AKADEMIA GÓRNICZO-HUTNICZA
IM. STANISŁAWA STASZICA W KRAKOWIE

Parsery LR(1) – część 2

Teoria kompilacji

Dr inż. Janusz Majewski
Katedra Informatyki

Przedrostki żywotne
Sytuacje dopuszczalne

Żywotny przedrostek (viable prefix)

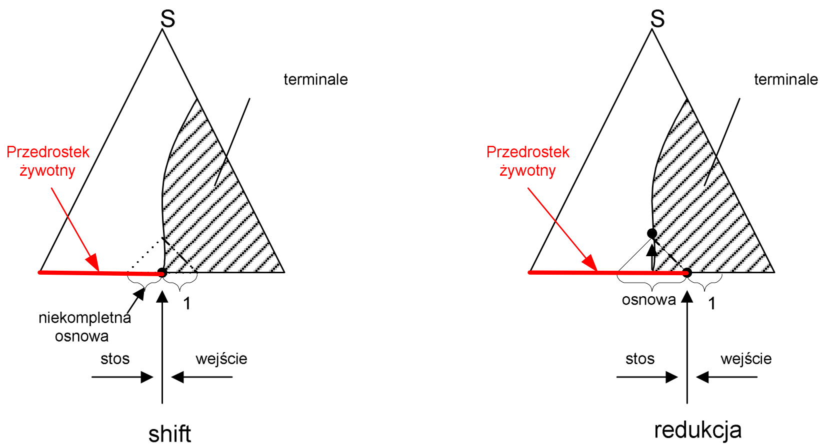
γ - żywotny (aktywny) prefiks gramatyki G
$\Leftrightarrow$ γ - prefiks łańcucha $\alpha\beta$

$$S \underset{R}{\overset{*}{\Rightarrow}} \alpha A w \underset{R}{\Rightarrow} \alpha\beta w$$

gdzie: $\alpha, \beta, \gamma \in (\Sigma \cup V)^*$ $w \in \Sigma^*$ $A \in V$

Żywotny przedrostek jest to łańcuch będący przedrostkiem pewnej prawostronnie wyprowadzalnej formy zdaniowej, nie wychodzący poza prawy koniec jej osnowy.

Przedrostki żywotne
Sytuacje dopuszczalne



Przedrostki żywotne
Sytuacje dopuszczalne

<u>LR(1)–sytuacja (LR(1)–item)</u>

$[A \to \beta_1 \bullet \beta_2, v]$ - jest LR(1)–sytuacją, gdy $(A \to \beta_1\beta_2) \in P$

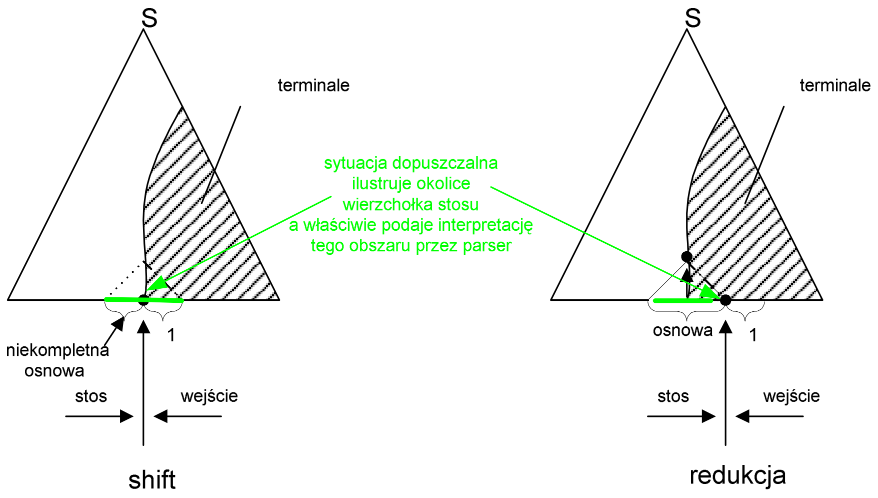
<u>LR(1)–sytuacja dopuszczalna (LR(1)–item valid for viable prefix)</u>

