As explained before, concerning SLR(1), it is possible to consult the next set to determine, in the reduction states, for which symbols it would be possible to perform reductions.

- The key concept is the *look-ahead symbol*. Using this symbol will allow us to bear in mind, together with the input symbol that is being studied, the next one that appears in the input.
- Because this symbol will also be used in a later step, it receives the name of *look-ahead symbol*.

In SLR(1), when we reduced a rule $R$ in a state $s$ only for the elements in the next set, that action is like considering all the look-ahead symbols for $R$ in all the possible states.

We have seen an example grammar which is not SLR(1):

1. $S \rightarrow A$
2. $S \rightarrow xb$
3. $A \rightarrow aAb$
4. $A \rightarrow B$
5. $B \rightarrow x$

- this grammar generates the following language:

\[ \{ xb \} \cup \{ a^nxb^n \mid n \geq 0 \} \]

Remember the augmented grammar:

1. $S' \rightarrow S\$
2. $S \rightarrow A$
3. $S \rightarrow xb$
4. $A \rightarrow aAb$
5. $A \rightarrow B$
6. $B \rightarrow x$

**Deterministic finite automata with the transitions**
• And this was the analysis table for the SLR(1) grammar, with the conflict

<table>
<thead>
<tr>
<th></th>
<th>Σ_a</th>
<th>Σ_w</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s3</td>
<td>s5</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td>r1</td>
</tr>
<tr>
<td>3</td>
<td>s3</td>
<td>s8</td>
</tr>
<tr>
<td>4</td>
<td>r4</td>
<td>r4</td>
</tr>
<tr>
<td>5</td>
<td>r4</td>
<td>r5</td>
</tr>
<tr>
<td>6</td>
<td>r2</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r5</td>
<td>r5</td>
</tr>
<tr>
<td>9</td>
<td>r3</td>
<td>r3</td>
</tr>
</tbody>
</table>

(0) S'→SS
(1) S→A
(2) S→xb
(3) A→aAb
(4) A→B
(5) B→x

• The last example seen was an example of an ambiguous grammar which was not SLR(1).

  (1) E→E+E
  (2) E→E*E
  (3) E→i

• Remember the transition diagram in the SLR(1) automata
And the SLR(1) matrix showing the conflicts

<table>
<thead>
<tr>
<th></th>
<th>*</th>
<th>+</th>
<th>i</th>
<th>$</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>s1</td>
<td>2</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>r3</td>
<td>r3</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>s4</td>
<td>s3</td>
<td>acc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td></td>
<td>s1</td>
<td>6</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td></td>
<td>s1</td>
<td>5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>r2</td>
<td>r2</td>
<td>r1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>r1</td>
<td>r1</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The grammar is not SLR(1)

In the analysis, at each step, we shall take into consideration:
- The current symbol that is being treated
- The k symbols that can follow it (k look-ahead symbols).

Each state in the automata will have several copies, as many as possible sequences of k symbols that can follow the current state.

The example used to illustrate LR(1) and LALR(1) grammars will be the following:

\[ \{ xb, a^n xb^n \mid n \geq 0 \} \]

This is not the simplest grammar we can build for this language, but it is a good example, because:
- It is not SLR(1)
- It is LR(1)
- It is LALR, but the automata obtained with this method will be different than the one obtained for an LR(1) parser
- We shall see, in this introductory example, the initial state s0 and the transition from s0 to s5.
LR(1) Analysis
Look-ahead symbols in an LR(1) analyser: introductory example

- Considering the initial state,
  - As in LR(0) and SLR(1), it was calculated as the closure of the following configuration:
    \[ S' \rightarrow \bullet S $ \]

- \( S \) is the old axiom, the non-terminal symbol to which we should derive the complete derivation tree after we finish the analysis. Therefore, the only look-ahead symbol that we should expect at this position is the end-of-program symbol (\( $ \))

- Therefore, that configuration in the initial state will have the set of look-ahead symbols
  \[ \{ $ \} \]

- The next diagram shows this example.

- Furthermore, the initial state will contain the closure of the configuration \( S' \rightarrow \bullet S \)

- For calculating closures, we need to calculate the look-ahead symbols for each of the new configurations added:
  - This case will be solved with the help of the \textit{first} and \textit{next} sets.
  - We shall use the parts of the right-hand sides of the non-terminal symbol that follows the dot.
  - For instance, in the initial state \( s_0 \)
    \[
    s_0 = \{ S' \rightarrow \bullet S \{ $ \},
    S \rightarrow \bullet A \{ $ \},
    S \rightarrow \bullet xb \{ $ \},
    A \rightarrow \bullet aAb \{ $ \},
    A \rightarrow \bullet B \{ $ \} \}
    \]

- the two new configurations added can have, as look-ahead, anything that could appear after the \( S \) in \( S' \rightarrow \bullet S \)

- Because the \( S \) was at the end of the rule, and the look-ahead symbol for that rule was \( $ \), the look-ahead symbols for the new configurations are the same.

