1$  to  $n - j + 1$  do
     $T[i, j] := \emptyset$ 
    for  $k := 1$  to  $j - 1$  do
       $T[i, j] \leftarrow \{A : \exists A \rightarrow BC \in P : B \in T[i, k] \wedge C \in T[i+k, j-k]\}$ 
return  $S \in T[1, n]$ 

```

Лема След $j - 1$ изпълнения на цикъла по j , $j \geq 1$,
за всяко $i = 1, \dots, n - j + 1$:

$$T[i, j] := \left\{ A \in V : A \xrightarrow{*} x_i \cdots x_{i+j-1} \right\}.$$



Анализ

Нека $V = 1..|V|$

```

for  $i := 1$  to  $n$  do  $T[i, 1] := \{A \in V : A \rightarrow x_i \in P\}$  // “евтино”
for  $j := 2$  to  $n$  do //  $\leq n$  ПЪТИ
    for  $i := 1$  to  $n - j + 1$  do //  $\leq n$  ПЪТИ
         $T[i, j] := \emptyset$  // Двоичен вектор с големина  $|V| \leq |P|$ 
        for  $k := 1$  to  $j - 1$  do //  $\leq n$  ПЪТИ
            foreach  $A \rightarrow BC \in P$  do //  $\leq |P|$  ПЪТИ
                if  $B \in T[i, k] \wedge C \in T[i + k, j - k]$  then //  $\mathcal{O}(1)$ 
                    добавяме  $A$  към  $T[i, j]$  //  $\mathcal{O}(1)$ 
return  $S \in T[1, n]$ 

```

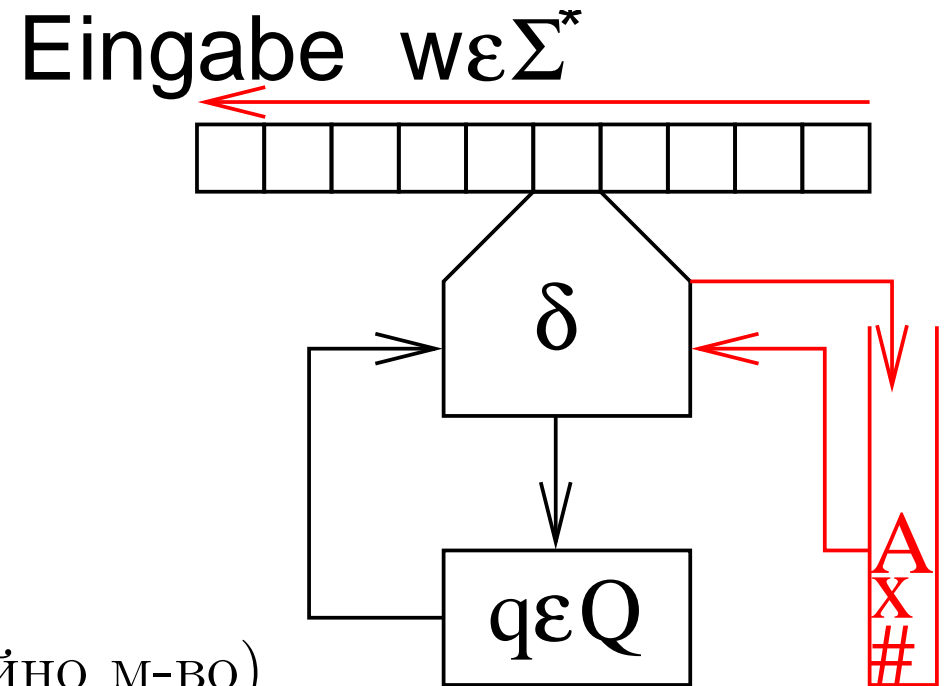
Време: $\mathcal{O}(n \times n \times (|V| + n \times |P|)) = \mathcal{O}(n^3 |P|)$



1.3.5 Стекови автомати

$K = (Q, \Sigma, \Gamma, \delta, s, \#)$:

