Compiler Construction

Lecture 12: Syntactic Analysis VII ($SLR(1)$ Parsing)

Thomas Noll

Lehrstuhl für Informatik 2
(Software Modeling and Verification)

RWTH Aachen University

noll@cs.rwth-aachen.de

http://www-i2.informatik.rwth-aachen.de/i2/cc10/

Winter semester 2010/11

Outline

$LR(0)$ Items and Sets

Definition ($LR(0)$ items and sets)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_\Sigma$ be start separated by $S' \to S$ and $S' \Rightarrow_r^* \alpha A w \Rightarrow_r \alpha\beta_1\beta_2 w$ (i.e., $A \to \beta_1\beta_2 \in P$).

- $[A \to \beta_1 \cdot \beta_2]$ is called an $LR(0)$ item for $\alpha\beta_1$.
- Given $\gamma \in X^*$, $LR(0)(\gamma)$ denotes the set of all $LR(0)$ items for γ, called the $LR(0)$ set (or: $LR(0)$ information) of γ.
- $LR(0)(G) := \{LR(0)(\gamma) \mid \gamma \in X^*\}$.

Corollary

1. *For every $\gamma \in X^*$, $LR(0)(\gamma)$ is finite.*
2. *$LR(0)(G)$ is finite.*
3. *The item $[A \to \beta\cdot] \in LR(0)(\gamma)$ indicates the possible reduction $(w, \alpha\beta, z) \vdash (w, \alpha A, zi)$ where $\pi_i = A \to \beta$ and $\gamma = \alpha\beta$.*
4. *The item $[A \to \beta_1 \cdot Y\beta_2] \in LR(0)(\gamma)$ indicates an incomplete handle β_1 (to be completed by shift operations or ε-reductions).*

$LR(0)$ Conflicts

Definition ($LR(0)$ conflicts)

Let $G = \langle N, \Sigma, P, S\rangle \in CFG_\Sigma$ and $I \in LR(0)(G)$.

- I has a shift/reduce conflict if there exist $A \to \alpha_1 a \alpha_2, B \to \beta \in P$ such that
$$[A \to \alpha_1 \cdot a\alpha_2], [B \to \beta\cdot] \in I.$$
- I has a reduce/reduce conflict if there exist $A \to \alpha, B \to \beta \in P$ with $A \neq B$ or $\alpha \neq \beta$ such that
$$[A \to \alpha\cdot], [B \to \beta\cdot] \in I.$$

Lemma

$G \in LR(0)$ iff no $I \in LR(0)(G)$ contains conflicting items.

Proof.

omitted □

Computing $LR(0)$ Sets I

Theorem (Computing $LR(0)$ sets)

Let $G = \langle N, \Sigma, P, S \rangle \in CFG_\Sigma$ be start separated by $S' \to S$ and reduced.

1. $LR(0)(\varepsilon)$ *is the least set such that*
   - $[S' \to \cdot S] \in LR(0)(\varepsilon)$ *and*
   - *if* $[A \to \cdot B\gamma] \in LR(0)(\varepsilon)$ *and* $B \to \beta \in P$, *then* $[B \to \cdot\beta] \in LR(0)(\varepsilon)$.
2. $LR(0)(\alpha Y)$ *(*$\alpha \in X^*, Y \in X$*) is the least set such that*
   - *if* $[A \to \gamma_1 \cdot Y\gamma_2] \in LR(0)(\alpha)$, *then* $[A \to \gamma_1 Y \cdot \gamma_2] \in LR(0)(\alpha Y)$ *and*
   - *if* $[A \to \gamma_1 \cdot B\gamma_2] \in LR(0)(\alpha Y)$ *and* $B \to \beta \in P$, *then* $[B \to \cdot\beta] \in LR(0)(\alpha Y)$.

Example (cf. Example 11.2)

$$G: \quad \begin{array}{lcl} S' & \to & S \\ S & \to & B \mid C \\ B & \to & aB \mid b \\ C & \to & aC \mid c \end{array}$$

