Automaty a gramatiky (BI-AAG)

6. Bezkontextové gramatiky

Jan Holub

Katedra teoretické informatiky

Fakulta informačních technologií

ČVUT v Praze

Bezkontextové gramatiky

- Bezkontextové gramatiky
  - Dokáží popsat převážnou většinu syntaktických struktur programovacích jazyků.
  - Jsou známy algoritmy pro efektivní analýzu věty bezkontextových jazyků.
- Syntaktická analýza:
  - Je daný řetězec w generován danou gramatikou G?
  - Jaká je struktura tohoto řetězce?

Jednoznačné a nejednoznačné gramatiky

Definice

Bezkontextová gramatika $G = (N, T, P, S)$ je *nejednoznačná (víceznačná)*, jestliže existuje věta $w \in L(G)$ taková, že je výsledkem nejméně dvou různých derivačních stromů. V opačném případě je gramatika *jednoznačná*.

Jednoznačné a nejednoznačné gramatiky

Příklad

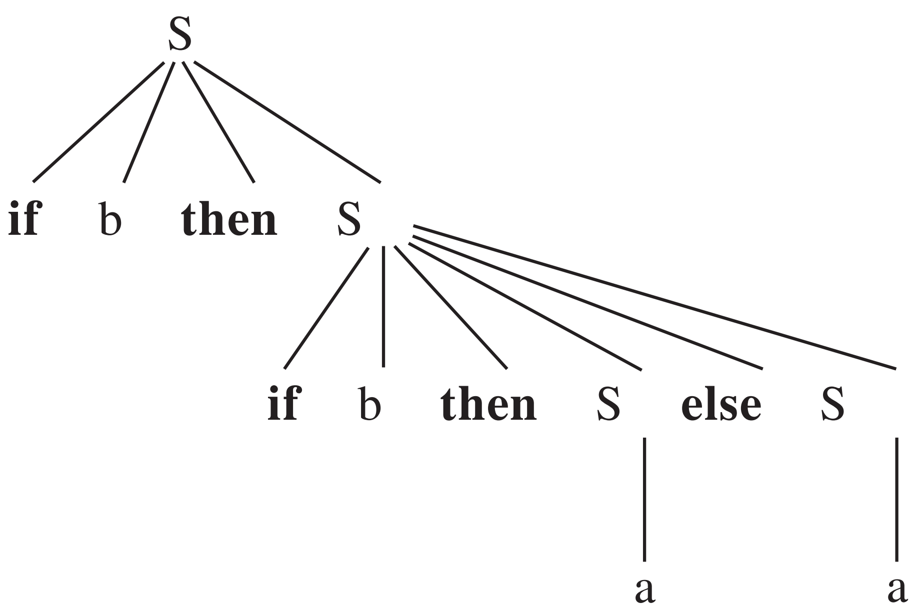
$G = (\{S\}, \{$*a,b,***if***,***then***,***else**$\}$*,P,S)*, kde P:

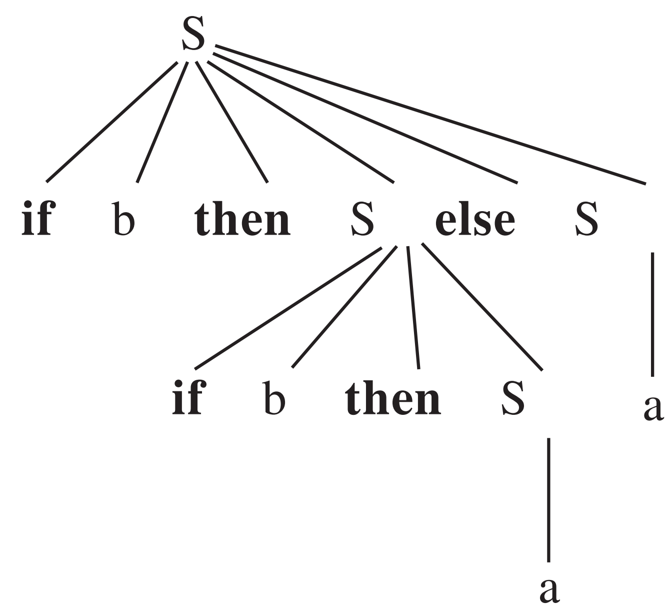
$S \rightarrow$ **if** b **then** S **else** S

$S \rightarrow$ **if** b **then** S

$S \rightarrow a$

Gramatika je nejednoznačná: **if** b **then if** b **then** a **else** a





Jednoznačné a nejednoznačné gramatiky

Příklad
BG obsahuje pravidlo $A \rightarrow AA$ a proto je nejednoznačná. Pro větnou formu AAA existují dva derivační stromy.