$[A \to \beta_1 \bullet \beta_2, v]$ – LR(1)–sytuacja jest sytuacją dopuszczalną dla żywotnego prefiksu $\alpha\beta_1$ wtedy i tylko wtedy, gdy

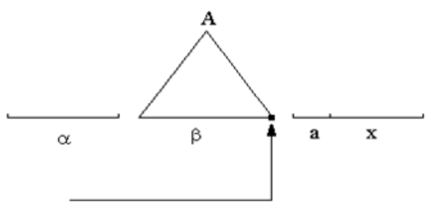
$\exists$ wywód:

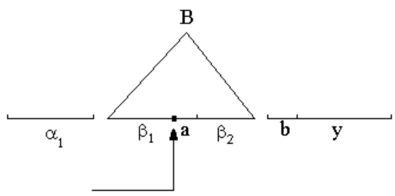
$$\left(S \overset{*}{\underset{R}{\Rightarrow}} \alpha A w \underset{R}{\Rightarrow} \alpha\beta_1\beta_2 w \quad \wedge \quad v \in FIRST_1(w) \right)$$

Przedrostki żywotne
Sytuacje dopuszczalne



Przedrostki żywotne
Sytuacje dopuszczalne





Sytuacja ***[A → β•, a]*** dopuszczalna dla żywotnego przedrostka $\alpha\beta$

<u>Decyzja</u>: redukcja wg produkcji $A \rightarrow \beta$ przy podglądaniu *a* na wejściu

Efekt: nowa konfiguracja z żywotnym przedrostkiem αA

Sytuacja $\boldsymbol{[B \rightarrow \beta_1 \bullet a \beta_2, b]}$ dopuszczalna dla żywotnego przedrostka $\alpha_1\beta_1$, przy czym $a \in \Sigma$.

<u>Decyzja</u>: przesunięcie (shift) terminala *a* z wejścia na stos

Efekt: nowa konfiguracja opisana sytuacją: $\boldsymbol{[B \rightarrow \beta_1 a \bullet \beta_2, b]}$ dopuszczalną dla żywotnego przedrostka $\alpha_1\beta_1 a$

Przykład

Gramatyka: $S \to SaSb \mid \varepsilon$ analizowane słowo: *aabb*

$$S \Rightarrow^* \underbrace{S}_{S}\; \underbrace{Sa}_{\alpha}\,\underbrace{S}_{A}\,\underbrace{b}_{w} \Rightarrow \underbrace{Sa}_{\alpha}\; \underbrace{SaS}_{\beta_1}\,\underbrace{b}_{\beta_2}\,\underbrace{b}_{w}$$

Forma zdaniowa:	Stos (bez stanów)	Wejście	Sytuacja
SaSaSbb	(*) *SaSaS*	↓*bb*	$[S \to SaS\bullet b,\ b]$ dopuszczalna dla żywotnego prefiksu *SaSaS*
SaSaSbb	(**) *SaSaSb*	↓*b*	$[S \to SaSb\bullet,\ b]$ dopuszczalna dla żywotnego prefiksu *SaSaSb*

(*) Żywotnemu prefiksowi *SaSaS* odpowiada stan T_6

(**) Żywotnemu prefiksowi *SaSaSb* odpowiada stan T_7

Domykanie zbioru LR(1)-sytuacji dopuszczalnych

Załóżmy, że sytuacja $[A \to \alpha \bullet B\beta,\ a]$ jest dopuszczalna dla pewnego żywotnego prefiksu γ, co oznacza, że istnieje wyprowadzenie prawostronne:

$$S \overset{*}{\underset{R}{\Rightarrow}} \delta A a x \underset{R}{\Rightarrow} \underbrace{\delta\alpha}_{\gamma} B\beta a x$$