$[S' \to \cdot S] \in$

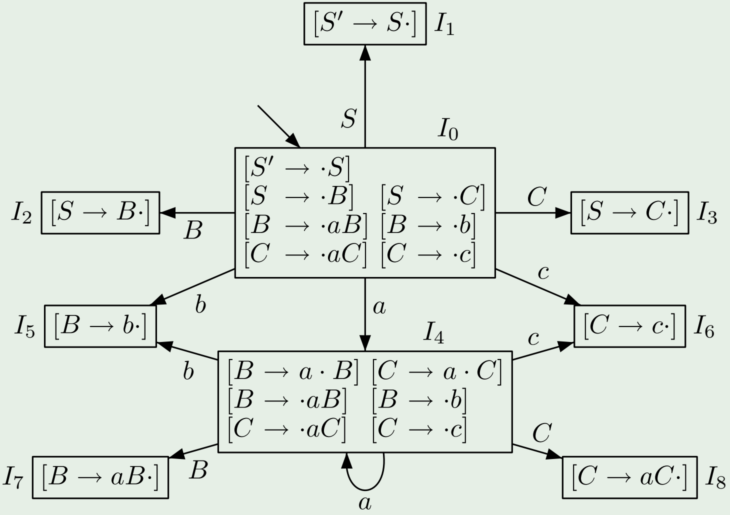
$LR(0)(\varepsilon)$ $\quad \begin{array}{l}[A \to \cdot B\gamma] \in LR(0)(\varepsilon), B \to \beta \in P \\ \implies [B \to \cdot\beta] \in LR(0)(\varepsilon)\end{array}$ $\quad \begin{array}{l}[A \to \gamma_1 \cdot Y\gamma_2] \in LR(0)(\alpha) \\ \implies [A \to \gamma_1 Y \cdot \gamma_2] \in LR(0)(\alpha Y)\end{array}$

$$\begin{array}{llllll}
I_0 := LR(0)(\varepsilon): & [S' \to \cdot S] & [S \to \cdot B] & [S \to \cdot C] & [B \to \cdot aB] \\
 & [B \to \cdot b] & [C \to \cdot aC] & [C \to \cdot c] & \\
I_1 := LR(0)(S): & [S' \to S\cdot] & & & \\
I_2 := LR(0)(B): & [S \to B\cdot] & & & \\
I_3 := LR(0)(C): & [S \to C\cdot] & & & \\
I_4 := LR(0)(a): & [B \to a \cdot B] & [C \to a \cdot C] & [B \to \cdot aB] & [B \to \cdot b] \\
 & [C \to \cdot aC] & [C \to \cdot c] & & \\
I_5 := LR(0)(b): & [B \to b\cdot] & & & \\
I_6 := LR(0)(c): & [C \to c\cdot] & & & \\
I_7 := LR(0)(aB): & [B \to aB\cdot] & & & \\
I_8 := LR(0)(aC): & [C \to aC\cdot] & & &
\end{array}$$

$(LR(0)(aa) = LR(0)(a) = I_4$, $LR(0)(ab) = LR(0)(b) = I_5$,
$LR(0)(ac) = LR(0)(c) = I_6$, $I_9 := LR(0)(\gamma) = \emptyset$ in all remaining cases)

Example (continued)

Representation of goto funtion as finite automaton:



Outline

Reduce/Reduce Conflicts

Example 12.1

$$G: \quad \begin{aligned} S' &\to S \\ S &\to Aa \mid Bb \\ A &\to a \\ B &\to a \end{aligned}$$

$$\begin{array}{lllll}
LR(0)(\varepsilon): & [S' \to \cdot S] & [S \to \cdot Aa] & [S \to \cdot Bb] & [A \to \cdot a] \quad [B \to \cdot a] \\
LR(0)(S): & [S' \to S\cdot] & & & \\
LR(0)(A): & [S \to A \cdot a] & & & \\
LR(0)(B): & [S \to B \cdot a] & & & \\
LR(0)(a): & [A \to a\cdot] & [B \to a\cdot] & & \\
LR(0)(Aa): & [S \to Aa\cdot] & & & \\
LR(0)(Ba): & [S \to Ba\cdot] & & &
\end{array}$$

Note: G is unambiguous

Shift/Reduce Conflicts

Example 12.2

$$G: \quad \begin{aligned} S' &\to S \\ S &\to aS \mid a \end{aligned}$$

$$\begin{array}{lllll} LR(0)(\varepsilon): & [S' \to \cdot S] & [S \to \cdot aS] & [S \to \cdot a] & \\ LR(0)(S): & [S' \to S\cdot] & & & \\ LR(0)(a): & [S \to a \cdot S] & [S \to \cdot aS] & \color{red}{[S \to \cdot a]} & \color{red}{[S \to a\cdot]} \\ LR(0)(aS): & [S \to aS\cdot] & & & \end{array}$$

