

Exercise 1 (Characterisation of $LL(1)$):

(3 Points)

In the lecture two characterizations of $LL(1)$ have been given:

- $G \in LL(1)$ iff for all leftmost derivations of the form

$$S \Rightarrow_l^* wA\alpha \begin{cases} \Rightarrow_l w\beta\alpha \\ \Rightarrow_l w\gamma\alpha \end{cases}$$

such that $\beta \neq \gamma$, it follows that $\text{fi}(\beta\alpha) \cap \text{fi}(\gamma\alpha) = \emptyset$.

- $G \in LL(1)$ iff for all pairs of rules $A \rightarrow \beta \mid \gamma \in P$ (where $\beta \neq \gamma$):

$$\text{la}(A \rightarrow \beta) \cap \text{la}(A \rightarrow \gamma) = \emptyset$$

- Lift the second definition to $LL(k)$ for $k \in \mathbb{N}^+$. (The first definition was given for $k \in \mathbb{N}^+$ in the lecture.)
- Show that the definitions are not equivalent by showing that the following grammar is in $LL(2)$ according to the first definition but not according to the second definition (also referred to as *strong* $LL(2)$ property).

$$\begin{aligned} S &\rightarrow aAab \mid bAbb \\ A &\rightarrow a \mid \varepsilon \end{aligned}$$

- Explain (in a few words) why the definitions are not equivalent.

Lösung: _____

a)

Second definition:

$G \in LL(k)$ iff for all pairs of rules $A \rightarrow \beta \mid \gamma \in P$ (where $\beta \neq \gamma$):

$$\text{la}_k(A \rightarrow \beta) \cap \text{la}_k(A \rightarrow \gamma) = \emptyset$$

where $\text{la}_k(A \rightarrow \beta) = \text{first}_k(\beta \text{ follow}_k(A))$.

b)

$$\begin{aligned}\text{first}_2(S) &= \{aa, ba, bb\} \\ \text{first}_2(A) &= \{a, \epsilon\} \\ \text{follow}_2(S) &= \{\epsilon\} \\ \text{follow}_2(A) &= \{ab, bb\}\end{aligned}$$

$$\begin{aligned}\text{la}_2(S \rightarrow aAab) &= \{aa\} \\ \text{la}_2(S \rightarrow bAbb) &= \{ba, bb\} \\ \text{la}_2(A \rightarrow a) &= \{aa, ab\} \\ \text{la}_2(A \rightarrow \epsilon) &= \{ab, bb\}\end{aligned}$$

$\text{la}_2(A \rightarrow a) \cap \text{la}_2(A \rightarrow \epsilon) = \{ab\} \neq \emptyset$ and therefore the grammar is not in $LL(2)$ according to the second definition.

Check all leftmost derivations as in the definition:

$$\begin{aligned}S &\Rightarrow_l aAab \\ S &\Rightarrow_l bAbb\end{aligned}$$

$$\text{first}_2(aAab) \cap \text{first}_2(bAbb) = \{aa\} \cap \{ba, bb\} = \emptyset$$

$$\begin{aligned}S &\Rightarrow_l aAab \Rightarrow_l aaab \\ S &\Rightarrow_l aAab \Rightarrow_l aab\end{aligned}$$

$$\text{first}_2(aab) \cap \text{first}_2(ab) = \{aa\} \cap \{ab\} = \emptyset$$

$$\begin{aligned}S &\Rightarrow_l bAbb \Rightarrow_l babb \\ S &\Rightarrow_l bAbb \Rightarrow_l bbb\end{aligned}$$

$$\text{first}_2(bbb) \cap \text{first}_2(bb) = \{ab\} \cap \{bb\} = \emptyset$$

Thus, the grammar is in $LL(2)$ according to the first definition.

c)

The second definition does not take into account the word read so far, i.e.

$$\text{la}_2(A \rightarrow a) = \text{first}_2(abb) \cup \text{first}_2(aab) = \{aa, ab\} \text{ and}$$

$$\text{la}_2(A \rightarrow \epsilon) = \text{first}_2(ab) \cup \text{first}_2(bb) = \{ab, bb\}$$

where $\text{first}_2(aab)$ and $\text{first}_2(ab)$ only occur when a precedes A and
 $\text{first}_2(abb)$ and $\text{first}_2(bb)$ only occur when b precedes A .

