Pushdown Automata and Parser

Reinhard Wilhelm
Universität des Saarlandes
wilhelm@cs.uni-sb.de
and
Mooly Sagiv
Tel Aviv University
sagiv@math.tau.ac.il

25. Oktober 2011
Pushdown Automata

Memory unboundedly extensible at one end,
- grows (by push),
- shrinks (by pop),
- test for emptiness.
### Example Automaton

Accepted language \( L = \{ a^i b^i \mid i \geq 0 \} \)

Context Free Grammar \( S \rightarrow aSb | \varepsilon \)

<table>
<thead>
<tr>
<th>top-stack</th>
<th>input</th>
<th>( \varepsilon )</th>
<th>$ $</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>a</td>
<td>(3)</td>
<td>(4)</td>
</tr>
<tr>
<td>1</td>
<td>b</td>
<td>(3)</td>
<td>(3)</td>
</tr>
<tr>
<td>2</td>
<td>ε</td>
<td>(2)</td>
<td>(1)</td>
</tr>
<tr>
<td>0</td>
<td>$</td>
<td>(3)</td>
<td>(3)</td>
</tr>
</tbody>
</table>

**State Definitions:**
- **state 0:** Initial state,
- **state 1:** reading a’s,
- **state 2:** reading b’s,
- **state 3:** error state,
- **state 4:** final state.
Pushdown Automata and Parser

Pushdown Automaton (PDA) Definition

A tuple $P = (V, Q, \Delta, q_0, F)$ where:

- $V$ — input-alphabet
- $Q$ — finite set of states (stack symbols)
- $q_0 \in Q$ — initial state
- $F \subseteq Q$ — final states
- $\Delta \subseteq (Q^+ \times (V \cup \{\varepsilon\})) \times Q^*$
- Alternatively: $\delta: (Q^+ \times (V \cup \{\varepsilon\})) \rightarrow 2^{Q^*}$ where $\delta$ is a partial function
The Language Accepted by a PDA

- PDA $P = (V, Q, \Delta, q_0, F)$
- For $\gamma \in Q^+$, $w \in V^*$, $(\gamma, w)$ is a configuration
- The binary relation step on configurations is defined by:
  $$(\gamma, aw) \vdash_P (\gamma', w)$$
  if
  - $\gamma \equiv \gamma_1 \gamma_2$
  - $\gamma' \equiv \gamma_1 \gamma_3$
  - $(\gamma_2, a, \gamma_3) \in \Delta$
- $\vdash_P^*$ is the reflexive transitive closure of $\vdash_P$
- The language accepted by $P$

$$L(P) = \{ w \in V^* \mid \exists q_f \in F : (q_0, w) \vdash_M^* (q_f, \varepsilon) \}$$
Deterministic Pushdown Automaton

- For every $a \in V$, $(\gamma_1, a, \gamma_2), (\gamma'_1, a, \gamma'_2) \in \Delta$ such that $\gamma'_1$ is a suffix of $\gamma_1$ implies
  - $\gamma_1 = \gamma'_1$ and
  - $\gamma_2 = \gamma'_2$

- There exist no $(\gamma_1, \varepsilon, \gamma_2), (\gamma'_1, a, \gamma'_2) \in \Delta$ such that $a \in V \cup \{\varepsilon\}$ and $\gamma'_1$ is a suffix of $\gamma_1$ or vice versa.
Theoretical Results

Theorem

For every context free grammar $G$ there exists a non-deterministic pushdown automaton $P$ such that $L(G) = L(P)$.