Note: G is unambiguous

Outline

The $LR(0)$ Action Function

The automaton will be defined using another table, the action function, which determines the shift/reduce decision.
(Reminder: $\pi_0 = S' \to S$)

Definition 12.3 ($LR(0)$ action function)

The $LR(0)$ action function

$$\text{act} : LR(0)(G) \to \{\mathsf{red}\ i \mid i \in [p]\} \cup \{\mathsf{shift}, \mathsf{accept}, \mathsf{error}\}$$

is defined by

$$\text{act}(I) := \begin{cases} \mathsf{red}\ i & \text{if } \pi_i = A \to \alpha \text{ and } [A \to \alpha\cdot] \in I \ (i \neq 0) \\ \mathsf{shift} & \text{if } [A \to \alpha_1 \cdot a\alpha_2] \in I \\ \mathsf{accept} & \text{if } [S' \to S\cdot] \in I \\ \mathsf{error} & \text{if } I = \emptyset \end{cases}$$

Corollary 12.4

For every $G \in CFG_\Sigma$, $G \in LR(0)$ iff act *is well defined.*

Together, act and goto form the $LR(0)$ parsing table of G.

Example 12.5 (cf. Example 11.10)

$$
\begin{array}{llll}
G: & S' & \rightarrow S & (0) \\
 & S & \rightarrow B \mid C & (1,2) \\
 & B & \rightarrow aB \mid b & (3,4) \\
 & C & \rightarrow aC \mid c & (5,6)
\end{array}
$$

$LR(0)(G)$	act	goto					
		S	B	C	a	b	c
I_0	shift	I_1	I_2	I_3	I_4	I_5	I_6
I_1	accept						
I_2	red 1						
I_3	red 2						
I_4	shift		I_7	I_8	I_4	I_5	I_6
I_5	red 4						
I_6	red 6						
I_7	red 3						
I_8	red 5						
I_9	error						

(empty $= I_9$)

$$
\begin{array}{lll}
I_0 := LR(0)(\varepsilon): & [S' \rightarrow \cdot S] & \\
 & [S \rightarrow \cdot B] & [S \rightarrow \cdot C] \\
 & [B \rightarrow \cdot aB] & [B \rightarrow \cdot b] \\
 & [C \rightarrow \cdot aC] & [C \rightarrow \cdot c] \\
I_1 := LR(0)(S): & [S' \rightarrow S \cdot] & \\
I_2 := LR(0)(B): & [S \rightarrow B \cdot] & \\
I_3 := LR(0)(C): & [S \rightarrow C \cdot] & \\
I_4 := LR(0)(a): & [B \rightarrow a \cdot B] & [C \rightarrow a \cdot C] \\
 & [B \rightarrow \cdot aB] & [B \rightarrow \cdot b] \\
 & [C \rightarrow \cdot aC] & [C \rightarrow \cdot c] \\
I_5 := LR(0)(b): & [B \rightarrow b \cdot] & \\
I_6 := LR(0)(c): & [C \rightarrow c \cdot] & \\
I_7 := LR(0)(aB): & [B \rightarrow aB \cdot] & \\
I_8 := LR(0)(aC): & [C \rightarrow aC \cdot] & \\
I_9 := \emptyset & &
\end{array}
$$

The $LR(0)$ Parsing Automaton I

Definition 12.6 ($LR(0)$ parsing automaton)

Let $G = \langle N, \Sigma, P, S \rangle \in LR(0)$. The (deterministic) $LR(0)$ parsing automaton of G is defined by the following components.

- Input alphabet Σ
- Pushdown alphabet $\Gamma := LR(0)(G)$
- Output alphabet $\Delta := [p] \cup \{0, \mathsf{error}\}$
- Configurations $\Sigma^* \times \Gamma^* \times \Delta^*$
- Initial configuration (w, I_0, ε) where $I_0 := LR(0)(\varepsilon)$
- Final configurations $\{\varepsilon\} \times \{\varepsilon\} \times \Delta^*$
- Transitions:

shift: $(aw, \alpha I, z) \vdash (w, \alpha I I', z)$ if $\mathrm{act}(I) = \mathsf{shift}$ and $\mathrm{goto}(I, a) = I'$

reduce: $(w, \alpha I I_1 \ldots I_n, z) \vdash (w, \alpha I I', zi)$ if $\mathrm{act}(I_n) = \mathsf{red}\, i$, $\pi_i = A \to Y_1 \ldots Y_n$, and $\mathrm{goto}(I, A) = I'$

accept: $(\varepsilon, I_0 I, z) \vdash (\varepsilon, \varepsilon, z\,0)$ if $\mathrm{act}(I) = \mathsf{accept}$

error: $(w, \alpha I, z) \vdash (\varepsilon, \varepsilon, z\,\mathsf{error})$ if $\mathrm{act}(I) = \mathsf{error}$