Przypuśćmy, że $\beta a x \underset{R}{\Rightarrow} by$ (*by* – łańcuch terminalny rozpoczynający się symbolem *b*). Wtedy dla każdej produkcji $B \to \eta$

$$S \overset{*}{\underset{R}{\Rightarrow}} \delta A a x \underset{R}{\Rightarrow} \underbrace{\delta\alpha}_{\gamma} B\beta a x \underset{R}{\Rightarrow} \underbrace{\delta\alpha}_{\gamma} \eta\beta a x \underset{R}{\Rightarrow} \underbrace{\delta\alpha}_{\gamma} \eta b y$$

Czyli dla żywotnego prefiksu γ dopuszczalna jest także sytuacja $[B \to \bullet\eta,\ b]$, gdzie:

$$b \in FIRST_1(\beta a x) = FIRST_1(\beta a)$$

Algorytm domykania zbioru sytuacji dopuszczalnych

We: Zbiór I sytuacji dopuszczalnych dla pewnego żywotnego prefiksu (w gramatyce uzupełnionej G')
Wy: Zbiór I będący domknięciem wejściowego zbioru sytuacji dopuszczalnych
Metodę ilustruje funkcja CLOSURE(I);

function CLOSURE(I);
begin
    repeat
        for każda sytuacja $[A \to \alpha \bullet B\beta,\ a] \in I$ do
            for każda produkcja $(B \to \eta) \in P'$ do
                for każdy $b \in FIRST_1(\beta a)$ do
                    $I := I \cup \{[B \to \bullet\eta,\ b]\}$;
    until nic nowego nie dodano do I;
    return (I);
end;

Funkcja GOTO

Funkcja GOTO:
$\{I(\gamma)\colon \gamma$ - żywotny prefiks$\} \times \{X\colon X \in (V \cup \Sigma)\} \mapsto \{I(\gamma')\colon \gamma' \in (V \cup \Sigma)^+,\ \gamma' = \gamma X\}$
gdzie: $I(\gamma)$ – zbiór wszystkich sytuacji dopuszczalnych dla prefiksu żywotnego γ

Przykład: $S \to SaSb \mid \varepsilon$

(a) $S \overset{*}{\underset{R}{\Rightarrow}} SaSaS \bullet bb$
(b) $S \overset{*}{\underset{R}{\Rightarrow}} SaSaS \bullet abbb$

Prefiksem żywotnym obu form zdaniowych jest $SaSaS$

Przykład funkcji GOTO

I_6={[$S \to SaS \bullet b$, a]
[$S \to SaS \bullet b$, b]
[$S \to S \bullet aSb$, a]
[$S \to S \bullet aSb$, b]}
I_6 – zbiór sytuacji dopuszczalnych dla prefiksu żywotnego *SaSaS*

$\boldsymbol{I_7 = GOTO(I_6, b)}$

I_7={[$S \to SaSb \bullet$, a]
[$S \to SaSb \bullet$, b]}
I_7 – zbiór sytuacji dopuszczalnych dla prefiksu żywotnego *SaSaSb*

$\boldsymbol{I_4 = GOTO(I_6, a)}$

I_4={[$S \to Sa \bullet Sb$, a]
[$S \to Sa \bullet Sb$, b]
[$S \to \bullet SaSb$, a]
[$S \to \bullet SaSb$, b]
[$S \to \bullet$, a]
[$S \to \bullet$, b]}
I_4 – zbiór sytuacji dopuszczalnych dla prefiksu żywotnego *SaSaSa*

Wyznaczanie funkcji GOTO

<u>We:</u> I – zbiór wszystkich sytuacji dopuszczalnych dla prefiksu aktywnego γ , $X \in (V \cup \Sigma)$
<u>Wy:</u> J – zbiór wszystkich sytuacji dopuszczalnych dla prefiksu aktywnego γX
<u>Metodę</u> ilustruje funkcja GOTO (I , X)

<u>function</u> GOTO (I , X);
<u>begin</u>
$J := \varnothing$;
<u>for</u> każda_sytuacja $[A \to \alpha \bullet X \beta, a] \in I$ <u>do</u>
$J := J \cup \{ [A \to \alpha X \bullet \beta, a] \}$;
<u>return</u> CLOSURE (J);
<u>end</u>;

