Department of Information and Computing Sciences
Utrecht University

INFOB3TC – Solutions for Exam 2

Johan Jeuring

Friday, 30 January 2015, 11:00–13:00

Please keep in mind that there are often many possible solutions and that these example solutions may contain mistakes.

Questions

Regular expressions and languages

1 (5+5 points). Consider the grammars for the regular languages L_1 and L_2:

$$\begin{array}{ll} L_1: & S \rightarrow \mathtt{b}A \mid \mathtt{a}S \\ & A \rightarrow \mathtt{a}S \mid \varepsilon \\ L_2: & S \rightarrow \mathtt{b}S \mid S\mathtt{a} \mid \varepsilon \end{array}$$

Give a regular expression for each language. •

Solution 1.

$$\begin{array}{ll} L_1: & (\mathtt{a} + \mathtt{ba})^*\mathtt{b} \\ L_2: & \mathtt{b}^*\mathtt{a}^* \end{array}$$

Marking
You either get full points or no points for both subquestions.
No penalty for using notation in a novel way (such as ?a instead of a?).
L_2 was answered correctly by almost everybody.
The most common mistakes in L_1 were: empty string part of the language, and accepts the string bb.

∘

2 (10+10 points)**.** For each language definition below, show whether or not the language is regular. If it is regular, give one of the following:

(a) a regular grammar in an acceptable form,

(b) a regular expression, or

(c) a finite state automaton.

If the language is not regular, prove that using the pumping lemma for regular languages.

(a) $\{\, \mathsf{o}^m \mathsf{p}^n \mid n = m + 1 \,\}$

(b) $\{\, \mathsf{3}^j \mathsf{7}^k \mid j > 2, k < 5 \,\}$

•

Solution 2.

(a) The language $L = \{\, \mathsf{o}^m \mathsf{p}^n \mid n = m+1 \,\}$ is not regular. To prove it, we must assume it is regular and find a contradiction with the pumping lemma.
Let $x = \varepsilon$, $y = \mathsf{o}^m$, $z = \mathsf{p}^{m+1}$.
Then, $xyz = \mathsf{o}^m \mathsf{p}^{m+1} \in L$ and $|y| \geqslant m$.
From the pumping lemma, we know there must be a loop in y, i.e. $y = uvw$ with $q = |v| > 0$ such that $xuv^i wz \in L$ for all $i \in \mathbb{N}$.
Let $i = 2$. We expect $xuv^2wz \in L$. If $u = \mathsf{o}^s$, $v = \mathsf{o}^q$, $w = \mathsf{o}^r$, then we expect $\mathsf{o}^s \mathsf{o}^{2\ q} \mathsf{o}^r \mathsf{p}^{m+1} \in L$. But it does not, because $s + 2\ q + r > m$ Therefore, L is not regular.

(b) Two possible options:

$\mathsf{333}^+(\varepsilon + \mathsf{7} + \mathsf{77} + \mathsf{777} + \mathsf{7777})$
$\mathsf{333}^+\mathsf{7?7?7?7?}$

Marking
a:
No proof : -8
Essential parts missing in the proof: -2 ... -8
m and n swapped: -1
b:
Five 7's, no zero 7's, two 3's: -2
j and k instead of 3 and 7: -1
Exactly three 3's: -2

◦

LL parsing

In these exercises we will look at the grammar

$$
\begin{aligned}
&M \to \texttt{<}E\texttt{>}M \mid \varepsilon \\
&E \to Q \mid Q\,\texttt{;}\,E \\
&Q \to \texttt{0} \mid \texttt{1} \mid M
\end{aligned}
$$

3 (15 points)**.** Complete the table below by computing the values in the columns for the appropriate rows. Use *True* and *False* for property values and set notation for everything else.

NT	Production	*empty*	*emptyRhs*	*first*	*firstRhs*	*follow*	*lookAhead*
M							
	$M \to \texttt{<}E\texttt{>}M$						
	$M \to \varepsilon$						
E							
	$E \to Q$						
	$E \to Q\,\texttt{;}\,E$						
Q							
	$Q \to \texttt{0}$						
	$Q \to \texttt{1}$						
	$Q \to M$						

Solution 3.