Example 12.7 (cf. Example 12.5)

$$
\begin{array}{llll}
G: & S' & \rightarrow S & (0) \\
 & S & \rightarrow B \mid C & (1,2) \\
 & B & \rightarrow aB \mid b & (3,4) \\
 & C & \rightarrow aC \mid c & (5,6)
\end{array}
$$

$LR(0)(G)$	act	goto S	B	C	a	b	c
I_0	shift	I_1	I_2	I_3	I_4	I_5	I_6
I_1	accept						
I_2	red 1						
I_3	red 2						
I_4	shift		I_7	I_8	I_4	I_5	I_6
I_5	red 4						
I_6	red 6						
I_7	red 3						
I_8	red 5						
I_9	error						

(empty $= I_9$)

$LR(0)$ parsing of aac:

$$
\begin{array}{ll}
 & (aac, I_0, \varepsilon) \\
\vdash & (ac, I_0 I_4, \varepsilon) \\
\vdash & (c, I_0 I_4 I_4, \varepsilon) \\
\vdash & (\varepsilon, I_0 I_4 I_4 I_6, \varepsilon) \\
\vdash & (\varepsilon, I_0 I_4 I_4 I_8, 6) \\
(*) & \\
\vdash & (\varepsilon, I_0 I_4 I_8, 65) \\
\vdash & (\varepsilon, I_0 I_3, 655) \\
\vdash & (\varepsilon, I_0 I_1, 6552) \\
\vdash & (\varepsilon, \varepsilon, 65520)
\end{array}
$$

Check by rightmost derivation (on the board)

Remark: in the corresponding computation of NBA(G), $(*)$ is nondeterministic

Theorem 12.8 (Correctness of $LR(0)$ Parsing Automaton)

If $G \in LR(0)$, then the $LR(0)$ parsing automaton of G is deterministic, and for every $w \in \Sigma^*$ and $z \in \{0, \ldots, p\}^*$:

$$(w, I_0, \varepsilon) \vdash^* (\varepsilon, \varepsilon, z) \quad \textit{iff} \quad \overleftarrow{z} \textit{ is a rightmost analysis of } w$$

Proof.

omitted $\square$

Outline

Removing Conflicts in $LR(0)$ Parsing

In practice: often $G \notin LR(0)$

Example 12.9

$$G_{AE}: \quad E' \to E \qquad E \to E\texttt{+}T \mid T$$
$$T \to T\texttt{*}F \mid F \qquad F \to \texttt{(}E\texttt{)} \mid \texttt{a} \mid \texttt{b}$$

$LR(0)(G_{AE})$ with conflicts:

I_0 :	$[E' \to \cdot E]$	$[E \to \cdot E\texttt{+}T]$	I_1 :	$[E' \to E\cdot]$	$[E \to E \cdot \texttt{+}T]$
	$[E \to \cdot T]$	$[T \to \cdot T\texttt{*}F]$	I_2 :	$[E \to T\cdot]$	$[T \to T \cdot \texttt{*}F]$
	$[T \to \cdot F]$	$[F \to \cdot\texttt{(}E\texttt{)}]$	I_3 :	$[T \to F\cdot]$	
	$[F \to \cdot\texttt{a}]$	$[F \to \cdot\texttt{b}]$			
I_4 :	$[F \to \texttt{(} \cdot E\texttt{)}]$	$[E \to \cdot E\texttt{+}T]$	I_5 :	$[F \to \texttt{a}\cdot]$	
	$[E \to \cdot T]$	$[T \to \cdot T\texttt{*}F]$	I_6 :	$[F \to \texttt{b}\cdot]$	
	$[T \to \cdot F]$	$[F \to \cdot\texttt{(}E\texttt{)}]$	I_7 :	$[E \to E\texttt{+} \cdot T]$	$[T \to \cdot T\texttt{*}F]$
	$[F \to \cdot\texttt{a}]$	$[F \to \cdot\texttt{b}]$		$[T \to \cdot F]$	$[F \to \cdot\texttt{(}E\texttt{)}]$
				$[F \to \cdot\texttt{a}]$	$[F \to \cdot\texttt{b}]$
I_8 :	$[T \to T\texttt{*} \cdot F]$	$[F \to \cdot\texttt{(}E\texttt{)}]$	I_9 :	$[F \to \texttt{(}E \cdot \texttt{)}]$	$[E \to E \cdot \texttt{+}T]$
	$[F \to \cdot\texttt{a}]$	$[F \to \cdot\texttt{b}]$	I_{10} :	$[E \to E\texttt{+}T\cdot]$	$[T \to T \cdot \texttt{*}F]$
I_{11} :	$[T \to T\texttt{*}F\cdot]$		I_{12} :	$[F \to \texttt{(}E\texttt{)}\cdot]$	

