

8. Chomského normální forma a CYK

Formální jazyky a automaty

Jiří Balun

Obsah

1 Chomského normální forma (CNF)

- Popis CNF
- Odstranění počátečního neterminálu z pravé strany pravidel
- Odstranění ε -pravidel
- Odstranění jednoduchých pravidel
- Odstranění neterminálů, které nic nederivují
- Odstranění nedosažitelných symbolů
- Odstranění pravidel v nesprávném tvaru

2 Algoritmus CYK

- Popis algoritmu CYK
- Příklad

Chomského normální forma (CNF)

Popis CNF

Intuice: každou CFG lze převést do speciálního tvaru, který nám umožní snadněji generovat řetězce a dokazovat další výsledky.

Definice

CFG $\mathcal{G} = (N, \Sigma, P, S)$ je v **Chomského normální formě** (CNF z *Chomsky normal form*) pokud všechna pravidla jsou v jednom z těchto tvarů (kde $A, B, C \in N$ a $a \in \Sigma$):

- $A \rightarrow BC$ (neterminál generuje dva neterminály),
 - $A \rightarrow a$ (neterminál generuje jeden terminál),
 - $S \rightarrow \varepsilon$, kde se S nevyskytuje na pravé straně jiného pravidla.
-
- **důležitá vlastnost:** každý derivační strom gramatiky v CNF je binární
 - kromě CNF existují i další normální formy, například **Greibachova normální forma**
 - její pravidla jsou ve tvaru $A \rightarrow aA_1 \dots A_n$, kde $n \geq 0$, tj. pravá strana pravidel začíná terminálem, za kterým může následovat libovolný počet neterminálů
 - tato forma usnadňuje výběr dalšího pravidla při generování/ověřování

Popis CNF

Věta

Ke každé CFG $\mathcal{G}$ existuje CFG $\mathcal{G}'$ v CNF taková, že $L(\mathcal{G}) = L(\mathcal{G}')$.

Idea důkazu

Převědeme G do CNF v těchto krocích:

- 1 odstraníme počáteční neterminál z pravé strany všech pravidel,
- 2 odstraníme ε -pravidla,
- 3 odstraníme jednoduchá pravidla,
- 4 odstraníme neterminály, které nic nederivují,
- 5 odstraníme nedosažitelné symboly,
- 6 odstraníme zbývající pravidla v nesprávném tvaru.

Výsledkem tohoto procesu je gramatika $\mathcal{G}'$ taková, že $L(\mathcal{G}) = L(\mathcal{G}')$. □

- záleží na pořadí jednotlivých úprav (např. krok 1 vytváří jednoduché pravidlo, proto musí předcházet kroku 3)

Odstranění počátečního neterminálu z pravé strany

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N', \Sigma, P', S')$, kde S' není na pravé straně jiného pravidla

- 1 Pokud se počáteční neterminál S vyskytuje na pravé straně nějakého pravidla, vytvoříme nový neterminál S' a přidáme nové pravidlo $S' \rightarrow S$ do $\mathcal{G}'$.
- 2 Zvolíme S' jako nový počáteční neterminál $\mathcal{G}' = (N', \Sigma, P', S')$.

Příklad

Mějme gramatiku $\mathcal{G} = (\{S\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid a \mid b \mid aSa \mid bSb$$

ke které vytvoříme $\mathcal{G}' = (\{S, S'\}, \{a, b\}, P', S')$ s pravidly:

$$S' \rightarrow S$$

$$S \rightarrow \varepsilon \mid a \mid b \mid aSa \mid bSb$$

Odstranění ε -pravidel

Intuice: postupně upravujeme pravidla gramatiky tak, aby ε -pravidlo „probublalo“ až do startovního neterminálu (kde je povoleno).