Kanoniczny system zbiorów LR(1)-sytuacji dopuszczalnych

$\mathcal{J}$ – kanoniczny system zbiorów LR(1) sytuacji dopuszczalnych

$\mathcal{J}$ – jest zbiorem wszystkich zbiorów *I(γ)* LR(1)-sytuacji dopuszczalnych,

gdzie : γ – żywotny prefiks w gramatyce *G'*

Konstrukcja kanonicznego systemu zbiorów LR(1)-sytuacji dopuszczalnych

We: G' – gramatyka uzupełniona <V', Σ, P', S'>
dla gramatyki G = < V, Σ, P, S > $\in \mathcal{G}_{BK}$

Wy: $\mathcal{J}$ – kanoniczny system zbiorów LR(1) sytuacji dopuszczalnych dla G.

Metodę ilustruje funkcja ITEMS (G');

```
function ITEMS (G');
begin
        J := {CLOSURE ({[S' → •S, $]})};
        repeat
          for każdy zbiór I ∈ J do
            for każdy X ∈ (V ∪ Σ) do
              if GOTO(I , X) ≠ ∅ then
               J := J ∪ {GOTO(I ,X)};
        until nic nowego nie dodano do J;
        return J;
end;
```

Przykład

Wyznaczanie systemu kanonicznego zbiorów LR(1)-sytuacji dopuszczalnych dla gramatyki uzupełnionej

$S' \to S$

$S \to SaSb \mid \varepsilon$

$I_0 = I(\varepsilon) = \{[S' \to \bullet S, \$],$ ⟶ $FIRST_1(\$) = \{\$\}$

(domykamy)

$[S \to \bullet SaSb, \$],$ ⟶ $FIRST_1(aSb\$) = \{a\}$

$[S \to \bullet, \$],$

(domykamy powtórnie)

$[S \to \bullet SaSb, a]$ ⟶ $FIRST_1(aSba) = \{a\}$

$[S \to \bullet, a]\}$

(nic nowego nie da się dołączyć)

Przykład c.d.

Uproszczenie zapisu:

zamiast: $[A \to \beta_1 \bullet \beta_2, x], [A \to \beta_1 \bullet \beta_2, y]$

piszemy: $[A \to \beta_1 \bullet \beta_2, x \mid y]$)

Ostatecznie:

$I_0 = \{[S \to \bullet S, \$],$

$[S \to \bullet SaSb, \$ \mid a],$

$[S \to \bullet, \$ \mid a]\}$

Przykład c.d.

$I_o = \{[S \rightarrow \bullet S\ ,\ \$]\ ,\ [S \rightarrow \bullet SaSb,\ \$ \mid a],\ [S \rightarrow \bullet,\ \$ \mid a]\}$

$I_1 = I\ (S) = GOTO(I_o\ ,\ S) = GOTO(I(\varepsilon),\ S)$

$I_1 = \{[S' \rightarrow S\bullet,\ \$],$

$[\ S \rightarrow S\bullet aSb,\ \$ \mid a]\}$ (domknięcie nie daje nowych sytuacji)

$GOTO(I_o,\ a) = GOTO(I(\varepsilon),\ a) = \varnothing$

$GOTO(I_o,\ b) = GOTO(I(\varepsilon),\ b) = \varnothing$

(gdyż ani „a” ani „b” nie są żywotnymi prefiksami w tej gramatyce)

Przykład c.d.