Proof: A PDA is given which emulates the original grammar.
Context Free Items

- A (context–free) item is a triple \((A, \alpha, \beta)\) where \(A \rightarrow \alpha\beta \in P\)
- An item \((A, \alpha, \beta)\) is denoted by \([A \rightarrow \alpha.\beta]\)
- Interpretation:
  
  "In an attempt to recognize a word for \(A\), a word for \(\alpha\) has already been recognized"

\(\alpha\) — **history** of the item \([A \rightarrow \alpha.\beta]\)

\([A \rightarrow \alpha.]\) — A complete item

\(IT_G\) — The set of items of \(G\)

\(\text{hist}([A_1 \rightarrow \alpha_1.\beta_1][A_2 \rightarrow \alpha_2.\beta_2] \ldots [A_n \rightarrow \alpha_n.\beta_n]) = \alpha_1\alpha_2 \ldots \alpha_n\)
Extended Context Free Grammar

- New start symbol \( S' \)
- Additional production \( S' \rightarrow S \)
The Item Pushdown Automaton

- A context-free-grammar $G = (V_N, V_T, P, S)$
- $P_G = (V_T, IT_G, \delta, [S' \to S], \{[S' \to S]\})$
- Control $\delta$

<table>
<thead>
<tr>
<th>top-stack</th>
<th>inp.</th>
<th>new top-stack</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>$([X \to \beta.Y\gamma])$</td>
<td>$\varepsilon$</td>
<td>$([X \to \beta.Y\gamma][Y \to .\alpha])$</td>
<td>$Y \to \alpha \in P$ “expand”</td>
</tr>
<tr>
<td>$([X \to \beta.a\gamma])$</td>
<td>$a$</td>
<td>$([X \to \beta.a.\gamma])$</td>
<td>“shift”</td>
</tr>
<tr>
<td>$([X \to \beta.Y\gamma][Y \to \alpha.])$</td>
<td>$\varepsilon$</td>
<td>$([X \to \beta.Y.\gamma])$</td>
<td>“reduce”</td>
</tr>
</tbody>
</table>

Sources of **nondeterminism**: expansion transitions; there may be several productions for $Y$. 
Example:

\[P = \{1 : S' \rightarrow S, 2 : S \rightarrow \varepsilon, 3 : S \rightarrow aSb\}\]