- Q , състояния
- Σ , азбука
- Γ азбука за стека,
 $\Sigma \cup \{\#\} \subseteq \Gamma$
- $\delta : Q \times (\Sigma \cup \varepsilon) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$,
функция на прехода; (крайно м-во)
- $s \in Q$, начално състояние
- $\# \notin \Sigma$: край на стека,



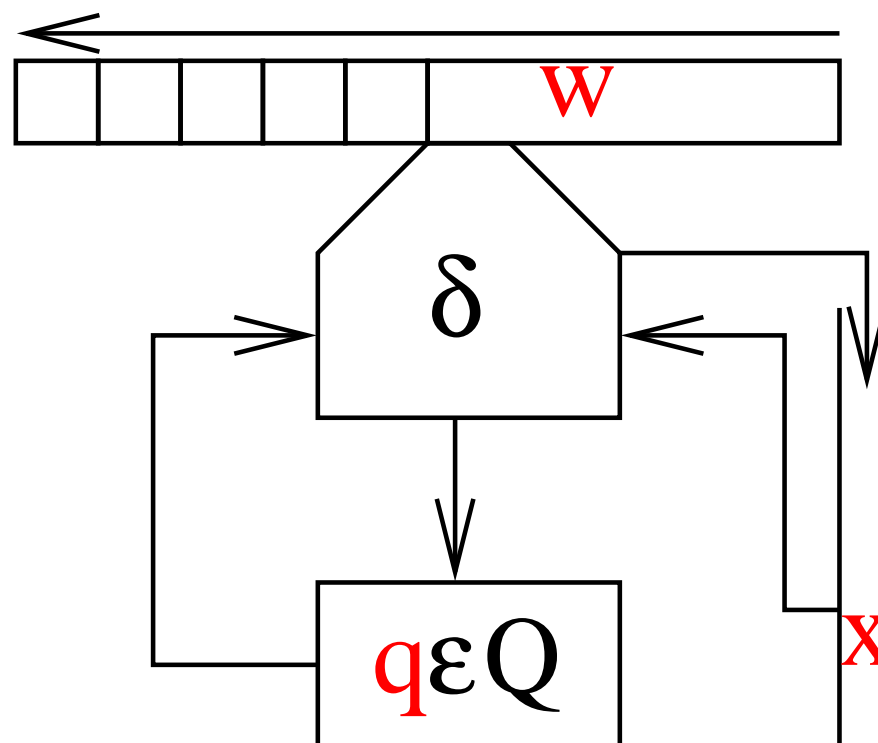
$\approx \varepsilon\text{NFA} + \text{стекова памет} - \text{крайните състояния}$

$\approx \text{специална 2-лентова-машина на Тюринг}$



Конфигурация на стеков автомат

$$(q, w, x) \in Q \times \Sigma^* \times \Gamma^*$$





Как работи един стеков автомат

Възможни преходи между конфигурации.

Четене на един входен символ a :

$$(q, aw, bx) \stackrel{(q', x') \in \delta(q, a, b)}{\vdash} (q', w, x'x)$$

ϵ -преход:

$$(q, w, bx) \stackrel{(q', x') \in \delta(q, \epsilon, b)}{\vdash} (q', w, x'x)$$

$$(q, w, \alpha) \vdash^* (q', w, \beta) \iff$$

$$\exists C_1 \dots C_n (C_1 = (q, w, \alpha) \ \& \ C_n = (q', w, \beta) \ \& \ C_1 \vdash C_2 \dots C_{n-1} \vdash C_n)$$

$$(q, w, \alpha) \vdash^* (q, w, \alpha)$$



Стековите автомати като разпознаватели

$$K = (Q, \Sigma, \Gamma, \delta, s, \#).$$

$$L(K)?$$

Дефиниция:

K приема $w \in \Sigma^*$ т.т.к.