$I_1 = \{[S' \rightarrow S\bullet,\ \$],\ [\ S \rightarrow S\bullet aSb,\ \$ \mid a]\}$

$I_2 = I\ (Sa) = GOTO\ (I(S),\ a) = GOTO(I_1,\ a)$

$I_2 = \ \{[S \rightarrow Sa\bullet Sb,\ \$ \mid a],$ $\longrightarrow$ $FIRST_1(b\$) = \{b\}$, $FIRST_1(ba) = \{b\}$

(domykamy)

$[S \rightarrow \bullet SaSb,\ b]$ $\longrightarrow$ $FIRST_1(aSbb) = \{a\}$

$[S \rightarrow \bullet,\ b]\ ,$

(domykamy powtórnie)

$[S \rightarrow \bullet SaSb,\ a]$ $\longrightarrow$ $FIRST_1(aSba) = \{a\}$

$[\ S \rightarrow \bullet,\ a]\}$

(nic nowego nie da się dołączyć)

$I_2 = \{[S \rightarrow Sa\bullet Sb,\ \$ \mid a],\ [S \rightarrow \bullet SaSb,\ b \mid a],\ [S \rightarrow \bullet,\ b \mid a]\}$

Przykład c.d.

$I_1 = \{[S' \to S\bullet, \$], [\, S \to S\bullet aSb, \$ \mid a]\}$

GOTO(I_1, S) = GOTO(I(S), S) = ∅

GOTO(I_1, b) = GOTO(I(S), b) = ∅

(gdyż ani „SS" ani „Sb" nie są żywotnymi prefiksami w tej gramatyce)

$I_2 = \{[S \to Sa\bullet Sb, \$ \mid a], [S \to \bullet SaSb, b \mid a], [S \to \bullet, b \mid a]\}$

$I_3 = I(SaS) = GOTO(I(Sa), S) = GOTO(I_2, S)$

$I_3 = \{[S \to SaS\bullet b\,, \$ \mid a]\,, [S \to S\bullet aSb, b \mid a]\}$

(domknięcie nie daje nowych sytuacji)

GOTO(I_2, a) = GOTO(I_2, b) = ∅

Przykład c.d.

$I_3 = \{[S \to SaS\bullet b, \$ \mid a], [S \to S\bullet aSb, b \mid a]\}$

$I_4 = I(SaSa) = GOTO(I(SaS), a) = GOTO(I_3, a)$

$I_4 = \{[S \to Sa\bullet Sb, b \mid a], [S \to \bullet SaSb, b \mid a], [S \to \bullet, b \mid a]\}$

(**Uwaga:** Podobny zbiór sytuacji miał nr 2 i różnił się tylko postacią prawych stron sytuacji)

$I_5 = I(SaSb) = GOTO(I(SaS), b) = GOTO(I_3, b)$

$I_5 = \{[S \to SaSb\bullet, \$ \mid a]\}$

GOTO(I_3, S) = ∅

$I_6 = I(SaSaS) = GOTO(I(SaSa), S) = GOTO(I_4, S)$

$I_6 = \{[S \to SaS\bullet b, b \mid a], [S \to S\bullet aSb, b \mid a]\}$

(Porównaj zbiór I_3)

Przykład c.d.

$I_4 = \{[S \to Sa\bullet Sb,\ b \mid a],\ [S \to \bullet SaSb,\ b \mid a],\ [S \to \bullet,\ b \mid a]\}$
$I_5 = \{[S \to SaSb\bullet,\ \$ \mid a]\}$
$I_6 = \{[S \to SaS\bullet b,\ b \mid a],\ [S \to S\bullet aSb,\ b \mid a]\}$

$\mathrm{GOTO}(I_4, a) = \mathrm{GOTO}(I_4, b) = \varnothing$
$\mathrm{GOTO}(I_5, a) = \mathrm{GOTO}(I_5, b) = \mathrm{GOTO}(I_5, b) = \varnothing$

$I_7 = I(SaSaSb) = GOTO(I\ (SaSaS), b) = GOTO(I_6, b)$
$I_7 = \{[S \to SaSb\bullet,\ b \mid a]\}$ (Porównaj zbiór I_5)

$I(SaSaSa) = GOTO(I(SaSaS), a) = GOTO(I_6, a) = \boldsymbol{I_4}$
(**Uwaga:** tutaj otrzymaliśmy zbiór sytuacji identycznych ze zbiorem I_4 dla prefiksu aktywnego „SaSa”)