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N, \Sigma, P', S)$ bez ε -pravidel

- 1 Inicializace: položíme $\mathcal{G}' = \mathcal{G}$.
- 2 Odebereme z $\mathcal{G}'$ libovolné pravidlo ve tvaru $A \rightarrow \varepsilon$, kde A není počáteční neterminál.
- 3 Dále pro každé pravidlo $B \rightarrow \alpha$ obsahující A na pravé straně:
 - přidáme do $\mathcal{G}'$ nová pravidla ve tvaru $B \rightarrow \beta$, kde β je řetězec α , ve kterém každý výskyt A může být nahrazen ε (např. pro $B \rightarrow xAyAz \in P$ přidáme ještě $B \rightarrow xyAz \mid xAyz \mid xyz$),
 - pokud je pravidlo ve tvaru $B \rightarrow A$, pak přidáme nové pravidlo $B \rightarrow \varepsilon$, ale jen v případě, že jsme takové pravidlo již neodstranili (jinak by mohlo dojít k zacyklení).
- 4 Kroky 2 a 3 opakujeme, dokud neodstraníme všechna ε -pravidla mimo počáteční neterminál (pravidlo $S \rightarrow \varepsilon$, kde S je počáteční neterminál, je CNF povoleno).

Příklad: odstranění ε -pravidel

Příklad (část 1/2)

Mějme CFG $\mathcal{G} = (\{S, A, B\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid AbAbA$$

$$A \rightarrow a \mid AA \mid B$$

$$B \rightarrow \varepsilon$$

1 Odstraníme pravidlo $B \rightarrow \varepsilon$ a přidáme nové pravidlo $A \rightarrow \varepsilon$

- už neexistuje žádné pravidlo s B na levé straně, proto B bude odstraněno v jednom z pozdějších kroků převodu do CNF.

$$S \rightarrow \varepsilon \mid AbAbA$$

$$A \rightarrow a \mid AA \mid \cancel{B} \mid \varepsilon$$

$$B \rightarrow \cancel{\varepsilon}$$

Příklad (část 2/2)

2 Odstraníme pravidlo $A \rightarrow \varepsilon$ a přidáme nová pravidla:

$$S \rightarrow \varepsilon \mid AbAbA$$

$$A \rightarrow a \mid AA \mid \not\in$$

- $A \rightarrow A$ kvůli pravidlu $A \rightarrow AA$ (nahradíme jeden výskyt A za ε)
- $S \rightarrow bAbA$ kvůli pravidlu $S \rightarrow AbAbA$ (nahradíme první výskyt A za ε)
- $S \rightarrow AbbA$ kvůli pravidlu $S \rightarrow AbAbA$ (nahradíme druhý výskyt A za ε)
- a tak dále pro všechny další kombinace: $S \rightarrow AbAb \mid Abb \mid bAb \mid bbA \mid bb$

Dostaneme výslednou gramatiku $\mathcal{G}' = (\{S, A, B\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid AbAbA \mid bAbA \mid AbbA \mid AbAb \mid Abb \mid bAb \mid bbA \mid bb$$

$$A \rightarrow a \mid AA \mid A$$

Odstranění jednoduchých pravidel

Intuice: pravidla ve tvaru $A \rightarrow B$ (tzv. *jednoduchá pravidla*), kde $A, B \in N$, nejsou v CNF povolena (můžeme vygenerovat vždy jen dva neterminály).

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N, \Sigma, P', S)$ bez jednoduchých pravidel

- 1 Inicializace: položíme $\mathcal{G}' = \mathcal{G}$.
- 2 Odstraníme libovolné pravidlo $A \rightarrow B$ z $\mathcal{G}'$.
- 3 Přidáme do $\mathcal{G}'$ nová pravidla $A \rightarrow \alpha_1 \mid \dots \mid \alpha_k$ takové, že platí:
 - $B \rightarrow \alpha_1 \mid \dots \mid \alpha_k$ jsou všechna pravidla s B na levé straně,
 - přidáme jen pravidla, které jsme už tímto procesem neodstranili (jinak dojde k zacyklení).
- 4 Kroky 2 a 3 opakujeme, dokud neodstraníme všechna jednoduchá pravidla.