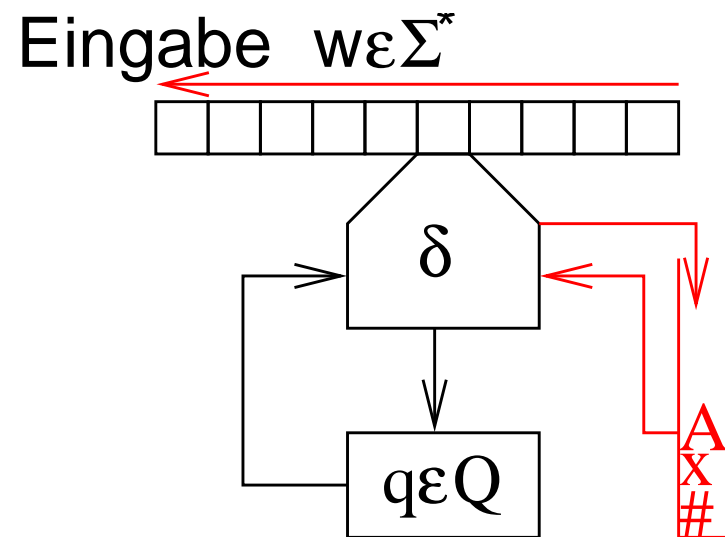
\exists редица от (**разрешени** от δ)

конфигурации

$(s, w, \#) \vdash \dots \vdash (q, \varepsilon, \varepsilon)$ с $q \in Q$ произволно.

“Приемане с празен стек”

$$L(K) := \{w \in \Sigma^* : K \text{ приема } w\}.$$





Упражнение: $\{w\$w^R : w \in \{a,b\}^*\}$

$$K = (\{0, 1\}, \{a, b, \$\}, \{a, b, \#\}, \delta, 0, \#)$$

$$\delta(0, \$, k) = \{(1, k)\}$$

$$\delta(0, i, k) = \{(0, ik)\} \text{ за } i \in \{a, b\}$$

$$\delta(1, i, i) = \{(1, \varepsilon)\}$$

$$\delta(1, \varepsilon, \#) = \{(1, \varepsilon)\}$$

$$(0, ba\$ab, \#) \vdash$$

$$(0, a\$ab, b\#) \vdash$$

$$(0, \$ab, ab\#) \vdash$$

$$(1, ab, ab\#) \vdash$$

$$(1, b, b\#) \vdash$$

$$(1, \varepsilon, \#) \vdash$$

$$(1, \varepsilon, \varepsilon)$$



Упражнение: $\{ww^R : w \in \{a,b\}^*\}$

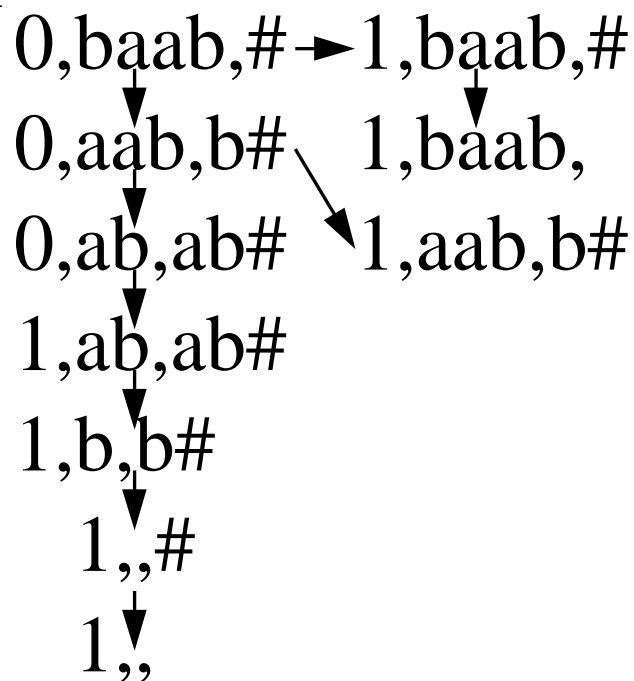
$$K = (\{0, 1\}, \{a, b\}, \{a, b, \#\}, \delta, 0, \#)$$

$$\delta(0, \varepsilon, k) = \{(1, k)\}$$

$$\delta(0, i, k) = \{(0, ik)\}$$

$$\delta(1, i, i) = \{(1, \varepsilon)\}$$

$$\delta(1, \varepsilon, \#) = \{(1, \varepsilon)\}$$





Твърдение: L е контекстно-свободен т.т.к. \exists
недетерминистичен стеков автомат (NstackA)

$$M : L(M) = L$$



Д-во: L е контекстно-свободен $\longrightarrow \exists \text{NstackA } M : L(M) = L$

Нека $G = (V, \Sigma, P, S)$ е граматика с $L(G) = L$.