<table>
<thead>
<tr>
<th>top-stack</th>
<th>input</th>
<th>new top-stack</th>
<th>comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>[S' \rightarrow .S]</td>
<td>(\varepsilon)</td>
<td>[S' \rightarrow .S][S \rightarrow .] ]</td>
<td>(e_{1,2})</td>
</tr>
<tr>
<td>[S' \rightarrow .S]</td>
<td>(\varepsilon)</td>
<td>[S' \rightarrow .S][S \rightarrow .aSb] ]</td>
<td>(e_{1,3})</td>
</tr>
<tr>
<td>[S \rightarrow a.Sb]</td>
<td>(\varepsilon)</td>
<td>[S \rightarrow a.Sb][S \rightarrow .] ]</td>
<td>(e_{2,2})</td>
</tr>
<tr>
<td>[S \rightarrow a.Sb]</td>
<td>(\varepsilon)</td>
<td>[S \rightarrow a.Sb][S \rightarrow .aSb] ]</td>
<td>(e_{2,3})</td>
</tr>
<tr>
<td>[S \rightarrow .aSb]</td>
<td>(a)</td>
<td>[S \rightarrow a.Sb] ]</td>
<td>(s_1)</td>
</tr>
<tr>
<td>[S \rightarrow aS.b]</td>
<td>(b)</td>
<td>[S \rightarrow aSb.] ]</td>
<td>(s_2)</td>
</tr>
<tr>
<td>[S' \rightarrow .S][S \rightarrow .]</td>
<td>(\varepsilon)</td>
<td>[S' \rightarrow .S] ]</td>
<td>(r_1)</td>
</tr>
<tr>
<td>[S' \rightarrow .S][S \rightarrow aSb.]</td>
<td>(\varepsilon)</td>
<td>[S' \rightarrow .S] ]</td>
<td>(r_2)</td>
</tr>
<tr>
<td>[S \rightarrow a.Sb][S \rightarrow .]</td>
<td>(\varepsilon)</td>
<td>[S \rightarrow aS.b] ]</td>
<td>(r_3)</td>
</tr>
<tr>
<td>[S \rightarrow a.Sb][S \rightarrow aSb.]</td>
<td>(\varepsilon)</td>
<td>[S \rightarrow aS.b] ]</td>
<td>(r_4)</td>
</tr>
<tr>
<td>Top-Stack</td>
<td>Input</td>
<td>New Top-Stack</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>-------</td>
<td>-------------------------------</td>
<td></td>
</tr>
<tr>
<td>$[S \rightarrow .E]$</td>
<td>$\varepsilon$</td>
<td>$[S \rightarrow .E][E \rightarrow .E + T]$</td>
<td></td>
</tr>
<tr>
<td>$[S \rightarrow .E]$</td>
<td>$\varepsilon$</td>
<td>$[S \rightarrow .E][E \rightarrow .T]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .E + T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow .E + T][E \rightarrow .E + T]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .E + T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow .E + T][E \rightarrow .T]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow (.E)]$</td>
<td>$\varepsilon$</td>
<td>$[F \rightarrow (.E)][E \rightarrow .E + T]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow (.E)]$</td>
<td>$\varepsilon$</td>
<td>$[F \rightarrow (.E)][E \rightarrow .T]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow .T][T \rightarrow .T * F]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow .T][T \rightarrow .F]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .T * F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow .T * F][T \rightarrow .T * F]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .T * F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow .T * F][T \rightarrow .F]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow E + .T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow E + .T][T \rightarrow .T * F]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow E + .T]$</td>
<td>$\varepsilon$</td>
<td>$[E \rightarrow E + .T][T \rightarrow .F]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow .F][F \rightarrow .(E)]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow .F][F \rightarrow .id]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow T * .F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow T * .F][F \rightarrow .(E)]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow T * .F]$</td>
<td>$\varepsilon$</td>
<td>$[T \rightarrow T * .F][F \rightarrow .id]$</td>
<td></td>
</tr>
<tr>
<td>Top-Stack</td>
<td>Input</td>
<td>New Top-Stack</td>
<td></td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>--------------</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow .(E)]$</td>
<td>(</td>
<td>$[F \rightarrow (E)]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow .id]$</td>
<td>id</td>
<td>$[F \rightarrow id]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow (E.)]$</td>
<td>)</td>
<td>$[E \rightarrow (E.)]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow E + T]$</td>
<td>+</td>
<td>$[E \rightarrow E + .T]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow T * F]$</td>
<td>*</td>
<td>$[T \rightarrow T * .F]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .F][F \rightarrow id.]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow F.]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow T * .F][F \rightarrow id.]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow T * F.]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .F][F \rightarrow (E.)]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow F.]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow T * .F][F \rightarrow (E.)]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow T * F.]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .T * F][T \rightarrow F.]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow T * .F]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .T][T \rightarrow F.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow T.]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow E + .T][T \rightarrow F.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow E + T.]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow E + .T][T \rightarrow T * F.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow E + T.]$</td>
<td></td>
</tr>
<tr>
<td>$[T \rightarrow .T * F][T \rightarrow T * F.]$</td>
<td>$\epsilon$</td>
<td>$[T \rightarrow T * .F]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .T][T \rightarrow T * F.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow T.]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow (.E)][E \rightarrow T.]$</td>
<td>$\epsilon$</td>
<td>$[F \rightarrow (E.)]$</td>
<td></td>
</tr>
<tr>
<td>$[F \rightarrow (.E)][E \rightarrow E + T.]$</td>
<td>$\epsilon$</td>
<td>$[F \rightarrow (E.)]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .E + T][E \rightarrow T.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow E. + T]$</td>
<td></td>
</tr>
<tr>
<td>$[E \rightarrow .E + T][E \rightarrow E + T.]$</td>
<td>$\epsilon$</td>
<td>$[E \rightarrow E. + T]$</td>
<td></td>
</tr>
<tr>
<td>$[S \rightarrow .E][E \rightarrow T.]$</td>
<td>$\epsilon$</td>
<td>$[S \rightarrow E.]$</td>
<td></td>
</tr>
<tr>
<td>$[S \rightarrow .E][E \rightarrow E + T.]$</td>
<td>$\epsilon$</td>
<td>$[S \rightarrow E.]$</td>
<td></td>
</tr>
</tbody>
</table>
### Accepting $id + id \ast id$