Příklad: odstranění jednoduchých pravidel

Příklad (část 1/2)

Mějme CFG $\mathcal{G} = (\{S, A, B\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid bb \mid B$$

$$A \rightarrow a \mid AA \mid B$$

$$B \rightarrow A$$

1 Odstraníme pravidlo $B \rightarrow A$ a přidáme do B všechny pravidlo z A :

- $B \rightarrow a$ (kvůli pravidlu $A \rightarrow a$)
- $B \rightarrow AA$ (kvůli pravidlu $A \rightarrow AA$)
- $B \rightarrow B$ (kvůli pravidlu $A \rightarrow B$)

$$S \rightarrow \varepsilon \mid bb \mid B$$

$$A \rightarrow a \mid AA \mid B$$

$$B \rightarrow \cancel{A} \mid a \mid AA \mid B$$

Příklad (část 2/2)

- 2 Odstraníme pravidlo $B \rightarrow B$ a nic nepřidáme (všechny pravidla z B už v B jsou).
- 3 Odstraníme pravidlo $A \rightarrow B$ a opět nic nepřidáme (všechny pravidla z B už v A jsou).

$$S \rightarrow \varepsilon \mid bb \mid B$$

$$A \rightarrow a \mid AA \mid \cancel{B}$$

$$B \rightarrow a \mid AA \mid \cancel{B}$$

- 4 Odstraníme pravidlo $S \rightarrow B$ a přidáme do S všechny pravidlo z B :
 - přidáme $S \rightarrow a$ (kvůli pravidlu $B \rightarrow a$) a $S \rightarrow AA$ (kvůli pravidlu $B \rightarrow AA$)

$$S \rightarrow \varepsilon \mid bb \mid \cancel{B} \mid a \mid AA$$

$$A \rightarrow a \mid AA$$

$$B \rightarrow a \mid AA$$

Takto upravená gramatika $\mathcal{G}'$ je bez jednoduchých pravidel.

Odstranění neterminálů, které nic nederivují

Intuice: iterativně počítáme množinu neterminálů N_i , které derivují řetězec skládající se pouze z terminálů (umí ukončit svůj výpočet).

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N', \Sigma, P', S)$, kde každý neterminál je schopný ukončit své derivování

- 1 Inicializace: položíme $N_0 = \emptyset$ a $\mathcal{G}' = \mathcal{G}$.
- 2 Průběžný krok: postupně konstruujeme další $N_i \subseteq N$, dokud neplatí $N_i = N_{i-1}$.
 - $N_i = N_{i-1} \cup \{A \mid A \rightarrow \alpha \in P \wedge \alpha \in (N_{i-1} \cup \Sigma)^*\}$
- 3 Nechť N_e je poslední vypočítaná množina N_i :
 - v $\mathcal{G}'$ odstraníme všechny neterminály, které nejsou v N_e a pravidla, v nichž se tyto symboly vyskytují, formálně $N' = N \cap N_e$ a $P' = P \cap (N_e \times (N_e \cup \Sigma)^*)$.

Poznámka: může se stát, že po tomto kroku je $N' = \emptyset$, potom ale platí $L(\mathcal{G}) = \emptyset$.

Příklad: odstranění neterminálů, které nic nederivují

Příklad

Mějme CFG $\mathcal{G} = (\{S, A, B, C\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow bbC \mid aBB$$

$$A \rightarrow a \mid AA \mid B$$

$$B \rightarrow A \mid aC$$

$$C \rightarrow bC$$

- $N_0 = \emptyset$
- $N_1 = \{A\}$ (kvůli pravidlu $A \rightarrow a$)
- $N_2 = \{A, B\}$ (kvůli $A \in N_1$ a pravidlu $B \rightarrow A$)
- $N_3 = \{S, A, B\}$ (kvůli $B \in N_2$ a pravidlu $S \rightarrow aBB$)
- $N_3 = N_4 = \{S, A, B\}$ (odstraníme C a všechny jeho pravidla)

$\forall \mathcal{G}' = (\{S, A, B\}, \{a, b\}, P', S)$ umí každý neterminál ukončit své derivování:

$$S \rightarrow aBB$$

$$A \rightarrow a \mid AA \mid B$$

$$B \rightarrow A$$

Odstranění nedosažitelných symbolů

Intuice: iterativně počítáme množinu symbolů $V_i \subseteq \Sigma \cup N$, které lze vygenerovat z počátečního neterminálu.

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N', \Sigma', P', S)$ bez nedosažitelných symbolů

- 1 Inicializace: položíme $V_0 = \{S\}$.
- 2 Průběžný krok: postupně konstruujeme další $V_i \subseteq \Sigma \cup N$ dokud neplatí $V_i = V_{i-1}$.
 - $V_i = V_{i-1} \cup \{X \mid A \in V_{i-1} \wedge A \rightarrow \alpha X \beta \in P, \text{ pro nějaké } A \in N \text{ a } \alpha, \beta \in (N \cup \Sigma)^*\}$
- 3 Nechť V_e je poslední vypočítaná množina V_i :
 - v $\mathcal{G}'$ odstraníme všechny terminály i neterminály, které nejsou v V_e a pravidla, v nichž se tyto symboly vyskytují, formálně $N' = N \cap V_e$, $\Sigma' = \Sigma \cap V_e$ a $P' = P \cap (V_e \times V_e^*)$.

