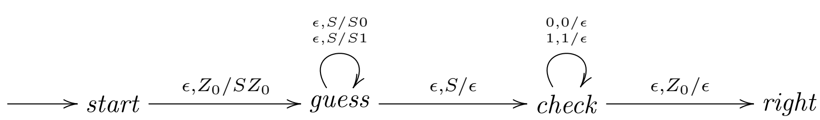
CFL4 - Solution Sketches

March 9, 2020

We sketch here the solutions for the exercises of lecture CFL3 in a brief manner. Note that a proper solution would require more detailed descriptions, explanations, and in some cases examples. Some of the exercises may have more than one solution, and we just show one of them.

Exercise 4.1.(1)



Exercise 4.1.(2)

$$(start, 010, Z_0) \vdash (guess, 010, SZ_0) \vdash (guess, 010, S1Z_0) \vdash (check, 010, 1Z_0) \not\vdash$$

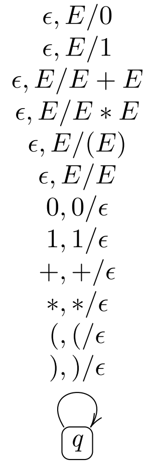
Exercise 4.1.(3)

$$(start, 010, Z_0) \vdash (guess, 010, SZ_0) \vdash (check, 010, Z_0) \not\vdash (right!, 010, \epsilon) \not\vdash$$

Exercise 4.1.(4)

$$(start, 010, Z_0) \vdash (guess, 010, SZ_0) \vdash (guess, 010, S0Z_0) \vdash (guess, 010, S10Z_0) \vdash (guess, 010, S010Z_0) \vdash$$
$$(check, 010, 010Z_0) \vdash (check, 10, 10Z_0) \vdash (check, 0, 0Z_0) \vdash (check, \epsilon, Z_0) \vdash (right!, \epsilon, \epsilon)$$

Exercise 4.2.



Exercise 4.3 The original grammar is

$$
\begin{array}{lcl}
B & \rightarrow & V\bullet \mid \texttt{not}\ B \mid V\ \texttt{or}\ B \mid V\ \texttt{and}\ B \\
V & \rightarrow & \texttt{true} \mid \texttt{false}
\end{array}
$$

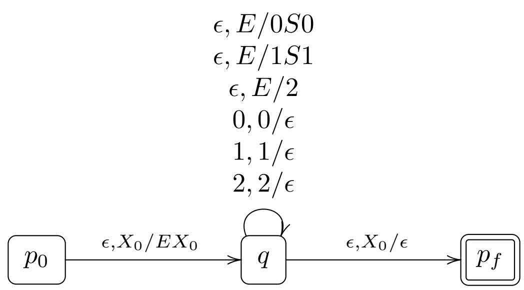
After factorising all productions of the form $B \rightarrow V\gamma$ we obtain

$$
\begin{array}{lcl}
B & \rightarrow & VA \mid \texttt{not}\ B \\
A & \rightarrow & \bullet \mid \texttt{or}\ B \mid \texttt{and}\ B \\
V & \rightarrow & \texttt{true} \mid \texttt{false}
\end{array}
$$

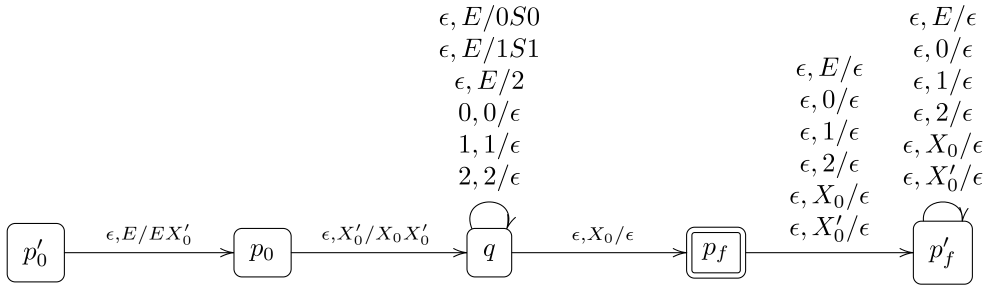
We can then complete the parsing table as follows

	not	$\bullet$	or	and	true	false
B	$B \rightarrow \texttt{not}\ B$				$B \rightarrow VA$	$B \rightarrow VA$
A		$A \rightarrow \bullet$	$A \rightarrow \texttt{or}\ V$	$A \rightarrow \texttt{and}\ V$		
V					$V \rightarrow \texttt{true}$	$V \rightarrow \texttt{false}$