$\mathrm{GOTO}(I_6, S) = \varnothing$
$\mathrm{GOTO}(I_7, S) = \mathrm{GOTO}(I_7, a) = \mathrm{GOTO}(I_7, b) = \varnothing$

Zgodny zbiór sytuacji dopuszczalnych

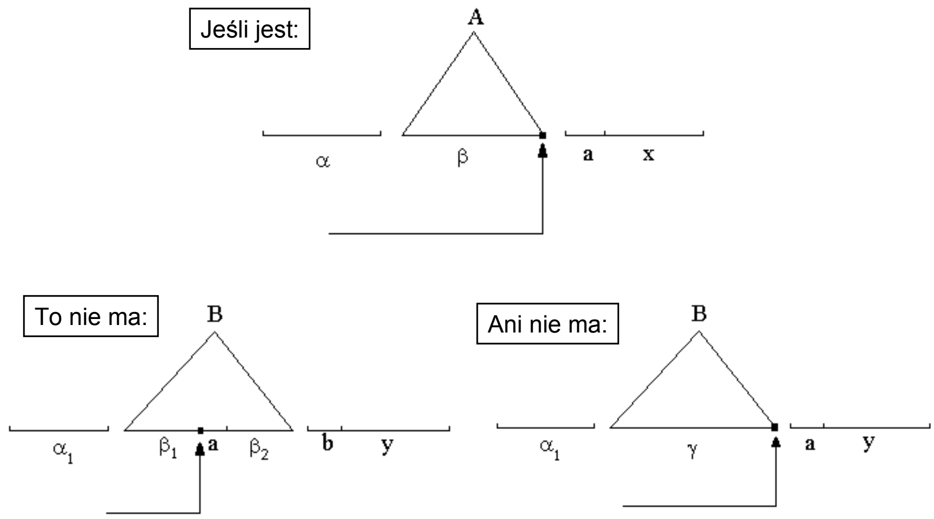
$G = < V, \Sigma, P, S > \in \mathcal{G}_{BK}$
I – zbiór LR(1)-sytuacji dla gramatyki G

Definicja: Zbiór sytuacji I jest zgodny
$\Leftrightarrow$ nie zawiera dwu różnych sytuacji:

(i) typu $[A \to \beta\bullet,\ a]$
i $[B \to \beta_1\bullet a\beta_2,\ b]$

(ii) typu $[A \to \beta\bullet,\ a]$
i $[B \to \gamma\bullet,\ a]$

Zgodny zbiór sytuacji dopuszczalnych



Czy gramatyka jest LR(1)?

<u>Twierdzenie:</u> Niech $G = < V, \Sigma, P, S > \in \mathcal{G}_{BK}$

$\mathcal{J}$ – kanoniczny system zbiorów LR(1)-sytuacji dopuszczalnych dla G

G jest LR(1)-gramatyką $\Leftrightarrow (\forall I \in \mathcal{J})$ (I – jest zgodny)

Twierdzenie to stanowi najczęściej stosowany sposób przeprowadzenia weryfikacji, czy gramatyka G jest klasy LR(1).

Konstruowanie tablic parsera LR(1)

We: G' – gramatyka uzupełniona dla gramatyki bezkontekstowej G
Wy: Tablica parsera LR(1) – funkcje ***f*** i ***g***

Metoda:

1. Konstruujemy $\mathcal{J}$ – kanoniczny system zbiorów LR(1) sytuacji dopuszczalnych dla G'
2. Badamy zgodność każdego zbioru $I \in \mathcal{J}$
   Jeśli choć jeden zbiór I nie jest zgodny, gramatyka nie jest LR(1) ↔ **STOP !!!**
   Numerujemy następnie produkcję gramatyki G'.