Lze odstranit nahrazením pravidla $A \rightarrow AA$ pravidly
$A \rightarrow AB$
$A \rightarrow B$. $\square$

Jednoznačné a nejednoznačné gramatiky

- V mnoha případech lze nejednoznačnost z gramatiky odstranit.
- Inherentně (podstatně) nejednoznačné jazyky:
  Nelze je generovat jednoznačnou gramatikou.
- Pro bezkontextové gramatiky nelze sestrojit algoritmus, který by rozhodl, zda daná gramatika je nejednoznačná, a vhodnou modifikací této gramatiky získat gramatiku jednoznačnou.

Příklad

$G = (\{S_1, S_2\}, \{a, b, \mathbf{if}, \mathbf{then}, \mathbf{else}\}, P, S_1)$, kde P:

$S_1 \to \mathbf{if}\ b\ \mathbf{then}\ S_1 \mid \mathbf{if}\ b\ \mathbf{then}\ S_2\ \mathbf{else}\ S_1 \mid a$

$S_2 \to \mathbf{if}\ b\ \mathbf{then}\ S_2\ \mathbf{else}\ S_2 \mid a$

Symbol $\mathbf{else}$ se vztahuje vždy k nejbližšímu předchozímu $\mathbf{then}$.

Transformace bezkontextových gramatik

Věta
Existuje algoritmus, který určí, zda jazyk generovaný danou bezkontextovou gramatikou je prázdný. □

Algoritmus Je $L(G)$ neprázdný?
Vstup: Bezkontektová gramatika $G = (N, T, P, S)$.
Výstup: Ano, jestliže $L(G) \neq \emptyset$, ne jinak.
Metoda: Konstruujeme množiny $N_0, N_1, \ldots$ takto:

1. $N_0 = \emptyset$, $i := 1$
2. $N_i = \{A : A \rightarrow \alpha \in P, \alpha \in (N_{i-1} \cup T)^*\} \cup N_{i-1}$
3. Jestliže $N_i \neq N_{i-1}$, pak $i := i + 1$ a proveď (2), jinak $N_t = N_i$.
4. Jestliže $S \in N_t$, pak ano, jinak ne. □

Transformace bezkontextových gramatik

Příklad

$G = (\{S, A, B\}, \{a, b\}, \{S \to a, S \to A, A \to AB, B \to b\}, S).$

$N_0 = \emptyset$
$N_1 = \{S, B\}$
$N_2 = \{S, B\}$
$N_1 = N_2 = N_t$
$S \in N_t \Rightarrow$ gramatika G generuje neprázdný jazyk.

Transformace bezkontextových gramatik

Definice

Symbol $X \in N \cup T$ je *nedostupný* v bezkontextové gramatice $G = (N, T, P, S)$, jestliže X se neobjeví v žádné větné formě, tj. neexistuje derivace tvaru $S \Rightarrow^{+} \alpha X \beta,\ \alpha, \beta \in (N \cup T)^*$.

Transformace bezkontextových gramatik

Algoritmus Vyloučení nedostupných symbolů.
Vstup: Bezkontextová gramatika $G = (N, T, P, S)$.
Výstup: $G' = (N', T', P', S)$ taková, že
a) $L(G') = L(G)$,
b) pro všechna $X \in N' \cup T'$ existují $\alpha, \beta \in (N' \cup T')^*$ taková, že $S \Rightarrow^* \alpha X \beta$.
Metoda:

1. $V_0 = \{S\}$ a $i := 1$.
2. $V_i = \{X : A \rightarrow \alpha X \beta \in P, A \in V_{i-1}\} \cup V_{i-1}$.
3. Jestliže $V_i \neq V_{i-1}$ pak $i := i + 1$ a proveď krok 2, jinak