NT	Production	*empty*	*emptyRhs*	*first*	*firstRhs*	*follow*	*lookAhead*
M		True		$\{\texttt{<}\}$		$\{\texttt{;},\texttt{>}\}$	
	$M \to \texttt{<}E\texttt{>}M$		False		$\{\texttt{<}\}$		$\{\texttt{<}\}$
	$M \to \varepsilon$		True		$\{\,\}$		$\{\texttt{;},\texttt{>}\}$
E		True		$\{\texttt{0},\texttt{1},\texttt{<},\texttt{;}\}$		$\{\texttt{>}\}$	
	$E \to Q$		True		$\{\texttt{0},\texttt{1},\texttt{<}\}$		$\{\texttt{0},\texttt{1},\texttt{<},\texttt{>}\}$
	$E \to Q\,\texttt{;}\,E$		False		$\{\texttt{0},\texttt{1},\texttt{<},\texttt{;}\}$		$\{\texttt{0},\texttt{1},\texttt{<},\texttt{;}\}$
Q		True		$\{\texttt{0},\texttt{1},\texttt{<}\}$		$\{\texttt{;},\texttt{>}\}$	
	$Q \to \texttt{0}$		False		$\{\texttt{0}\}$		$\{\texttt{0}\}$
	$Q \to \texttt{1}$		False		$\{\texttt{1}\}$		$\{\texttt{1}\}$
	$Q \to M$		True		$\{\texttt{<}\}$		$\{\texttt{<},\texttt{;},\texttt{>}\}$

Marking
-1 per erroneous cell

4 (10 points)**.** Is the above grammar LL(1)? Explain how you arrived at your answer. If the grammar is not LL(1), transform the grammar such that is LL(1) and complete a new table with only the rows that differ from the old table. •

Solution 4.

The above grammar is not LL(1) because the *lookAhead* sets of the *E* productions have a non-empty intersection. To make this grammar LL(1), we only need to left-factor *E*.

NT	Production	*empty*	*emptyRhs*	*first*	*firstRhs*	*follow*	*lookAhead*
	$E \to QF$		True		{`0`, `1`, `<`}		{`0`, `1`, `<`, `>`}
F		True		{`;`}		{`>`}	
	$F \to$ `;`E		False		{`;`}		{`;`}
	$F \to$ *epsilon*		True		{ }		{`>`}

Marking

The points are divided as follows: 4 points for the LL(1) question, 3 points for the transformation, and 3 points for the updated table.

◦

5 (5 points)**.** Show the steps that a parser for the above LL(1) grammar (after transformation if necessary) goes through to recognize the following input sequence:

```
<0;<1>>
```

For each step (one per line), show the stack, the remaining input, and the action (followed by the relevant symbol or production) performed. If you reach a step in which you cannot proceed, note the action as "error." •

Solution 5.

stack	**input**	**action**
M	`<0;<1>>`	initial state
$<E>M$	`<0;<1>>`	expand M
$E>M$	`0;<1>>`	match `<`
$QF>M$	`0;<1>>`	expand E
$\mathtt{0}F>M$	`0;<1>>`	expand Q
$F>M$	`;<1>>`	match `0`
$\mathtt{;}E>M$	`;<1>>`	expand F
$E>M$	`<1>>`	match `;`
$QF>M$	`<1>>`	expand E
$MF>M$	`<1>>`	expand Q
$<E>MF>M$	`<1>>`	expand M
$E>MF>M$	`1>>`	match `<`
$QF>MF>M$	`1>>`	expand E
$\mathtt{1}F>MF>M$	`1>>`	expand Q
$F>MF>M$	`>>`	match `1`
$>MF>M$	`>>`	expand F
$MF>M$	`>`	match `>`
$F>M$	`>`	expand M
$>M$	`>`	expand F
M	ε	match `>`
ε	ε	expand M

Marking
Not a left-most derivation (but otherwise OK): -3
Expanding and matching separated: -2
Error in the derivation: -2

○

LR parsing

Consider the following grammar, with start symbol S:

$$
\begin{aligned}
S &\rightarrow L = R \mid R \\
L &\rightarrow * \, R \mid \mathtt{i} \\
R &\rightarrow L
\end{aligned}
$$