Adding Lookahead I

Goal: resolving conflicts by considering first input symbol

Observations:

- $[A \to \beta_1 \cdot a\beta_2] \in LR(0)(\alpha\beta_1)$
  $\implies S' \Rightarrow_r^* \alpha A w \Rightarrow_r \underbrace{\alpha\beta_1}_{\text{pushdown}} \underbrace{a\beta_2 w}_{\text{next input symbol}}$

  Thus: shift only on lookahead a

- $[A \to \beta\cdot] \in LR(0)(\alpha\beta)$
  $\implies S' \Rightarrow_r^* \alpha A x w \Rightarrow_r \underbrace{\alpha\beta}_{\text{pushdown}} \underbrace{xw}_{\text{input}}$
  $\implies x \in \text{fo}(A) \subseteq \Sigma_\varepsilon$ ($x = \varepsilon$ only if $w = \varepsilon$)

  Thus: reduce with $A \to \beta$ only if lookahead $x \in \text{fo}(A)$

Adding Lookahead II

Example 12.10 (cf. Example 12.9)

$$
\begin{array}{llll}
G_{AE}: & E' & \to E & (0) \\
 & E & \to E\texttt{+}T \mid T & (1,2) \\
 & T & \to T\texttt{*}F \mid F & (3,4) \\
 & F & \to \texttt{(}E\texttt{)} \mid \texttt{a} \mid \texttt{b} & (5,6,7)
\end{array}
$$

$A \in N$	$\mathrm{fo}(A)$
E'	$\{\varepsilon\}$
E	$\{\texttt{+}, \texttt{)}, \varepsilon\}$

- $I_1 = \{[E' \to E\cdot], [E \to E \cdot \texttt{+}T]\}$:
  - accept on lookahead ε
  - shift on lookahead +
- $I_2 = \{[E \to T\cdot], [T \to T \cdot \texttt{*}F]\}$:
  - red 2 on lookahead $\texttt{+}/\texttt{)}/\varepsilon$
  - shift on lookahead *
- $I_{10} = \{[E \to E\texttt{+}T\cdot], [T \to T \cdot \texttt{*}F]\}$:
  - red 1 on lookahead $\texttt{+}/\texttt{)}/\varepsilon$
  - shift on lookahead *

$\Longrightarrow$ $SLR(1)$ parsing (Simple $LR(1)$)

The $SLR(1)$ Action Function

Definition 12.11 ($SLR(1)$ action function)

The $SLR(1)$ action function

$$\text{act} : LR(0)(G) \times \Sigma_\varepsilon \to \{\mathsf{red}\, i \mid i \in [p]\} \cup \{\mathsf{shift}, \mathsf{accept}, \mathsf{error}\}$$

is defined by

$$\text{act}(I, x) := \begin{cases} \mathsf{red}\, i & \text{if } \pi_i = A \to \alpha, [A \to \alpha \cdot] \in I \ (i \neq 0), \\ & \text{and } x \in \text{fo}(A) \\ \mathsf{shift} & \text{if } [A \to \alpha_1 \cdot x \alpha_2] \in I \text{ and } x \in \Sigma \\ \mathsf{accept} & \text{if } [S' \to S \cdot] \in I \text{ and } x = \varepsilon \\ \mathsf{error} & \text{otherwise} \end{cases}$$

Definition 12.12 ($SLR(1)$ grammar)

A grammar $G \in CFG_\Sigma$ has the $SLR(1)$ property (notation: $G \in SLR(1)$) if its $SLR(1)$ action function is well defined.

Together, act and the $LR(0)$ goto function (cf. Definition 11.9) form the $SLR(1)$ parsing table of G.