   $N' = V_i \cap N$,
   $T' = V_i \cap T$,
   $P' = \{A \rightarrow \alpha : A \in N', \alpha \in V_i^*, A \rightarrow \alpha \in P\}$,
   $G' = (N', T', P', S)$

Transformace bezkontextových gramatik

Příklad

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

$V_0 = \{S\}$
$V_1 = \{S\} \cup \{a, A\}$
$V_2 = \{S, A, a\} \cup \{B\}$
$V_3 = \{S, A, B, a, \} \cup \{b\}$
$V_4 = \{S, A, B, a, b\}$

Transformace bezkontextových gramatik

Definice

Symbol $X \in N \cup T$ je *zbytečný* $G = (N, T, P, S)$, jestliže neexistuje $S \Rightarrow^* wXy \Rightarrow^* wxy$, kde $w, x, y \in T^*$.

Transformace bezkontextových gramatik

Algoritmus Vyloučení zbytečných symbolů.

Vstup: BG $G = (N, T, P, S)$, $L(G) \neq \emptyset$.

Výstup: BG $G' = (N', T', P', S)$,

a) $L(G') = L(G)$ a

b) $\forall X \in N' \cup T'\ S \Rightarrow^* \alpha X \beta$.

Metoda:

1. Pomocí algoritmu 3.6 (Je $L(G)$ prázdný?) získáme N_t.
   $G_1 = (N_t, T, P_1, S)$,
   $P_1 = \{A \to \alpha : A \in N_t, \alpha \in (N_t \cup T)^*, A \to \alpha \in P\}$.
2. Algoritmus 3.9 (Vyloučení nedostupných symbolů) vyloučí všechny neterminální symboly z G takové, ze kterých není možno generovat terminální řetězce a poté všechny nedostupné symboly. Získáme $G' = (N', T', P', S)$.

Transformace bezkontextových gramatik

Příklad

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

Krok 1.:

$N_t = \{S, B\}$

$G_1 = (\{S, B\}, \{a, b\}, \{S \to a, B \to b\}, S)$

Krok 2.:

$V_0 = \{S\}$

$V_1 = \{S, a\}$

$V_2 = \{S, a\}$

$G' = (\{S\}, \{a\}, \{S \to a\}, S)$

Transformace bezkontextových gramatik

Věta

(Věta o vyloučení pravidla, věta o dosazování).
$G = (N, T, P, S)$ je bezkontextová gramatika a
$A \to \alpha B \beta \in P, B \in N, A \neq B$ a $\alpha, \beta \in (N \cup T)^*$.
Nechť $B \to \gamma_1 \mid \gamma_2 \mid \ldots \mid \gamma_k$ jsou všechna pravidla se symbolem B na levé straně v P. Nechť $G' = (N, T, P', S)$, kde
$P' = P \cup \{A \to \alpha\gamma_1\beta \mid \alpha\gamma_2\beta \mid \ldots \mid \alpha\gamma_k\beta\} \setminus \{A \to \alpha B \beta\}$. Pak
$L(G) = L(G')$.

Transformace bezkontextových gramatik

Definice

Bezkontextová gramatika $G = (N, T, P, S)$ je *bez cyklů*, jestliže v ní není možná derivace: $A \Rightarrow^+ A$ pro nějaké $A \in N$. □

Definice

Bezkontextová gramatika $G = (N, T, P, S)$ je bez ε-pravidel, když

1. P neobsahuje žádná ε-pravidla, nebo
2. P obsahuje jediné ε-pravidlo tvaru $S \to \varepsilon$ a S se nevyskytuje na pravé straně žádného pravidla v P. □

Definice

Bezkontextová gramatika $G = (N, T, P, S)$ je *vlastní*, jestliže je bez cyklů, bez ε-pravidel a nemá zbytečné symboly. □

Transformace bezkontextových gramatik

Věta

Jestliže je bezkontextová gramatika $G = (N, T, P, S)$ bez ε-pravidel a bez jednoduchých pravidel, pak je bez cyklů. $\square$

Věta

Jestliže L je bezkontextový jazyk, potom může být generován nějakou vlastní gramatikou G. $\square$

Definice

Bezkontextová gramatika $G = (N, T, P, S)$ je *redukovaná*, jestliže neobsahuje zbytečné symboly. $\square$