Příklad: odstranění nedosažitelných symbolů

Příklad

Mějme CFG $\mathcal{G} = (\{S, A, B, C, D\}, \{a, b, c\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid bb \mid BB$$

$$A \rightarrow a \mid AA \mid BBB$$

$$B \rightarrow AA \mid a$$

$$C \rightarrow bC \mid c$$

$$D \rightarrow SB \mid ab$$

- $V_0 = \{S\}$
- $V_1 = \{S, b, B\}$ (kvůli $S \in V_0$ a pravidlům $S \rightarrow bb \mid BB$)
- $V_2 = \{S, a, b, A, B\}$ (kvůli $B \in V_1$ a pravidlům $B \rightarrow a \mid AA$)
- $V_3 = V_4 = \{S, a, b, A, B\}$ (odstraníme c, C a D a všechny jejich pravidla)

Vypočítali jsme $\mathcal{G}' = (\{S, A, B\}, \{a, b\}, P', S)$ bez nedosažitelných symbolů:

$$S \rightarrow \varepsilon \mid bb \mid BB$$

$$A \rightarrow a \mid AA \mid BBB$$

$$B \rightarrow AA \mid a$$

Odstranění pravidel v nesprávném tvaru

Intuice: pořád nám ještě zbývají pravidla, které generují kombinace terminálů a neterminálů, nebo více než dva symboly.

Vstup: CFG $\mathcal{G} = (N, \Sigma, P, S)$

Výstup: CFG $\mathcal{G}' = (N', \Sigma', P', S)$

- 1 Inicializace: položíme $\mathcal{G}' = \mathcal{G}$.
- 2 Pro každý terminál $a \in \Sigma$ vytvoříme nový neterminál $\langle a \rangle$ a pravidlo $\langle a \rangle \rightarrow a$, které přidáme do $\mathcal{G}'$ jen v případě, že přidáme jiné pravidlo generující $\langle a \rangle$.
- 3 každé pravidlo ve tvaru $A \rightarrow x_i x_j$, kde aspoň jeden z x_i, x_j je terminál, nahraď novým pravidlem $A \rightarrow \hat{x}_i \hat{x}_j$, kde $\hat{x}_i$ je $\langle a \rangle$, pokud $x_i = a$, jinak x_i (a analogicky pro $\hat{x}_j$).
- 4 Každé pravidlo ve tvaru $A \rightarrow x_1 \dots x_k$, kde $k \geq 3$ a $x_1, \dots, x_k \in (N \cup \Sigma)$ jsou terminály nebo neterminály, nahradíme novými pravidly:

$$A \rightarrow \hat{x}_1 \langle x_2 \dots x_k \rangle, \quad \langle x_2 \dots x_k \rangle \rightarrow \hat{x}_2 \langle x_3 \dots x_k \rangle, \quad \dots, \quad \langle x_{k-1} x_k \rangle \rightarrow \hat{x}_{k-1} \hat{x}_k$$

kde $\langle x_2 \dots x_k \rangle, \dots, \langle x_{k-1} x_k \rangle$ jsou nové neterminály a $\hat{x}_i$ je $\langle a \rangle$, pokud $x_i = a$ (tedy x_i je nějaký terminál), jinak $\hat{x}_i$ je neterminál x_i .