Exercise 2 (Find an equivalent $LL(1)$ grammar):

(3 Points)

Consider the following grammar G :

$$\begin{aligned}S &\rightarrow (L) \mid a \\ L &\rightarrow L, S \mid L, SS \mid S \mid SS\end{aligned}$$

- a) Show that G is not an $LL(1)$ grammar.
- b) Transform G into an equivalent grammar satisfying the $LL(1)$ property.
- c) Prove that G has the $LL(1)$ property.

Lösung: _____

- a) First of $\text{La}(L \rightarrow L, S) \cap \text{La}(L \rightarrow L, SS)$ is not empty.
- b) First remove left recursion:

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow SL' \mid SSL' \\ L' &\rightarrow ,SL' \mid ,SSL' \mid \epsilon \end{aligned}$$

Then apply left factorisation:

$$\begin{aligned} S &\rightarrow (L) \mid a \\ L &\rightarrow SL'' \\ L' &\rightarrow ,SL'' \mid \epsilon \\ L'' &\rightarrow L' \mid SL' \end{aligned}$$

Rule	La
$S \rightarrow (L)$	$\{()\}$
$S \rightarrow a$	$\{a\}$
c) $L' \rightarrow ,SL''$	$\{ , ,)\}$
$L' \rightarrow \epsilon$	$\{ \}$
$L'' \rightarrow L'$	$\{ , ,)\}$
$L'' \rightarrow SL'$	$\{ ()\}$

For every nonterminal it holds that the intersection of their lookahead sets is empty. Therefore the grammar is $LL(1)$.

Exercise 3 ($LL(1)$ grammars are never ambiguous):

(1 Points)

Show that for every context-free grammar G the following holds:

$$G \text{ ambiguous} \Rightarrow G \notin LL(1)$$

Lösung: _____

$G = (N, \Sigma, P, S)$ ambiguous

$\Rightarrow \exists \omega \in \Sigma^*$ with at least two *left*-derivations $S \Rightarrow_i^* vA\alpha \Rightarrow_i v\beta\alpha \Rightarrow_i^* vv' = \omega$
 $\Rightarrow_i v\gamma\alpha \Rightarrow_i^* vv' = \omega$
 We assume this splitting to be the first difference appearing in the derivations.

As obviously $fi(v') = fi(v')$ and therefore $fi(\beta\alpha) \cap fi(\gamma\alpha) \neq \emptyset$

$\Rightarrow G \notin LL(1)$

Exercise 4 (Deterministic Top-Down Automaton):

(3 Points)

Consider the grammar $G = (N, \Sigma, P, start)$ covering some boolean expressions:

- $N := \{start, guard, rel\}$
- $\Sigma := \{AND, OR, ID, EQ, LEQ\}$
- $$\begin{array}{ll}
 start & \rightarrow guard \\
 guard & \rightarrow rel \mid guard \text{ AND } guard \mid guard \text{ OR } guard \\
 rel & \rightarrow ID \text{ EQ } ID \mid ID \text{ LEQ } ID
 \end{array}$$

- a) Construct $NTA(G)$. (Either give a transition table or depict the automaton and specify what the edge labelling means. Do not forget to give a numbering to the grammar rules.)
- b) Provide a run of $NTA(G)$ on the input $ID \text{ EQ } ID \text{ AND } ID \text{ LEQ } ID$.
- c) Construct an equivalent grammar G' with $G' \in LL(1)$.
- d) Specify the deterministic top-down parsing automaton of $DTA(G')$. (Again, either give a transition table as in the lecture or depict the automaton and specify what the edge labelling means. As before, do not forget to give a numbering to the grammar rules of G' .)
- e) Provide a run of $DTA(G')$ on the input $ID \text{ EQ } ID \text{ AND } ID \text{ LEQ } ID$.