- We can continue the closure with the rules for \( A \):
  \[
  s_o = \{ S' \rightarrow \bullet S \{ $ \},
  S \rightarrow \bullet A \{ $ \},
  S \rightarrow \bullet xb \{ $ \},
  A \rightarrow \bullet aAb \{ $ \},
  A \rightarrow \bullet B \{ $ \} \}
  \]

- As before, because \( A \) was at the end of the rule \( S \rightarrow \bullet A \), after the \( A \) we may find anything that we could find after the \( S \) (the look-ahead symbols for that configuration): \( \{ $ \} \)

- The same argumentation can be followed to show that the look-ahead symbols for the last configuration to be added to this state is also \( \{ $ \} \):
  \[
  s_o = \{ S' \rightarrow \bullet S \{ $ \},
  S \rightarrow \bullet A \{ $ \},
  S \rightarrow \bullet xb \{ $ \},
  A \rightarrow \bullet aAb \{ $ \},
  A \rightarrow \bullet B \{ $ \},
  B \rightarrow \bullet x \} \]
LR(1) Analysis
Look-ahead symbols in an LR(1) analyser: introductory example

• Deterministic Finite Automata with transitions

\[
\begin{align*}
S' &\rightarrow S(\$) \\
S &\rightarrow A(\$) \\
S &\rightarrow xB(\$) \\
A &\rightarrow aAb(\$) \\
A &\rightarrow B(\$) \\
B &\rightarrow x(\$)
\end{align*}
\]

• Let us continue the example with a transition between two states:

• Again, the closures will be solved with the help of the first and next sets.

• For instance, from the initial state \(s_0\)

\[
\begin{align*}
s_0 &= \\
&\{ S' \rightarrow S(\$), \\
&\quad S \rightarrow A(\$), \\
&\quad S \rightarrow xB(\$), \\
&\quad A \rightarrow aAb(\$), \\
&\quad A \rightarrow B(\$), \\
&\quad B \rightarrow x(\$) \}
\end{align*}
\]

• When we calculate \(s_5\) as the resulting state of \(go_to(s_0, a)\), the underlined configuration can be shifted to \(A \rightarrow aAb(\$)\)

• In summary, with respect to the state \(s_5\), because of the symbol A, \(A \rightarrow aAb(\$)\) and \(A \rightarrow B(\$)\) both have the look-ahead symbol \(b\):

\[
\begin{align*}
A &\rightarrow aAb(\$) \\
A &\rightarrow B(b) \\
B &\rightarrow x(b)
\end{align*}
\]

• The presence of \(A \rightarrow B(b)\) in \(s_5\) forces us to extend the closure with all the right-hand sides of B.

• The situation in this case is different, because there is no symbol after the B in the rule \((A \rightarrow B(\ast)\{b\})\)

• In this case, as in \(s_0\) before, the closure does not modify the set of look-ahead symbols, and the element which will be look-ahead symbol of the configuration B \(\rightarrow x\) will be \(\{b\}\).
**LR(1) Analysis**

Look-ahead symbols in an LR(1) analyser: introductory example

- So we can conclude that

\[
s_5 = \\
\{ A \rightarrow a \cdot Ab \{ \$ \} , \\
A \rightarrow aAb \{ b \} , \\
A \rightarrow B \{ b \} , \\
B \rightarrow x \{ b \} \}
\]

- as we can see in the following diagram:

---

**LR(1) Analysis**

Look-ahead symbols in an LR(1) analyser: formalisation

- Formally,

  - to calculate the look-ahead symbols for the states transitions diagram,
    - The initial configuration in the initial state have the following look-ahead symbols:
      \[
      \{ \$ \}
      \]
    - For calculating \( s_j = \text{closure}(s_i) \) for the elements \( s_i' \) of the form
      \[
      P \rightarrow \alpha \cdot N \{ \sigma_1, \ldots, \sigma_n \} , \quad N \in \Sigma_N
      \]
      - we need to add next(N), calculated using the previous rule, as look-ahead symbol for all the new configurations
        \[
        N \rightarrow \cdot \ldots 
        \]
      - This next(N) set will be
        - \( \text{first}(\beta \{ \sigma_1, \ldots, \sigma_n \}) \)
        - \( \text{next}(P) \) if \( \beta \{ \sigma_1, \ldots, \sigma_n \} \rightarrow \star \lambda \)

  Remember, from the calculation of the next set, that if \( \beta \) can derive the empty word, we have to add not only first(\( \beta \)) to the set next(N), but also first(\( \sigma \)) where \( \sigma \) is the symbol that follows \( \beta \) in the right-hand side of the rule, and next(P), if \( \beta \) is the last symbol in the right-hand side!!!!
Introductory exercise