Normální tvar podle Chomského

Definice

BG $G = (N, T, P, S)$ je v normálním tvaru podle Chomského, jestliže každé pravidlo v P má jeden z následujících tvarů:

1. $A \rightarrow BC$ pro $A, B, C \in N$.
2. $A \rightarrow a$ pro $a \in T, A \in N$.
3. Jestliže $\varepsilon \in L(G)$, pak $S \rightarrow \varepsilon$ je pravidlo v P a S se nevyskytuje na pravé straně žádného pravidla. $\square$

Normální tvar podle Chomského

Algoritmus Převod gram. do Chomského normálního tvaru
Vstup: Vlastní bezkontextová gramatika $G = (N, T, P, S)$ bez jednoduchých pravidel.
Výstup: BG $G' = (N', T, P', S)$ v Chomského normálním tvaru, $L(G) = L(G')$
Metoda:

1. $P' = \emptyset$.
2. Přidej všechna pravidla tvaru $A \rightarrow a$ z P do P'.
3. Přidej všechna pravidla tvaru $A \rightarrow BC$ z P do P'.
4. Jestliže $S \rightarrow \varepsilon$ je v P, přidej $S \rightarrow \varepsilon$ do P'.

Normální tvar podle Chomského

Algoritmus (pokračování):

5. Pro každé pravidlo tvaru $A \rightarrow X_1 X_2 \dots X_k \in P$, kde $k > 2$, přidej do P' následující pravidla (když $X_i \in N$, $X_i' = X_i$, když $X_i \in T$, X_i' je nový neterminální symbol):

$$\begin{array}{rcl} A & \rightarrow & X_1' \langle X_2 \dots X_k \rangle \\ \langle X_2 \dots X_k \rangle & \rightarrow & X_2' \langle X_3 \dots X_k \rangle \\ & \vdots & \\ \langle X_{k-2} \dots X_k \rangle & \rightarrow & X_{k-2}' \langle X_{k-1} X_k \rangle \\ \langle X_{k-1} X_k \rangle & \rightarrow & X_{k-1}' X_k', \end{array}$$

   kde každý symbol $\langle X_i \dots X_k \rangle$ je nový neterminální symbol v P.

Normální tvar podle Chomského

Algoritmus (pokračování):

6. Pro každé pravidlo tvaru $A \to X_1X_2$, kde X_1 nebo X_2 nebo X_1 i X_2 jsou v T, přidej do P' pravidla $A \to X_1'X_2'$.
7. Pro každý neterminální symbol a' vzniklý v krocích 5. a 6. přidej do P' pravidlo $a' \to a$. Konečně N' je sjednocení N se všemi nově vzniklými neterminálními symboly. Gramatika $G' = (N', T, P', S)$.

Normální tvar podle Chomského

Věta

Nechť L je bezkontextový jazyk. Pak L je jazyk generovaný nějakou gramatikou v Chomského normálním tvaru.

Normální tvar podle Chomského

Příklad

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

$S \to aAB \mid BA$
$A \to BBB \mid a$
$B \to AS \mid b$

krok 1: $A \to a$

$B \to b$ přidáme do P'.

krok 2: $S \to BA$

$B \to A\,S$ přidáme do P'.

krok 3: Neprovedeme nic.

krok 4: $S \to aAB \in P \Rightarrow S \to a'\langle AB\rangle$

$\langle AB\rangle \to AB$ přidáme do P'.

$A \to BBB \in P \Rightarrow A \to B\langle BB\rangle$

$\langle BB\rangle \to BB$ přidáme do P'.

Normální tvar podle Chomského

Příklad (pokračování)

krok 5: Neprovedeme nic.

krok 6: $a' \to a$ přidáme do P'.