Příklad: Odstranění pravidel v nesprávném tvaru

Příklad

Mějme CFG $\mathcal{G} = (\{S, A, B\}, \{a, b\}, P, S)$ s pravidly:

$$S \rightarrow \varepsilon \mid bB \mid BABA$$

$$A \rightarrow a \mid AA \mid BB$$

$$B \rightarrow AA \mid aa$$

- 1 vytvoříme nové neterminály $\langle a \rangle$ a $\langle b \rangle$ a pravidla $\langle a \rangle \rightarrow a$ a $\langle b \rangle \rightarrow b$
- 2 upravíme pravidla $S \rightarrow bB$ na $S \rightarrow \langle b \rangle B$, a dále $B \rightarrow aa$ na $B \rightarrow \langle a \rangle \langle a \rangle$
- 3 upravíme pravidlo $S \rightarrow BABA$ na pravidla $S \rightarrow B\langle ABA \rangle$, $\langle ABA \rangle \rightarrow A\langle BA \rangle$ a $\langle BA \rangle \rightarrow BA$, kde $\langle ABA \rangle$ a $\langle BA \rangle$ jsou nové neterminály

$$S \rightarrow \varepsilon \mid \langle b \rangle B \mid B\langle ABA \rangle$$

$$\langle a \rangle \rightarrow a$$

$$A \rightarrow a \mid AA \mid BB$$

$$\langle b \rangle \rightarrow b$$

$$B \rightarrow AA \mid \langle a \rangle \langle a \rangle$$

$$\langle ABA \rangle \rightarrow A\langle BA \rangle$$

$$\langle BA \rangle \rightarrow BA$$

Algorithmus CYK

Rozhodnutelnost náležení do jazyka CFG

Algoritmus CYK

- název pochází z jmen autorů Cocke, Younger a Kasami, publikován roku 1961
- rozhoduje zda řetězec $w \in \Sigma^*$ lze vygenerovat bezkontextovou gramatikou $\mathcal{G}$ v CNF
- využívá dynamického programování pro výpočet množin neterminálů, které generují jednotlivé podřetězce w
- časová složitost $O(n^3 \cdot |\mathcal{G}|)$, kde $n = |w|$

Věta

Mějme CFG $\mathcal{G} = (N, \Sigma, P, S)$, pak ověřit zda $w \in L(\mathcal{G})$ je rozhodnutelný problém.

Důkaz

- 1 Nejprve $\mathcal{G}$ převedeme do CNF,
- 2 poté použijeme algoritmus CYK se vstupem $\mathcal{G}$ a w .



Algoritmus CYK

Intuice: vypočítáme tabulku, kde každá buňka $X_{i,j}$ obsahuje množinu neterminálů, které generují podřetězec $w_i \dots w_j$ řetězce w (a v buňce $X_{1,n}$ tedy generují celý řetězec w).

Vstup: řetězec $w = w_1 \dots w_n \in \Sigma^*$ a CFG $\mathcal{G} = (N, \Sigma, P, S)$ v CNF

Výstup: true pokud $S \Rightarrow_{\mathcal{G}}^* w$, jinak false

1 Inicializace: vytvoř trojúhelníkovou tabulku pro slovo w délky n :

- vytvoř první řádek buněk $X_{1,1}$ až $X_{n,n}$
- každý sloupec i pak obsahuje buňky od $X_{i,i}$ (spodní řádek) až po $X_{i,n}$ (vrchní řádek)
- pod spodní řádek tabulky napiš slovo w , tak aby w_1 bylo pod buňkou $X_{1,1}$, dále w_2 pod buňkou $X_{2,2}$, $\dots$, a w_n bylo pod buňkou $X_{n,n}$

2 Základní krok: výpočet prvního řádku tabulky.

- do každé buňky $X_{i,i}$ přidáme neterminál A , pokud v $\mathcal{G}$ existuje pravidlo $A \rightarrow w_i$

3 Průběžný krok: počítáme řadky tabulky od spodu, postupujeme zleva do prava.

- do $X_{i,j}$ vložíme neterminál A pokud existuje index k a pravidlo $A \rightarrow BC$ takové, že:
($i \leq k < j$) $\wedge B \in X_{i,k} \wedge C \in X_{k+1,j}$ (potom A generuje podřetězec $w_i \dots w_j$)

4 Vrať true pokud $S \in X_{1,n}$ (tedy S generuje celý řetězec w), jinak vrať false.

Korektnost algoritmu CYK

Věta

Algoritmus CYK je korektní.

Idea důkazu

Základem algoritmu je tvar tabulky a gramatiky $\mathcal{G} = (N, \Sigma, P, S)$, která musí být v CNF:

- 1 každý neterminál se přepisuje přesně na dva neterminály, nebo jeden terminál,
- 2 řádek tabulky začínající buňkou $X_{1,j}$ odpovídá podřetězcům w délky j .