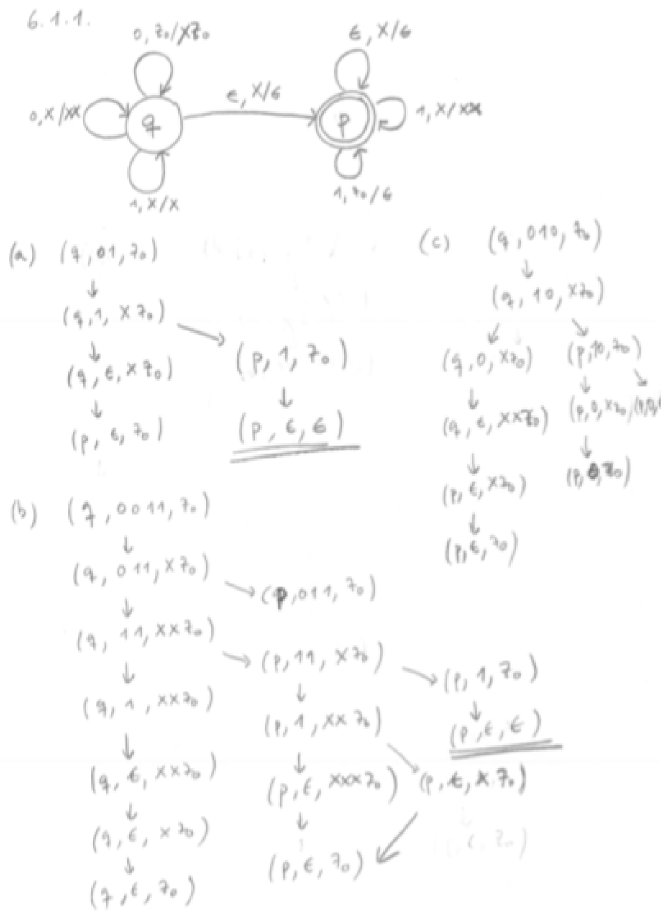
Exercise 4.4.(1)



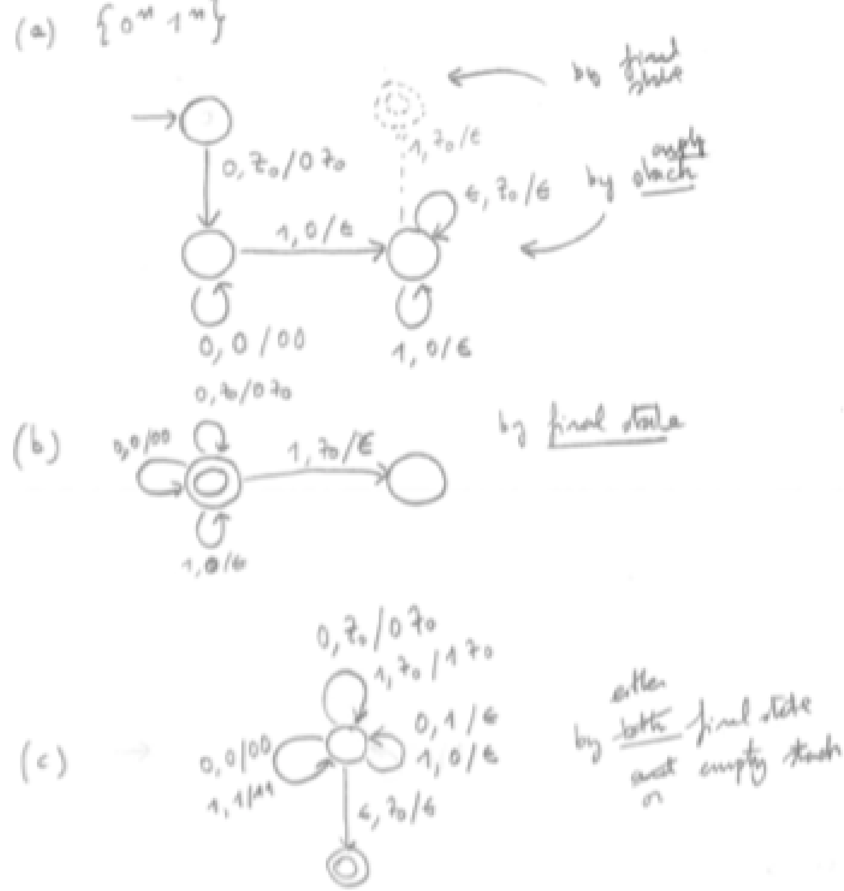
Exercise 4.4.(2)