Да разгледаме $\text{NstackA } M = (\{q\}, \Sigma, V \cup \Sigma, \delta, q, S)$ с

$$(1) \forall A \rightarrow \alpha \in P : (q, \alpha) \in \delta(q, \varepsilon, A),$$

$$(2) \forall a \in \Sigma : (q, \varepsilon) \in \delta(q, a, a)$$

Идея (инварианта): в стека се запомнят правилата на един извод, и при вход на терминален символ го вади от върха на стека.



$G = (V, \Sigma, P, S)$ граматика с $L(G) = L$.

$M = (\{q\}, \Sigma, V \cup \Sigma, \delta, q, S)$ с

$\forall A \rightarrow \alpha \in P : (q, \alpha) \in \delta(q, \varepsilon, A), \forall a \in \Sigma : (q, \varepsilon) \in \delta(q, a, a)$

Лема: Ако $w \in \Sigma^*, \alpha \in V(V \cup \Sigma)^* \cup \{\varepsilon\}$, то

$$S \Rightarrow^* w\alpha \iff (q, w, S) \vdash^*(q, \varepsilon, \alpha).$$

Следствие: $\alpha = \varepsilon \implies S \Rightarrow^* w \iff (q, w, S) \vdash^*(q, \varepsilon, \varepsilon)$.

Д-во: Нека $S \Rightarrow^* w\alpha$. Тогава има извод

$$u_0 = S \Rightarrow u_1 \cdots \Rightarrow u_n = w\alpha.$$

С индукция по дължината на извода ще покажем, че

$$(q, w, S) \vdash^*(q, \varepsilon, \alpha).$$



$$S \Rightarrow^* w\alpha \longrightarrow (q, w, S) \vdash^* (q, \varepsilon, \alpha)$$

Случай $n = 0$:

$$u_0 = S = w\alpha \longrightarrow w = \varepsilon \ \& \ \alpha = S \longrightarrow (q, w, S) \vdash^* (q, \varepsilon, \alpha).$$

Случай $n \rightsquigarrow n + 1$:

$$u_0 = S \Rightarrow u_1 \dots u_n \Rightarrow u_{n+1} = w\alpha. \text{ Нека } u_n = xA\beta, \ x \in \Sigma^*, \\ u_{n+1} = x\gamma\beta \text{ и } A \rightarrow \gamma \in P.$$

Тъй като $u_0 = S \Rightarrow u_1 \dots \Rightarrow u_n = xA\beta$, то по ИП
 $(q, x, S) \vdash^* (q, \varepsilon, A\beta)$.

$$\text{От } A \rightarrow \gamma \in P \longrightarrow (q, \varepsilon, A\beta) \vdash (q, \varepsilon, \gamma\beta).$$

Но $u_{n+1} = w\alpha = x\gamma\beta$, α започва с променлива и $x, w \in \Sigma^*$.

Следователно $w = xy$, $y \in \Sigma^*$, $y\alpha = \gamma\beta$. Така

$$(q, w, S) \vdash^* (q, y, \gamma\beta) = (q, y, y\alpha). \text{ Сега прилагайки } y \text{ пъти} \\ \text{преходи от тип (2): } (q, w, S) \vdash^* (q, y, y\alpha) \vdash^* (q, \varepsilon, \alpha).$$



$$(q, w, S) \vdash^* (q, \varepsilon, \alpha) \longrightarrow S \Rightarrow^* w\alpha$$

Индукция по броя n на преходите от тип (1).