Indukcí ukážeme, že v buňce $X_{i,j}$ jsou pouze neterminály generující podřetězec $w_i \dots w_j$:

- Základní krok: uvažujme podřetězce délky 1, tedy případ $i = j$. Neterminály generující w_i jsou v buňce $X_{i,i}$, jejíž obsah získáme triviálně z pravidel $A \rightarrow w_i \in P$.
- Průběžný krok: předpokládáme platnost tvrzení pro podřetězce délky m a dále jej dokážeme pro podřetězce délky $m + 1$:
 - buňka $X_{i,j}$ je na řádku $m + 1$, a dle algoritmu jsou všechny řádky pod ním již vypočítané,
 - neterminál A přidáme do $X_{i,j}$ pouze pokud existuje pravidlo $A \rightarrow BC$ a platí $B \in X_{i,k}$ a $C \in X_{k+1,j}$ (kde buňky $X_{i,k}$ a $X_{k+1,j}$ generují kratší řetězec, a proto jsou již vypočítané),
 - potom A generuje $w_i \dots w_j$, protože B generuje $w_i \dots w_k$ a C generuje $w_{k+1} \dots w_j$. □

Příklad: CYK

Příklad (část 1/7)

Pomocí algoritmu CYK rozhodněte zda $S \Rightarrow_{\mathcal{G}}^* baaba$ v gramatice $\mathcal{G} = (N, \Sigma, P, S)$.

1 inicializace trojúhelníkové tabulky pro slovo délky $|baaba| = 5$:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

1,5				
1,4	2,5			
1,3	2,4	3,5		
1,2	2,3	3,4	4,5	
1,1	2,2	3,3	4,4	5,5
b	a	a	b	a

Příklad (část 2/7)

2 základní krok, výpočet prvního řádku tabulky:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

Zde například:

- $X_{1,1}$ a $X_{4,4}$ obsahují B , protože v $\mathcal{G}$ je pravidlo $B \rightarrow b$
- $X_{2,2}$, $X_{3,3}$ a $X_{5,5}$ obsahují $\{A, C\}$, protože existují pravidla $A \rightarrow a$ a $C \rightarrow a$

1,5				
1,4	2,5			
1,3	2,4	3,5		
1,2	2,3	3,4	4,5	
1,1	2,2	3,3	4,4	5,5
$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
b	a	a	b	a

Příklad (část 3/7)

3 průběžný krok, výpočet druhého řádku tabulky:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

Zde například:

- $X_{1,2}$ obsahuje $\{S, A\}$, protože:
 - $A \rightarrow BA \wedge B \in X_{1,1} \wedge A \in X_{2,2}$
 - $S \rightarrow BC \wedge B \in X_{1,1} \wedge C \in X_{2,2}$
- $X_{2,3}$ obsahuje $\{B\}$, protože:
 - $B \rightarrow CC \wedge C \in X_{2,2} \wedge C \in X_{3,3}$

1,5				
1,4	2,5			
1,3	2,4	3,5		
1,2	2,3	3,4	4,5	
$\{S, A\}$	$\{B\}$	$\{S, C\}$	$\{S, A\}$	
1,1	2,2	3,3	4,4	5,5
$\{B\}$	$\{A, C\}$	$\{A, C\}$	$\{B\}$	$\{A, C\}$
b	a	a	b	a

Příklad (část 4/7)

3 průběžný krok, výpočet třetího řádku tabulky:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

Zde například:

- $X_{1,3} = \emptyset$, protože žádný neterminál neumí vygenerovat $w_1 w_2 w_3 = baa$
- $X_{2,4}$ obsahuje $\{B\}$, protože:
 - $B \rightarrow CC \wedge C \in X_{2,2} \wedge C \in X_{3,4}$
- $X_{3,5}$ obsahuje $\{B\}$, protože:
 - $B \rightarrow CC \wedge C \in X_{3,4} \wedge C \in X_{5,5}$

1,5				
1,4	2,5			
1,3 $\emptyset$	2,4 $\{B\}$	3,5 $\{B\}$		
1,2 $\{S, A\}$	2,3 $\{B\}$	3,4 $\{S, C\}$	4,5 $\{S, A\}$	
1,1 $\{B\}$	2,2 $\{A, C\}$	3,3 $\{A, C\}$	4,4 $\{B\}$	5,5 $\{A, C\}$
b	a	a	b	a