The $SLR(1)$ Parsing Table

Example 12.13 (cf. Example 12.9)

$I_0: [E' \to \cdot E]$, $[E \to \cdot E{+}T]$, $[E \to \cdot T]$, $[T \to \cdot T{*}F]$, $[T \to \cdot F]$, $[F \to \cdot (E)]$, $[F \to \cdot a]$, $[F \to \cdot b]$

$I_1: [E' \to E \cdot]$, $[E \to E \cdot {+}T]$

$I_2: [E \to T \cdot]$, $[T \to T \cdot {*}F]$

$I_3: [T \to F \cdot]$

$I_4: [F \to (\cdot E)]$, $[E \to \cdot E{+}T]$, $[E \to \cdot T]$, $[T \to \cdot T{*}F]$, $[T \to \cdot F]$, $[F \to \cdot (E)]$, $[F \to \cdot a]$, $[F \to \cdot b]$

$I_5: [F \to a \cdot]$

$I_6: [F \to b \cdot]$

$I_7: [E \to E{+} \cdot T]$, $[T \to \cdot T{*}F]$, $[T \to \cdot F]$, $[F \to \cdot (E)]$, $[F \to \cdot a]$, $[F \to \cdot b]$

$I_8: [T \to T{*} \cdot F]$, $[F \to \cdot (E)]$, $[F \to \cdot a]$, $[F \to \cdot b]$

$I_9: [F \to (E \cdot)]$, $[E \to E \cdot {+}T]$

$I_{10}: [E \to E{+}T \cdot]$, $[T \to T \cdot {*}F]$

$I_{11}: [T \to T{*}F \cdot]$

$I_{12}: [F \to (E) \cdot]$

$A \in N$	fo(A)
E'	$\{\varepsilon\}$
E	$\{+,), \varepsilon\}$
T	$\{+, *,), \varepsilon\}$
F	$\{+, *,), \varepsilon\}$

$LR(0)(G_{AE})$	act							goto								
	+	*	(	)	a	b	ε	E	T	F	+	*	(	)	a	b
I_0			shift		shift	shift		I_1	I_2	I_3			I_4		I_5	I_6
I_1	shift						accept				I_7					
I_2	red 2	shift		red 2			red 2					I_8				
I_3	red 4	red 4		red 4			red 4									
I_4			shift		shift	shift		I_9	I_2	I_3			I_4		I_5	I_6
I_5	red 6	red 6		red 6			red 6									
I_6	red 7	red 7		red 7			red 7									
I_7			shift		shift	shift			I_{10}	I_3			I_4		I_5	I_6
I_8			shift		shift	shift				I_{11}			I_4		I_5	I_6
I_9	shift			shift							I_7			I_{12}		
I_{10}	red 1	shift		red 1			red 1					I_8				
I_{11}	red 3	red 3		red 3			red 3									
I_{12}	red 5	red 5		red 5			red 5									

The $SLR(1)$ Parsing Automaton

Definition 12.14 ($SLR(1)$ parsing automaton)

The $SLR(1)$ parsing automaton is defined as in the $LR(0)$ case (see Definition 12.6), except for the transition relation:

shift: $(aw, \alpha I, z) \vdash (w, \alpha I I', z)$ if $\mathsf{act}(I, a) = \mathsf{shift}$ and $\text{goto}(I, a) = I'$

reduce$_a$: $(aw, \alpha I I_1 \ldots I_n, z) \vdash (aw, \alpha I I', zi)$ if $\mathsf{act}(I_n, a) = \mathsf{red}\; i$, $\pi_i = A \to Y_1 \ldots Y_n$, and $\text{goto}(I, A) = I'$

reduce$_\varepsilon$: $(\varepsilon, \alpha I I_1 \ldots I_n, z) \vdash (\varepsilon, \alpha I I', zi)$ if $\mathsf{act}(I_n, \varepsilon) = \mathsf{red}\; i$, $\pi_i = A \to Y_1 \ldots Y_n$, and $\text{goto}(I, A) = I'$

accept: $(\varepsilon, I_0 I, z) \vdash (\varepsilon, \varepsilon, z\, 0)$ if $\mathsf{act}(I, \varepsilon) = \mathsf{accept}$

error$_a$: $(aw, \alpha I, z) \vdash (\varepsilon, \varepsilon, z\, \mathsf{error})$ if $\mathsf{act}(I, a) = \mathsf{error}$

error$_\varepsilon$: $(\varepsilon, \alpha I, z) \vdash (\varepsilon, \varepsilon, z\, \mathsf{error})$ if $\mathsf{act}(I, \varepsilon) = \mathsf{error}$