Случай $n = 0$:

$$(q, w, S) \vdash^* (q, \varepsilon, \alpha) \longrightarrow w = \varepsilon, \alpha = S \longrightarrow S \Rightarrow^* w\alpha.$$

Случай $n \rightsquigarrow n + 1$:

$$(q, w, S) \stackrel{n(1)}{\vdash^*} (q, y, A\beta) \vdash (q, y, \gamma\beta) \stackrel{(2)}{\vdash^*} (q, \varepsilon, \alpha).$$

$$w = xy, A \rightarrow \gamma \in P, \gamma\beta = y\alpha!.$$

$$(q, x, S) \stackrel{n(1)}{\vdash^*} (q, \varepsilon, A\beta).$$

$$\text{По ИП } S \Rightarrow^* xA\beta \Rightarrow x\gamma\beta = xy\alpha = w\alpha \longrightarrow S \Rightarrow^* w\alpha.$$



Твърдение: За всеки $NstackA$ с едно състояние

$$M = (\{z\}, \Sigma, \Gamma, \delta, z)$$

има граматика G с $L(G) = L(M)$.

Д-во: Нека $G = (\Gamma, \Sigma, P, \#)$ с

$$P = \{A \rightarrow a\alpha : (z, \alpha) \in \delta(z, a, A)\} \quad a = \varepsilon \text{ разрешено!}$$

$L(G) \subseteq L(M)$:

$$x \in L(G) \longrightarrow \exists \text{ ЛЯВ ИЗВОД } A = \# \xRightarrow{*} x.$$

За всяко начало w на x , $x = wu$, за извода $\# \xRightarrow{*} w\alpha$ с

$$w \in \Sigma^*, \alpha \in \Gamma^*$$

\exists редица от **конфигурации** $(z, x, \#) \vdash \cdots \vdash (z, u, \alpha)$ на M .



$$\# \xRightarrow{*} w\alpha \Leftrightarrow (z, x, \#) \vdash \cdots \vdash (z, y, \alpha)$$

Индукция по w .

База: $w = \varepsilon$. От $\# \xRightarrow{*} \alpha$ следва $y = x, \alpha = \#(\alpha \in \Gamma^*)$.

Следователно $(z, x, \#) \vdash^*(z, y, \alpha)$.

Индукционна стъпка: wa

$$\# \xRightarrow{*} wA\beta \Rightarrow wa\alpha\beta. \quad a \in \varepsilon \cup \Sigma.$$

По ИП \exists редица от конфигурации $(z, x, \#) \vdash \cdots \vdash (z, ay, A\beta)$
с $way = x$.

Тъй като $A \rightarrow a\alpha \in P$ трябва $(z, \alpha) \in \delta(z, a, A)$.

Така, $(z, ay, A\beta) \vdash (z, y, \alpha\beta)$

$(z, x, \#) \vdash^*(z, y, \alpha\beta)$.

Следователно $\# \xRightarrow{*} x \longrightarrow (z, x, \#) \vdash^*(z, \varepsilon, \varepsilon)$

$(w = x \Rightarrow y = \varepsilon, \alpha = \varepsilon)$.



$L(M) \subseteq L(G)$:

$x \in L(M) \longrightarrow \exists$ редица от конфигурации

$(z, x, \#) \vdash \cdots \vdash (z, \varepsilon, \varepsilon)$

На това съответства **ЛЯВ ИЗВОД** $A = \# \xRightarrow{*} x$.

За всяко начало w на x и $x = wu$

От $(z, x, \#) \vdash^* (z, y, \alpha)$ следва, че съществува извод $\# \xRightarrow{*} w\alpha$.

Аналогично.