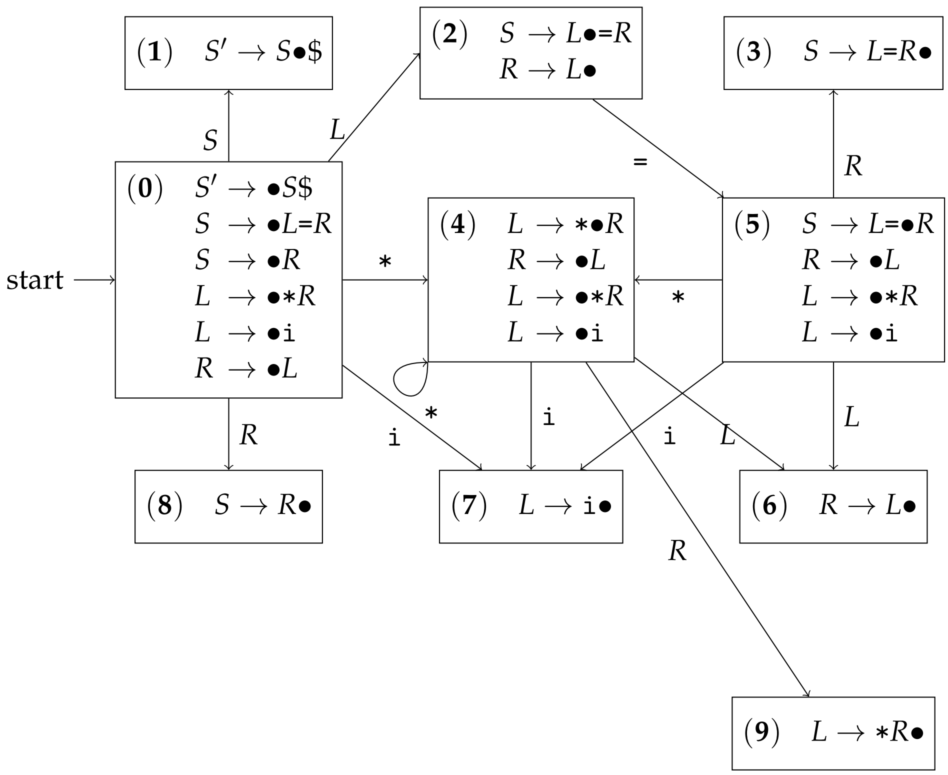
We augment the grammar above in preparation for LR parsing:

$$S' \rightarrow S\$$$

and S' becomes the new start symbol.

6 (10 points)**.** Compute the LR(0) automaton corresponding to the full grammar. Number each state for future reference. •

Solution 6.



Marking
No closures of item sets: -4
Missing transitions: -1 ... -3

○

7 (10 points)**.** Classify each state in your LR(0) automaton as a shift state, reduce state, or shift-reduce conflict state. Also mark potential reduce-reduce conflicts. If there are conflicts, would applying SLR(1) parsing help to resolve these? •

Solution 7. The states (**3**), (**6**), (**7**), (**8**), (**9**) are all reduce states. The states (**0**), (**1**), (**4**), (**5**) are all shift states. The state (**2**) is a shift-reduce state, and therefore there is a shift-reduce conflict.

Would a SLR(1) approach help in parsing this grammar? Since = is in the follow set of R (and L), we cannot distinguish between reducing, or shifting an =. So this grammar is also not SLR(1).

Marking

Shift/reduce conflicts wrong: -2

SLR(1) helps (or nothing for this question): -5

◦

8 (10 points)**.** Play through the LR parsing process for the word `**i=*i$`. If there is a choice somewhere, make this explicit. Show in each step at which state in your LR(0) automaton you are. •

Solution 8.

stack	input	remark
$(\mathbf{0})$	`**i=*i$`	shift
$(\mathbf{0}){*}(\mathbf{4})$	`*i=*i$`	shift
$(\mathbf{0}){*}(\mathbf{4})\ {*}(\mathbf{4})$	`i=*i$`	shift
$(\mathbf{0}){*}(\mathbf{4})\ {*}(\mathbf{4})\ \mathtt{i}(\mathbf{7})$	`=*i$`	reduce by $L \to \mathtt{i}$
$(\mathbf{0}){*}(\mathbf{4})\ {*}(\mathbf{4})L(\mathbf{6})$	`=*i$`	reduce by $R \to L$
$(\mathbf{0}){*}(\mathbf{4})\ {*}(\mathbf{4})R(\mathbf{9})$	`=*i$`	reduce by $L \to {*}R$
$(\mathbf{0}){*}(\mathbf{4})\ L(\mathbf{6})$	`=*i$`	reduce by $L \to R$
$(\mathbf{0}){*}(\mathbf{4})\ R(\mathbf{9})$	`=*i$`	reduce by $L \to {*}R$
$(\mathbf{0})L(\mathbf{2})$	`=*i$`	shift (could reduce here, but that would fail)
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})$	`*i$`	shift
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ {*}(\mathbf{4})$	`i$`	shift
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ {*}(\mathbf{4})\ \mathtt{i}(\mathbf{7})$	$	reduce by $L \to \mathtt{i}$
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ {*}(\mathbf{4})\ L(\mathbf{6})$	$	reduce by $R \to L$
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ {*}(\mathbf{4})\ R(\mathbf{9})$	$	reduce by $L \to {*}R$
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ L(\mathbf{6})$	$	reduce by $R \to L$
$(\mathbf{0})L(\mathbf{2})\ \mathtt{=}(\mathbf{5})\ R(\mathbf{3})$	$	reduce by $S \to L\mathtt{=}R$
$(\mathbf{0})S(\mathbf{1})$	$	ready

Marking

Not completely finished the derivation: -1 or -2
Only stack structure correct: -8
Stack structure half correct: -9

○