P': $S \to a'\langle AB\rangle \mid BA$

$A \to B\langle BB\rangle \mid a$

$B \to AS \mid b$

$\langle AB\rangle \to AB$

$\langle BB\rangle \to BB$

$a' \to a$

$N' = N \cup \{\langle AB\rangle, \langle BB\rangle, a'\}$

Normální tvar podle Chomského

Algoritmus Cocke-Younger-Kasami (CYK) algorithm
Vstup: Bezkontextová gramatika $G = (N, T, P, S)$
v Chomského normálním tvaru, $X \in T^*$
Výstup: Odpověď, zda $X \in L(G)$
Metoda: $X = \{x_1, x_2, \dots, x_n\}$

1. Inicializuj pole $P[n, n]$ na $\emptyset$.
2. Pro každé $i = 1..n$ a pro každé pravidlo $A \to t_i$ nastav $P[i, 1] := P[i, 1] \cup \{A\}$
3. Pro každé $i = 2..n$ udělej (délka řetězce)
   Pro každé $j = 1..n - i + 1$ udělej (pozice v textu)
    Pro každé $k = 1..i - 1$ udělej (délka gen. řetězce)
     Jestliže $B \in P[j, k]$, $C \in P[j + k, i - k]$ a $A \to BC$
     pak $P[j, i] := P[j, i] \cup \{A\}$
4. Jestliže $S \in P[1, n]$, pak $X \in L(G)$,
   jinak, $X \notin L(G)$.

Normální tvar podle Chomského

Příklad

Text $aaba$ a BG $G = (\{S, A, B, C\}, \{a, b\}, P, S)$, P:

$S \rightarrow AB \mid BC$
$A \rightarrow BA \mid a$
$B \rightarrow CC \mid b$
$C \rightarrow AB \mid a$

4				
3				
2				
1				
	a	a	b	a

Normální tvar podle Chomského

Věta

Je-li $G = (N, T, P, S)$ bezkontextová gramatika a $X \in T^*$ řetězec délky n, pak lze zjistit zda $X \in L(G)$ v čase $\mathcal{O}(n^3)$.

Normální tvar podle Greibachové

Definice

Neterminální symbol A v BG $G = (N, T, P, S)$ nazveme *rekurzivní*, jestliže existuje derivace $A \Rightarrow^+ \alpha A \beta$ pro nějaké α a $\beta \in (N \cup T)^*$.
Jestliže $\alpha = \varepsilon$, pak A nazveme *symbol rekurzivní zleva*, podobně jestliže $\beta = \varepsilon$, pak A nazveme *symbol rekurzivní zprava*.
Gramatiku s alespoň jedním neterminálním symbolem rekurzivním zleva (zprava) nazveme *rekurzivní zleva (zprava)*.
Gramatiku, ve které je alespoň jeden neterminální symbol rekurzivní, nazveme *rekurzivní*.

Normální tvar podle Greibachové

Věta

(O vyloučení rekurzivity zleva z jednotlivých pravidel).
Nechť $G = (N, T, P, S)$ je bezkontextová gramatika, ve které
$A \rightarrow A\,\alpha_1 \mid A\,\alpha_2 \mid \ldots \mid A\,\alpha_m \mid \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n$
jsou všechna pravidla v P s neterminálním symbolem A na levé straně a žádné $\beta_i, i = 1, 2, \ldots, n$, nezačíná symbolem A.
Nechť $G' = (N \cup \{A'\}, T, P', S)$ je bezkontextová gramatika, kde P' je množina P, ve které pravidla uvedená výše jsou nahrazena pravidly:
$A \rightarrow \beta_1 \mid \beta_2 \mid \ldots \mid \beta_n \mid \beta_1 A' \mid \beta_2 A' \mid \ldots \mid \beta_n A'$
$A' \rightarrow \alpha_1 \mid \alpha_2 \mid \ldots \mid \alpha_m \mid \alpha_1 A' \mid \alpha_2 A' \mid \ldots \mid \alpha_m A'$,
kde A' je nový neterminální symbol, který není v N. Pak
$L(G') = L(G)$.

Normální tvar podle Greibachové

Příklad

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

$E \to E + T \mid T$

$T \to T * F \mid F$

$F \to (E) \mid a$.

Odstraníme levou rekurzi:

$E \to T \mid TE'$

$E' \to +T \mid +TE'$

$T \to F \mid FT'$

$T' \to *F \mid *FT'$

$F \to (E) \mid a$

$G' = (\{E, E', T', T, F\}, \{+, *, (,), a\}, P', E)$, P' je množina pravidel uvedených výše.