Така за $w = x$ имаме :

$(z, x, \#) \vdash^* (z, \varepsilon, \varepsilon) \Rightarrow \# \xRightarrow{*} x$.



1.3.6 Детерминистични контекстно-свободни езици

$K = (Q, \Sigma, \Gamma, \delta, s, \#, F)$:

□ $Q, \Sigma, \Gamma, s, \#$ знаем.

□ $\delta : Q \times (\Sigma \cup \epsilon) \times \Gamma \rightarrow 2^{Q \times \Gamma^*}$, където

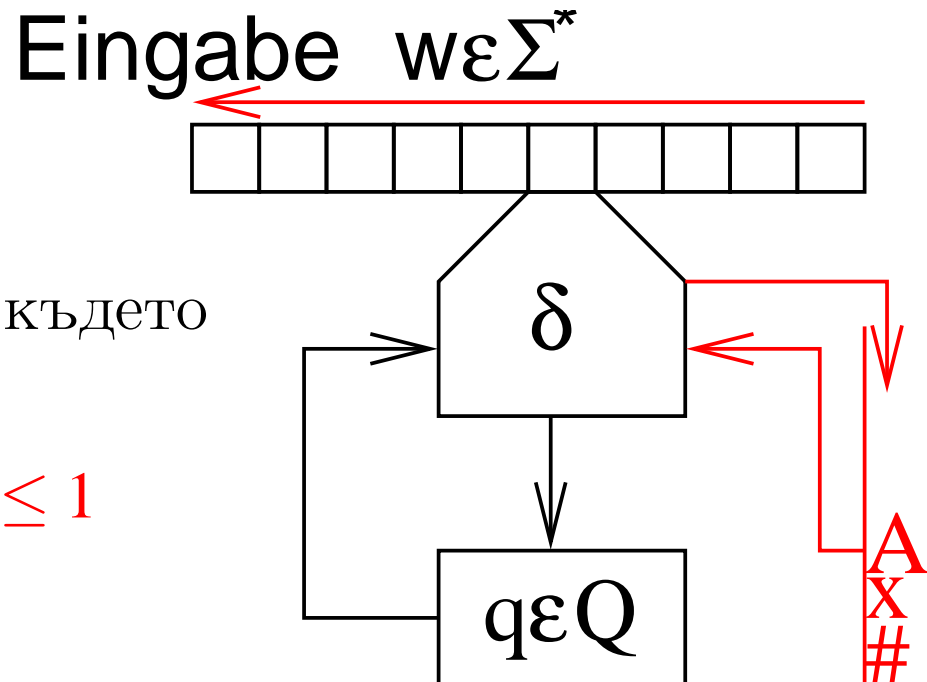
$\forall z \in Q, a \in \Sigma, A \in \Gamma$:

$$|\delta(z, a, A)| + |\delta(z, \epsilon, A)| \leq 1$$

K приема $w \in \Sigma^*$, ако

\exists редица от конфигурации (**допустими от δ**)

$(s, w, \#) \vdash \dots \vdash (f, \epsilon, \epsilon)$ с $f \in F$.





Твърдение:

$\forall \text{DstackA } K : \exists \text{NstackA}$ с едно състояние
 $K' : L(K) = L(K')$.

$\exists \text{NstackA}$ с едно състояние $K' : \nexists \text{DstackA}$
 $K : L(K) = L(K')$.



Компилатори

Наблюдение:

Проблемът за принадлежност на дума за $DstackA$ може да се разреши за **линейно време**.

Забележка:

Езиците разпознаващи се от $DstackA$ са точно **$LR(k)$** -езици.

Тези езикови фамилии са основата на синтаксиса на много програмни езици.



Свойства на затвореност за DstackA

Без Д-во:

Затворени относно: $\bar{\cdot}$, сечение с ргулярен език.