• Build the LR(1) analysis table for the following grammar that generates language $L = \{ xb, a^nxb^n | n \geq 0 \}$

  1. $S \rightarrow A$
  2. $S \rightarrow xb$
  3. $A \rightarrow aAb$
  4. $A \rightarrow B$
  5. $B \rightarrow x$

• Use them to analyse the following two strings:
  - $aaxbb$
  - $ax$

  The first step is to obtain the augmented grammar:

  0. $S' \rightarrow S$
  1. $S \rightarrow A$
  2. $S \rightarrow xb$
  3. $A \rightarrow aAb$
  4. $A \rightarrow B$
  5. $B \rightarrow x$

Deterministic Finite Automata with transitions

$^1 \text{first}(\$) = \{\}$

$^1 \text{first}(\$) = \{\}$
LR(1) Analysis

Constructing LR(1) analysis tables

• Deterministic Finite Automata with transitions

1first($)={$}
2first(b$)={b}
3first(b)={b}

S'→S($)  S→A($)  S→xb($)
A→aAb($)  A→xb(b)
A→B($)  B→x($)

S→A*($)  S→B*($)  A→B*($)  A→B*($)

S'→S*($)  S→A*($)  S→B*($)  A→aAb($)  A→xb(b)  A→B($)  B→x($)
LR(1) Analysis

Constructing LR(1) analysis tables

• Deterministic Finite Automata with transitions

1\text{first}($) = \{$$
2\text{first}(b$) = \{b
3\text{first}(b) = \{b
4\text{first}(bb) = \{b

• Deterministic Finite Automata with transitions

1\text{first}($) = \{$$
2\text{first}(b$) = \{b
3\text{first}(b) = \{b
4\text{first}(bb) = \{b

LR(1) Analysis

Constructing LR(1) analysis tables

- Deterministic Finite Automata with transitions

\[ S' \rightarrow S \{ \} \]

\[ S' \rightarrow A \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ S \rightarrow aA \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]

\[ S \rightarrow A \{ \} \]

\[ A \rightarrow aA \{ \} \]

\[ A \rightarrow B \{ \} \]

\[ B \rightarrow x \{ \} \]
LR(1) Analysis

Constructing LR(1) analysis tables

- Deterministic Finite Automata with transitions

\[ s_0 \rightarrow S' \rightarrow S \rightarrow A \rightarrow B \rightarrow x \rightarrow S \rightarrow A \rightarrow a \rightarrow A \rightarrow aA \rightarrow aab \rightarrow aabb \rightarrow \cdots \]

1. **Shifts in the table:**
   - It is the same as in LR(0)
   - They can be obtained by following the transitions in the table.
   - If the automata can go from \( s_i \) to \( s_j \) by means of symbol \( X \), then we shall add the following action:
     \[
     \text{Syntactic_table}[i, X] = \begin{cases} 
     s_j & \text{if } X \in \Sigma \\
     s_j & \text{if } X \in \Sigma 
     \end{cases}
     \]

2. **Reductions in the table:**
   - In the cells for the states which contain reduction configurations, of the form \( A \rightarrow \gamma \sigma_1 \cdots \sigma_n \), we have to add the reduction of the rule \( A \rightarrow \gamma \) only in the columns for their look-ahead non-terminal symbols, i.e., \( \sigma_1 \ldots \sigma_n \).
   - Therefore, this step is different to that in LR(0)
   
   \[ 1\text{first}($) = \{\}$ \]
   \[ 2\text{first}(b$) = \{b\} \]
   \[ 3\text{first}(b) = \{b\} \]
   \[ 4\text{first}(bb) = \{b\} \]

3. **Acceptation:**
   - It is the same as in LR(0) analysers
   - If a state \( s_i \) has a transition with the terminal symbol \$ to the final state with the configuration \( \text{axioma}' \rightarrow \text{axioma}$ \), we have to add the accept action to \( \text{Syntactic_table}[i, \]$ \).
   - Is it possible to find alternative techniques for acceptation in LR(1) parsers.