Lösung:

a)

1. $start \rightarrow guard$
2. $guard \rightarrow rel$
3. $guard \rightarrow guard \text{ AND } guard$
4. $guard \rightarrow guard \text{ OR } guard$
5. $rel \rightarrow ID \text{ EQ } ID$
6. $rel \rightarrow ID \text{ LEQ } ID$

The non-deterministic push down automaton $P = (Q, s_0, st, \delta)$.

- $\Sigma = \{ID, EQ, LEQ, AND, OR\}$.
- $N = \{start, guard, rel\}$.
- $X = \Sigma \cup N$.
- $P = \{1, 2, 3, 4, 5, 6\}$.
- $Q = \{s_0\}$

- $st = X^*$.
- The transition relation is defined as:
 - $(s_0, \text{start}, \epsilon) \rightarrow (s_0, \text{guard}, 1)$
 - $(s_0, \text{guard}, \epsilon) \rightarrow (s_0, \text{rel}, 2)$
 - $(s_0, \text{guard}, \epsilon) \rightarrow (s_0, \text{guard AND guard}, 3)$
 - $(s_0, \text{guard}, \epsilon) \rightarrow (s_0, \text{guard OR guard}, 4)$
 - $(s_0, \text{rel}, \epsilon) \rightarrow (s_0, \text{ID EQ ID}, 5)$
 - $(s_0, \text{rel}, \epsilon) \rightarrow (s_0, \text{ID LEQ ID}, 6)$
 - $(s_0, T, T) \rightarrow (s_0, \epsilon, \epsilon)$, where $T \in \Sigma$.

b) The input word is $w := \text{ID EQ ID AND ID LEQ ID}$. An accepting computation is as follows:
 $(w, \text{start}, \epsilon) \vdash (w, \text{guard}, 1) \vdash (w, \text{guard AND guard}, 13) \vdash (w, \text{rel AND guard}, 132) \vdash$
 $(w, \text{ID EQ ID AND guard}, 1325) \vdash^* (\text{ID LEQ ID}, \text{guard}, 1325) \vdash (\text{ID LE ID}, \text{rel}, 13252)$
 $\vdash (\text{ID LE ID}, \text{ID LE ID}, 132526) \vdash^* (\epsilon, \epsilon, 132526)$.

c) $G' :=$

- start $\rightarrow$ guard
- guard $\rightarrow$ rel subguard
- subguard $\rightarrow \epsilon$
- subguard $\rightarrow$ AND guard
- subguard $\rightarrow$ OR guard
- rel $\rightarrow$ ID subrel
- subrel $\rightarrow$ EQ ID
- subrel $\rightarrow$ LEQ ID.

Look-head table for G' :

π	$la(\pi)$
1	ID
2	ID
3	ϵ
4	AND
5	OR
6	ID
7	EQ
8	LEQ

	start	guard	subguard	rel	subrel	ID	EQ	LEQ	ϵ	AND	OR
ID	1	2				6					
EQ							7				
d) LEQ						8					
ϵ			3						accept		
AND			4							pop	
OR			5								pop

e) Let $w := \text{ID EQ ID AND ID LEQ ID}$. The accepting computation is as follows:
 $(w, \text{start}, \epsilon) \vdash (w, \text{guard}, 1) \vdash (w, \text{rel subguard}, 12) \vdash (w, \text{ID subrel subguard}, 126) \vdash$
 $(w[1\dots], \text{subrel subguard}, 126) \vdash (w[1\dots], \text{EQ ID subguard}, 1267) \vdash$
 $(w[3\dots], \text{subguard}, 1267) \vdash (w[3\dots], \text{AND guard}, 12674) \vdash (w[4\dots], \text{guard}, 12674) \vdash^*$
 $(w[4\dots], \text{ID subrel subguard}, 1267426) \vdash (\text{LEQ ID}, \text{subrel subguard}, 1267426) \vdash$
 $(\text{LEQ ID}, \text{LEQ ID subguard}, 12674268) \vdash (\epsilon, \text{subguard}, 12674268) \vdash (\epsilon, \epsilon, 12674683)$.