Konstruowanie tablic parsera LR(1)

3. for każdy zbiór $I_j \in \mathcal{J}$ do
   begin
   utwórz w tablicy parsera stan $T_j \in \mathfrak{T}$ dla analizowanego $I_j \in \mathcal{J}$;
   for każda sytuacja ze zbioru I_j do
   begin
   (a) if $[A \rightarrow \alpha \bullet a\beta,\ b] \in I_j$ and $a \in \Sigma$ and $GOTO(I_j, a) = I_k$ then
   $f(T_j, a) :=$ *shift* $- k$;
   (b) if $[A \rightarrow \alpha \bullet,\ a] \in I_j$ and $A \neq S'$ and i-numer produkcji $(A \rightarrow \alpha) \in P$ then
   $f(T_j, a) :=$ *red* $- i$;
   (c) if $[S' \rightarrow S \bullet,\ \$] \in I_j$ then
   $f(T_j, \$) :=$ *acc*;
   end;
   for każdy $A \in V$ do
   if $GOTO(I_j, A) \neq \varnothing$ and $GOTO(I_j, A) = I_k$ then
   $g(T_j, A) := T_k$;
   end;

Konstruowanie tablic parsera LR(1)

4. for każdy $T_j \in \mathfrak{I}$ do
   begin
   for każdy $a \in \Sigma \cup \{\$\}$ do
   if $f(T_j, a)$ nieokreślone then
   $f(T_j, a) :=$ *err* ;
   for każdy $A \in V$ do
   if $g(T_j, A)$ nieokreślone then
   $g(T_j, A) :=$ *err* ;
   end;
5. Stanem początkowym parsera jest ten stan, który odpowiada zbiorowi $I \in \mathcal{J}$, dla którego $[S' \rightarrow \bullet S\ ,\ \$] \in I$;

Gramatyki i parsery LR(0)

Omawiane postępowanie dotyczy gramatyk LR(1). Gramatyki LR(0) definiują stosunkowo wąską klasę języków bezkontekstowych posiadających własność przedrostkową. Poza tym dają z reguły tablice parsera o wyraźnie większym rozmiarze niż LR(1). Ponieważ postępowanie prowadzące do konstrukcji parsera LR(0) różni się nieco od przedstawionego powyżej (wymaga innego zdefiniowania funkcji ***f*** i ***g*** parsera), nie będzie przedmiotem naszego zainteresowania.

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: *(0) S' → S*
(1) S → SaSb
(2) S → ε

I_o = {[S'→•S , $] ,
[S →•SaSb , $ | a] ,
[S →•, $ | a]}

I_1 = {[S' →S•, $] ,
[S →S•aSb, $ | a]}

I_2 = {[S →Sa•Sb, $ | a] ,
[S →•SaSb, a | b] ,
[S →•, a | b]}

$I_1 = GOTO(I_o, S)$

$I_2 = GOTO(I_1, a)$

$I_3 = GOTO(I_2, S)$

	f			g
	a	b	$	S
T_o	**red – 2**		**red – 2**	T_1
T_1	shift – 2		acc	
T_2	red – 2	red – 2		T_3
…				

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: *(0) S' → S*
(1) S → SaSb
(2) S → ε

I_o = **{[S'→•S , $] ,**
[S →•SaSb , $ | a] ,
[S →•, $ | a]}

I_1 = {[S' →S•, $] ,
[S →S•aSb, $ | a]}

I_2 = {[S →Sa•Sb, $ | a] ,
[S →•SaSb, a | b] ,
[S →•, a | b]}

$I_1 = GOTO(I_o, S)$

$I_2 = GOTO(I_1, a)$

$I_3 = GOTO(I_2, S)$

	f			g
	a	b	$	S
T_o	red – 2		red – 2	**T_1**
T_1	shift – 2		acc	
T_2	red – 2	red – 2		T_3
…				

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: $(0)\ S' \to S$
$(1)\ S \to SaSb$
$(2)\ S \to \varepsilon$

$I_o = \{[S' \to \bullet S\ ,\ \$]\ ,$
$[S \to \bullet SaSb\ ,\ \$ \mid a]\ ,$
$[S \to \bullet,\ \$ \mid a]\}$