<table>
<thead>
<tr>
<th>Stack</th>
<th>Remaining Input</th>
</tr>
</thead>
<tbody>
<tr>
<td>$S \rightarrow .E$</td>
<td>id + id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T]$</td>
<td>id + id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T]$</td>
<td>id + id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T][T \rightarrow .F]$</td>
<td>id + id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T][T \rightarrow .F][F \rightarrow .id]$</td>
<td>id + id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T][T \rightarrow .F][F \rightarrow .id.]$</td>
<td>+id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T][T \rightarrow F.]$</td>
<td>+id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + T][E \rightarrow .T]$</td>
<td>+id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T]$</td>
<td>id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow .T \ast F]$</td>
<td>id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow .T \ast F][T \rightarrow .F]$</td>
<td>id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow .T \ast F][T \rightarrow .F][F \rightarrow .id]$</td>
<td>id * id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow .T \ast F][T \rightarrow .F][F \rightarrow .id.]$</td>
<td>*id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow .T \ast F][T \rightarrow .F.]$</td>
<td>*id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow T \ast F]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow T \ast .F]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow T \ast .F.[F \rightarrow .id]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow T \ast .F.[F \rightarrow .id.]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T][T \rightarrow T \ast F.]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E[E \rightarrow .E + .T]$</td>
<td>id</td>
</tr>
<tr>
<td>$S \rightarrow .E$</td>
<td>id</td>
</tr>
</tbody>
</table>
The Simulation Lemma

Lemma
If \([S' \rightarrow .S], uv) \vdash^*_{PG} (\rho, v) \) then \(hist(\rho) \xrightarrow[*]{G} u\)

Corollary: \(L(P_G) \subseteq L(G)\)
The Other Direction

Lemma
Let $A \in V_N$ and $w \in V_T^*$. If $A \xrightarrow{G}^* w$, there exists $A \rightarrow \alpha \in P$ such that for all $\rho \in I T_G^*$ and $v \in V_T^*$

$$(\rho[A \rightarrow .\alpha], wv) \vdash_{P_G}^* (\rho[A \rightarrow \alpha.], v)$$

Corollary: $L(P_G) \supseteq L(G)$
Automaton with Output

A tuple $P = (V, Q, \Delta, O, q_0, F)$ where:

- $V$ — input-alphabet
- $O$ — output-alphabet
- $Q$ — finite set of states
- $q_0 \in Q$ — initial state
- $F \subseteq Q$ — final states
- $\Delta \subseteq (Q^+ \times (V \cup \{\varepsilon\})) \times Q^* \times (O \cup \{\varepsilon\})$

Alternatively:
$\delta: (Q^+ \times (V \cup \{\varepsilon\})) \rightarrow 2^{Q^*} \times (O \cup \{\varepsilon\})$

where $\delta$ is a partial function
Left/Predictive/Top-Down Parser

\[ P^I_G = (V_T, IT_G, P, \delta_I, [S' \rightarrow .S], \{[S' \rightarrow S.]\}) \text{ where} \]
\[ \delta_I([X \rightarrow \beta.Y\gamma], \varepsilon) = \{([X \rightarrow \beta.Y\gamma][Y \rightarrow .\alpha], Y \rightarrow \alpha) \mid Y \rightarrow \alpha \in P\} \]

Configuration: \( IT_T^+ \times V_T^* \times P^* \)