Příklad (část 5/7)

3 průběžný krok, výpočet čtvrtého řádku tabulky:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

Zde například:

- $X_{1,4} = \emptyset$, protože žádný neterminál neumí vygenerovat $w_1w_2w_3w_4 = baab$
- $X_{2,5}$ obsahuje $\{S, A, C\}$, protože:
 - $S \rightarrow AB \wedge A \in X_{2,2} \wedge B \in X_{3,5}$
 - $A \rightarrow BA \wedge A \in X_{2,4} \wedge B \in X_{5,5}$
 - $C \rightarrow AB \wedge A \in X_{2,2} \wedge B \in X_{3,5}$

1,5				
1,4 $\emptyset$	2,5 $\{S, A, C\}$			
1,3 $\emptyset$	2,4 $\{B\}$	3,5 $\{B\}$		
1,2 $\{S, A\}$	2,3 $\{B\}$	3,4 $\{S, C\}$	4,5 $\{S, A\}$	
1,1 $\{B\}$	2,2 $\{A, C\}$	3,3 $\{A, C\}$	4,4 $\{B\}$	5,5 $\{A, C\}$
b	a	a	b	a

Příklad (část 6/7)

3 průběžný krok, výpočet pátého řádku tabulky:

$$S \rightarrow AB \mid BC$$

$$A \rightarrow BA \mid a$$

$$B \rightarrow CC \mid b$$

$$C \rightarrow AB \mid a$$

Zde například:

■ $X_{1,5}$ obsahuje $\{S, A, C\}$, protože:

- $S \rightarrow AB \wedge A \in X_{1,2} \wedge B \in X_{3,5}$
- $A \rightarrow BA \wedge A \in X_{1,1} \wedge B \in X_{2,5}$
- $C \rightarrow AB \wedge A \in X_{1,2} \wedge B \in X_{3,5}$

1,5 $\{S, A, C\}$				
1,4 $\emptyset$	2,5 $\{S, A, C\}$			
1,3 $\emptyset$	2,4 $\{B\}$	3,5 $\{B\}$		
1,2 $\{S, A\}$	2,3 $\{B\}$	3,4 $\{S, C\}$	4,5 $\{S, A\}$	
1,1 $\{B\}$	2,2 $\{A, C\}$	3,3 $\{A, C\}$	4,4 $\{B\}$	5,5 $\{A, C\}$
b	a	a	b	a

4 vrať true, protože $S \in X_{1,5}$, a tedy platí $S \Rightarrow_G^* baaba$.

Příklad (část 7/7)

Derivační strom lze vypočítat opačným průchodem tabulky z S v buňce $X_{1,5}$:

$$S \rightarrow BC \wedge B \in X_{1,1} \wedge C \in X_{2,5}$$

$$B \rightarrow b \wedge B \in X_{1,1}$$

$$C \rightarrow AB \wedge A \in X_{2,2} \wedge B \in X_{3,5}$$

$$A \rightarrow a \wedge A \in X_{2,2}$$

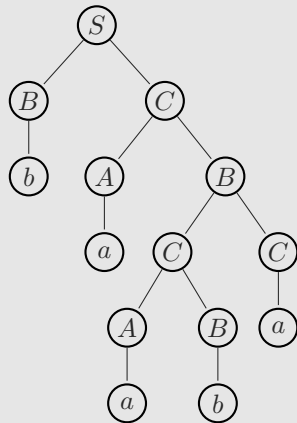
$$B \rightarrow CC \wedge C \in X_{3,4} \wedge C \in X_{5,5}$$

$$C \rightarrow AB \wedge A \in X_{3,3} \wedge B \in X_{4,4}$$

$$A \rightarrow a \wedge A \in X_{3,3}$$

$$B \rightarrow b \wedge C \in X_{4,4}$$

$$C \rightarrow a \wedge C \in X_{5,5}$$



Derivace odpovídající tomuto stromu:

$S \Rightarrow BC \Rightarrow bC \Rightarrow bAB \Rightarrow baB \Rightarrow baCC \Rightarrow baABC \Rightarrow baaBC \Rightarrow baabC \Rightarrow baaba$