Не са затворени относно: $\cup, \cap, \cdot, *$



1.3.7 Разрешими проблеми за контекстно-свободни езици

Проблемът за празнота

Function isEmpty($G = (V, \Sigma, P, S)$)

 Marked := Σ

 while $\exists A \rightarrow \alpha \in P : A \notin \text{Marked} \wedge \alpha \in \text{Marked}^*$ do

 Marked := Marked $\cup \{A\}$

 return $S \notin \text{Marked}$

Изпълнено е т.т.к. $L(G) = \emptyset$



Проблемът за крайност

Дадено: граматика $G = (V, \Sigma, P, S)$

Въпрос: $|L(G)| < \infty$?

Нека n е числото от Pumping лемата.

Наблюдение: $|L(G)| = \infty \Leftrightarrow \exists z \in L(G) : n \leq |z| < 2n$

Д-во:

$z \in L(G), n \leq |z| < 2n \longrightarrow$ Pumping лемата дава $|L| = \infty$.

Случай $|L(G)| = \infty$ да разгледаме $z \in L(G)$ с минимална $|z| \geq n$.

Да допуснем, че $|z| \geq 2n$.

Pumping лемата $\longrightarrow z = uvwxu, |vx| \geq 1, uvu \in L(G), |uvu| \geq n$.

Противоречие с минималността на $|z|$.



Проблемът за крайност

Дадено: граматика $G = (V, \Sigma, P, S)$

Въпрос: $|L(G)| < \infty$?

Нека n е числото от Pumping лемата.

Изпълнено е: $|L(G)| = \infty \Leftrightarrow \exists z \in L(G) : n \leq |z| < 2n$

Пълно изчепване алгоритъм:

Проблемът за принадлежността за всички думи z с дължина $n \leq |z| < 2n$.

Време: $\mathcal{O}\left(8 \cdot 2^{3|V|} \cdot |\Sigma|^{2 \cdot 2^{|V|}}\right)$

!

Забележка: Има и по-ефективен алгоритъм.



Разрешими проблеми за DstackA

- Еквивалентност: $L(K_1) = L(K_2)$?



Неразрешими проблеми за CFG

- $L(G_1) \cap L(G_2) = \emptyset$? Нямаат общи елементи
- $|L(G_1) \cap L(G_2)| = \infty$?
- $L(G_1) \cap L(G_2)$ контекстно-свободен?
- $L(G_1) \subseteq L(G_2)$?
- $L(G_1) = L(G_2)$?
- Неоднозначност: $\exists x \in L(G) : |\{\text{syntax tree}(x)\}| \geq 2$
- Дали $\overline{L(G)}$ е контекстно-свободен?
- Дали $L(G)$ е регулярен?
- Дали $L(G)$ е дет. контекстно-свободен?



Неразрешимост на $L(G_1) \cap L(G_2) = \emptyset$?

Разрешаваме РСР- Post Correspondence System

$K = (x_1, y_1) \cdots (x_k, y_k) \in (\{a, b\}^* \times \{a, b\}^*)^*$ с помощта на $\text{disjoint}(G_1, G_2)$:

Дефинираме $\Sigma = \{a, b, 1, \dots, k\}$

$G_1 = (\{S\}, \Sigma, P_1, S)$,

$G_2 = (\{S\}, \Sigma, P_2, S)$, където

$P_1 = \{S \rightarrow iSx_i, S \rightarrow ix_i : i \in 1..k\}$

$P_2 = \{S \rightarrow iSy_i, S \rightarrow iy_i : i \in 1..k\}$

$L(G_1) = \{i_n \cdots i_1 x_{i_1} \cdots x_{i_n} : i_\ell \in 1..k\}$

$L(G_2) = \{i_n \cdots i_1 y_{i_1} \cdots y_{i_n} : i_\ell \in 1..k\}$

$L(G_1) \cap L(G_2) \neq \emptyset$

Т.Т.К..

$\exists i_1, \dots, i_n : x_{i_1} \cdots x_{i_n} = y_{i_1} \cdots y_{i_n}$ т.т.к. K има решение.