4. **Error:**
   - It is the same as in LR(0)
   - All the empty cells have associated the error action.
## LR(1) Analysis

### Constructing LR(1) analysis tables: method

#### Analysis table

The following is an example of analysis with two strings:

- **aaxbb**
- **ax**

<table>
<thead>
<tr>
<th>Σᵣ</th>
<th>Σₓ</th>
<th>S'</th>
<th>S</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>s4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td>r5</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>s7</td>
<td>s9</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Go-to</th>
</tr>
</thead>
</table>

### LR(1) Analysis: exercise

#### {xb, aⁿxbⁿ | n≥0}

<table>
<thead>
<tr>
<th>Σᵣ</th>
<th>Σₓ</th>
<th>S'</th>
<th>S</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>s4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td>r5</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>s7</td>
<td>s9</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Go-to</th>
</tr>
</thead>
</table>

- (0) S'→S$ $
- (1) S→A$
- (2) S→xb
- (3) A→aAb
- (4) A→B
- (5) B→x

#### {xb, aⁿxbⁿ | n≥0}

<table>
<thead>
<tr>
<th>Σᵣ</th>
<th>Σₓ</th>
<th>S'</th>
<th>S</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>s4</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td>r5</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td>6</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>s7</td>
<td>s9</td>
<td>11</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r5</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Action</th>
<th>Go-to</th>
</tr>
</thead>
</table>

- (0) S'→S$ $
- (1) S→A$
- (2) S→xb
- (3) A→aAb
- (4) A→B
- (5) B→x
### LR(1) Analysis: exercise

#### Grammar

$\{xb, a^i x b^n | n \geq 0\}$

#### LR(1) Automaton

<table>
<thead>
<tr>
<th>State</th>
<th>Symbol</th>
<th>Action</th>
<th>Go-to</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>r1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td>r5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td>s9</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>r4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td>r3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

---

### LR(1) Analysis: exercise

#### Grammar

$\{xb, a^i x b^n | n \geq 0\}$

#### LR(1) Automaton

<table>
<thead>
<tr>
<th>State</th>
<th>Symbol</th>
<th>Action</th>
<th>Go-to</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>a</td>
<td>b</td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>r1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td>r5</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td>s9</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>r4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r5</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td>r3</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
LR(1) Analysis: exercise

\[
\begin{align*}
(0) & \quad S' \rightarrow S \\
(1) & \quad S \rightarrow A \\
(2) & \quad S \rightarrow xb \\
(3) & \quad A \rightarrow aAb \\
(4) & \quad A \rightarrow B \\
(5) & \quad B \rightarrow x \\
\end{align*}
\]

\[
\begin{array}{cccccc}
(0) & (1) & (2) & (3) & (4) & (5) \\
S' & S & A & B & x & a \\
\end{array}
\]

\[
\begin{array}{cccccc}
(0) & (1) & (2) & (3) & (4) & (5) \\
S' & S & A & B & x & a \\
\end{array}
\]

Go-to

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
E & a & b & x & S & A \\
\end{array}
\]

Action

\[
\begin{array}{cccccc}
0 & 1 & 2 & 3 & 4 & 5 \\
E & a & b & x & S & A \\
\end{array}
\]
### LR(1) Analysis: exercise

**Grammar:**

\[
\{xb, a^nxb^n \mid n \geq 0\}
\]

**LR(1) Parse Table:**

<table>
<thead>
<tr>
<th>State</th>
<th>Symbol</th>
<th>a</th>
<th>x</th>
<th>$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>s5</td>
<td>s4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>acc</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>r1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>s13</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>s7</td>
<td>s9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>s10</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>s7</td>
<td>s9</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>r4</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>r5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>s12</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>r3</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>r2</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Rules:**

1. \( S' \rightarrow S $
2. \( S \rightarrow A \)
3. \( S \rightarrow xb \)
4. \( A \rightarrow aAb \)
5. \( A \rightarrow B \)
6. \( B \rightarrow x \)

**Actions:**

- 0: \( S' \rightarrow S $
- 1: \( S \rightarrow A \)
- 2: \( S \rightarrow xb \)
- 3: \( A \rightarrow aAb \)
- 4: \( A \rightarrow B \)
- 5: \( B \rightarrow x \)
LR(1) Analysis: exercise

\[
\begin{align*}
\Sigma_I & \quad \Sigma_n \\
0 & \quad a \quad b \quad x \quad S' \quad S \quad A \quad B \\
1 & \quad acc \\
2 & \quad r1 \\
3 & \quad r4 \\
4 & \quad s13 \quad r5 \\
5 & \quad a \quad b \quad x \quad 8 \\
6 & \quad s10 \\
7 & \quad s9 \\
8 & \quad r4 \\
9 & \quad r5 \\
10 & \quad s12 \\
11 & \quad r3 \\
12 & \quad r3 \\
13 & \quad r2
\end{align*}
\]

\(\{xb, a^nxb^n \mid n \geq 0\}\)

LR(1) Analysis

Evaluation

- Power:
  - It can be shown that LR(1) is the most powerful analysis algorithm amongst those that analyse the string from left to right using just 1 look-ahead symbol.

- Efficiency:
  - As can be observed from the examples, there are considerably more states in LR(1) analysers than in LR(0) analysers.

LR(k) Analysis

Generalisation to k look-ahead symbols (k > 1)

- This same technique can be extended to consider any number of look-ahead symbols.
- This generalisation is out of the scope of this course.