$I_1 = GOTO(I_o, S)$

$I_1 = \{[S' \to S\bullet,\ \$]\ ,$
$[S \to S\bullet aSb,\ \$ \mid a]\}$

$I_2 = GOTO(I_1, a)$

$I_2 = \{[S \to Sa\bullet Sb,\ \$ \mid a]\ ,$
$[S \to \bullet SaSb,\ a \mid b]\ ,$
$[S \to \bullet,\ a \mid b]\}$

$I_3 = GOTO(I_2, S)$

	f			*g*
	a	*b*	*$*	*S*
T_o	<u>red</u> – 2		<u>red</u> – 2	T_1
T_1	<u>shift</u> – 2		**<u>acc</u>**	
T_2	<u>red</u> – 2	<u>red</u> – 2		T_3
…				

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: $(0)\ S' \to S$
$(1)\ S \to SaSb$
$(2)\ S \to \varepsilon$

$I_o = \{[S' \to \bullet S\ ,\ \$]\ ,$
$[S \to \bullet SaSb\ ,\ \$ \mid a]\ ,$
$[S \to \bullet,\ \$ \mid a]\}$

$I_1 = GOTO(I_o, S)$

$I_1 = \{[S' \to S\bullet,\ \$]\ ,$
$[S \to S\bullet aSb,\ \$ \mid a]\}$

$I_2 = GOTO(I_1, a)$

$I_2 = \{[S \to Sa\bullet Sb,\ \$ \mid a]\ ,$
$[S \to \bullet SaSb,\ a \mid b]\ ,$
$[S \to \bullet,\ a \mid b]\}$

$I_3 = GOTO(I_2, S)$

	f			*g*
	a	*b*	*$*	*S*
T_o	<u>red</u> – 2		<u>red</u> – 2	T_1
T_1	**<u>shift</u> – 2**		<u>acc</u>	
T_2	<u>red</u> – 2	<u>red</u> – 2		T_3
…				

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: (0) $S' \to S$
(1) $S \to SaSb$
(2) $S \to \varepsilon$

$I_o = \{[S' \to \bullet S\,,\ \$]\,,$ $\quad I_1 = GOTO(I_o, S)$
$[S \to \bullet SaSb\,,\ \$ \mid a]\,,$
$[S \to \bullet,\ \$ \mid a]\}$

$I_1 = \{[S' \to S\bullet,\ \$]\,,$ $\quad I_2 = GOTO(I_1, a)$
$[S \to S\bullet aSb,\ \$ \mid a]\}$

$I_2 = \{[S \to Sa\bullet Sb,\ \$ \mid a]\,,$ $\quad I_3 = GOTO(I_2, S)$
$[S \to \bullet SaSb,\ a \mid b]\,,$
$[S \to \bullet,\ a \mid b]\}$

	f			g
	a	b	\$	S
T_o	red – 2		red – 2	T_1
T_1	shift – 2		acc	
T_2	red – 2	red – 2		T_3
…				

Przykład

Konstrukcja tablicy parsera LR(1) dla gramatyki: (0) $S' \to S$
(1) $S \to SaSb$
(2) $S \to \varepsilon$

$I_o = \{[S' \to \bullet S\,,\ \$]\,,$ $\quad I_1 = GOTO(I_o, S)$
$[S \to \bullet SaSb\,,\ \$ \mid a]\,,$
$[S \to \bullet,\ \$ \mid a]\}$

$I_1 = \{[S' \to S\bullet,\ \$]\,,$ $\quad I_2 = GOTO(I_1, a)$
$[S \to S\bullet aSb,\ \$ \mid a]\}$

$I_2 = \{[S \to Sa\bullet Sb,\ \$ \mid a]\,,$ $\quad I_3 = GOTO(I_2, S)$
$[S \to \bullet SaSb,\ a \mid b]\,,$
$[S \to \bullet,\ a \mid b]\}$

	f			g
	a	b	\$	S
T_o	red – 2		red – 2	T_1
T_1	shift – 2		acc	
T_2	**red – 2**	**red – 2**		T_3
…				