Step: \((\rho[X \rightarrow \beta.Y\gamma], w, o) \vdash_{P_G^I} (\rho([X \rightarrow \beta.Y\gamma][Y \rightarrow .\alpha], w, o(Y \rightarrow \alpha)) \)
Right/Bottom-Up Parser

\[ P_G^r = (V_T, IT_G, P, \delta_r, [S' \rightarrow .S], \{[S' \rightarrow S.]\}) \text{ where} \]
\[ \delta_r([X \rightarrow \beta.Y\gamma][Y \rightarrow \alpha.], \varepsilon) = \{([X \rightarrow \beta Y.\gamma], Y \rightarrow \alpha)\} \]

Configuration: \( IT_G^+ \times V_T^* \times P^* \)

Step: \((\rho[X \rightarrow \beta.Y\gamma][Y \rightarrow \alpha.], w, o) \vdash_{P_G} (\rho([X \rightarrow \beta Y.\gamma], w, o(Y \rightarrow \alpha)))\)
Deterministic Parsers

**LL(k):** Deterministic left parsers
- Read the input from left to right
- Find leftmost derivation
- Take decisions as early as possible, i.e. on expansion
- Use $k$ symbols look ahead to decide about expansions

**LR(k):** Deterministic right parsers
- Read the input from left to right
- Find rightmost derivation in reverse order
- Delay decisions as long as possible, i.e. until reduction
- Use $k$ tokens look ahead to
  - decide whether to shift or reduce (in “shift-reduce-conflicts”)
  - decide by which rule to reduce (in “reduce-reduce-conflicts”)

### Example: Predictive Parser

\[ S' \rightarrow S, \ S \rightarrow aSb|\varepsilon \]

1-symbol look ahead for expansions

<table>
<thead>
<tr>
<th>top-stack</th>
<th>LA</th>
<th>new top-stack</th>
<th>used production</th>
</tr>
</thead>
</table>
| ([S' \rightarrow .S]) | $  | ([S \rightarrow .]  
|              |    | ([S' \rightarrow .S]) 
|              |    | ([S \rightarrow .aSb])  
|              |    | ([S' \rightarrow .S]) | \( S \rightarrow \varepsilon \) |
| ([S \rightarrow a.Sb]) | a  | ([S \rightarrow .]  
|              |    | ([S \rightarrow .aSb])  
|              |    | ([S \rightarrow a.Sb]) | \( S \rightarrow aSb \) |
| ([S \rightarrow a.Sb]) | b  | ([S \rightarrow .]  
|              |    | ([S \rightarrow a.Sb])  
|              |    | ([S \rightarrow .aSb])  
|              |    | ([S \rightarrow a.Sb]) | \( S \rightarrow \varepsilon \) |
| ([S \rightarrow a.Sb]) | a  | ([S \rightarrow .]  
|              |    | ([S \rightarrow .aSb])  
|              |    | ([S \rightarrow a.Sb])  
|              |    | ([S \rightarrow a.Sb]) | \( S \rightarrow aSb \) |
shift rules

<table>
<thead>
<tr>
<th>top-stack</th>
<th>Input</th>
<th>new top-stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>([S → .aSb])</td>
<td>a</td>
<td>([S → a.Sb])</td>
</tr>
<tr>
<td>([S → aS.b])</td>
<td>b</td>
<td>([S → aSb.])</td>
</tr>
</tbody>
</table>

reduction rules

<table>
<thead>
<tr>
<th>top-stack</th>
<th>Input</th>
<th>new top-stack</th>
</tr>
</thead>
<tbody>
<tr>
<td>([S → .] (S’ → .S)]</td>
<td>ε</td>
<td>([S’ → S.])</td>
</tr>
<tr>
<td>([S → aSb.] (S’ → .S)]</td>
<td>ε</td>
<td>([S’ → S.])</td>
</tr>
<tr>
<td>([S → .] (S → a.Sb)]</td>
<td>ε</td>
<td>([S → aS.b])</td>
</tr>
<tr>
<td>([S → aSb.] (S → a.Sb)]</td>
<td>ε</td>
<td>([S → aS.b])</td>
</tr>
</tbody>
</table>