Normální tvar podle Greibachové

Definice

Bezkontextová gramatika G je v normálním tvaru podle Greibachové, jestliže G je bez ε-pravidel a každé pravidlo, které neobsahuje prázdný řetězec na pravé straně, má tvar $A \to a\alpha$, kde $a \in T, \alpha \in N^*$. $\square$

Normální tvar podle Greibachové

Algoritmus Vyloučení rekurzivity zleva.
Vstup: Vlastní bezkontextová gramatika $G = (N, T, P, S)$ bez jednoduchých pravidel.
Výstup: BG G' bez rekurzivity zleva, $L(G) = L(G')$.
Metoda:

1. Zvolíme uspořádání $N = \{A_1, \ldots, A_r\}$: $A_i \to \alpha$, α začíná buď terminálním symbolem nebo neterminálním symbolem A_j pro $j \geq i$.
   Položme $i := 1$.

Normální tvar podle Greibachové

Algoritmus (pokračování):

2. Všechna pravidla $A_i \to A_i\alpha_1 \mid \ldots \mid A_i\alpha_m \mid \beta_1 \mid \ldots \mid \beta_n$ z P s neterminálním symbolem A_i na levé straně, kde žádné β_j nezačíná neterminálním symbolem A_k pro $k \leq i$, nahradíme tato pravidla pravidly:
   $A_i \to \beta_1 \mid \ldots \mid \beta_n \mid \beta_1 A'_i \mid \ldots \mid \beta_n A'_i$,
   $A'_i \to \alpha_1 \mid \ldots \mid \alpha_m \mid \alpha_1 A'_i \mid \ldots \mid \alpha_m A'_i$, kde A'_i je nový neterminální symbol.
3. Jestliže $i = r$, máme výslednou gramatiku G' a končíme, jinak $i := i + 1$ a $j := 1$.
4. Jestliže pro neterminální symbol A_j existují pravidla $A_j \to \beta_1 \mid \ldots \mid \beta_m$, pak nahradíme všechna pravidla tvaru $A_i \to A_j\alpha$ pravidly $A_i \to \beta_1\alpha \mid \ldots \mid \beta_m\alpha$.
5. Jestliže $j = i - 1$, jdi na krok 2., jinak polož $j := j + 1$ a jdi na krok 4.

Normální tvar podle Greibachové

Věta

Každý bezkontextový jazyk může být generován gramatikou, která neobsahuje rekurzivitu zleva. $\square$

Normální tvar podle Greibachové

Příklad

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

$A \rightarrow BC \mid a$

$B \rightarrow CA \mid Ab$

$C \rightarrow AB \mid CC \mid a$. Aplikujme algoritmus 3.28 na tuto gramatiku.

Položme $A_1 = A, A_2 = B, A_3 = C$.

Krok 2: $(i = 1)$ beze změn.

Krok 4: $(i = 2, j = 1)$

Po dosazení za A dostaneme pro B pravidla: $B \rightarrow CA \mid BCb \mid ab$.

Krok 2: Odstraníme rekurzivitu zleva u symbolu B :

$B \rightarrow CA \mid ab \mid CA\,B' \mid abB'$

$B' \rightarrow CbB' \mid Cb$

Krok 4: $(i = 3, j = 1)$

$C \rightarrow BCB \mid aB \mid CC \mid a$

Normální tvar podle Greibachové

Příklad (pokračování)

Krok 4: $(i = 3, j = 2)$

$C \to CACB \mid abCB \mid CAB'CB \mid abB'CB \mid aB \mid CC \mid a$

Krok 2: $(i = 3)$

$C \to abCB \mid abB'CB \mid aB \mid a \mid abCBC' \mid abB'CBC' \mid aBC'$
$\quad \mid aC'$

$C' \to A\ CBC' \mid AB'CBC' \mid CC' \mid A\ CB \mid A\ B'CB \mid C$

Výsledná gramatika je $G' = (\{A, B, C, B', C'\}, \{a, b\}, P', A)$, kde P' obsahuje pravidla:

$A \to BC \mid a$

$B \to CA \mid ab \mid CA\ B' \mid abB'$

$B' \to CbB' \mid Cb$

$C \to abCB \mid abB'CB \mid aB \mid a \mid abCBC' \mid abB'CBC' \mid aBC'$
$\quad \mid aC'$

$C' \to ACBC' \mid AB'CBC' \mid CC' \mid ACB \mid AB'CB \mid C$