Exercise 4.5



Exercise 4.6 There are two ways of addressing this exercise. The first one is to try to design the PDAs directly. For example, one could end up with:

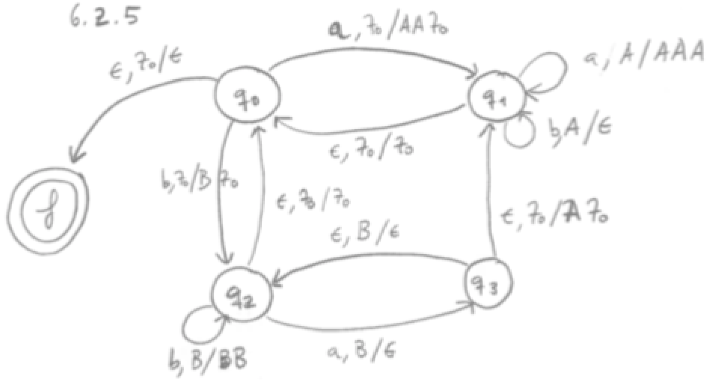


The second one is more systematic: each of the languages (a)–(c) can be easily defined with a CFG G as seen in previous exercises. One can then translate G in a PDA using the construction seen in class.

Exercise 4.7

EXERCISES Ø5

6.2.5



(a) (q_0, bab, Z_0)

$\downarrow$

(q_2, ab, BZ_0)

$\downarrow$

(q_3, b, Z_0)

$\downarrow$

(q_1, b, AZ_0)

$\downarrow$

(q_1, ϵ, Z_0)

$\downarrow$

(q_0, ϵ, Z_0)

$\downarrow$

$\underline{(f, \epsilon, \epsilon)}$

(b) (q_0, abb, Z_0)

$\downarrow$

(q_1, bb, AAZ_0)

(d) Words in $\{a, b\}$ with twice as much b's as a's.

Exercise 4.8

(←) Stack	Input	Rule
B	`true` `and` `true` `or` `true` $\bullet$	Choose $B \to VA$
VA	`true` `and` `true` `or` `true` $\bullet$	Choose $V \to$ `true`
`true`A	`true` `and` `true` `or` `true` $\bullet$	Match
A	`and` `true` `or` `true` $\bullet$	Choose $A \to$ `and`B
`and`B	`and` `true` `or` `true` $\bullet$	Match
B	`true` `or` `true` $\bullet$	Choose $B \to VA$
VA	`true` `or` `true` $\bullet$	Choose $V \to$ `true`
`true`A	`true` `or` `true` $\bullet$	Match
A	`or` `true` $\bullet$	Choose $A \to$ `or` B
`or`B	`or` `true` $\bullet$	Match
B	`true` $\bullet$	Choose $B \to AV$
VA	`true` $\bullet$	Choose $V \to$ `true`
`true`A	`true` $\bullet$	Match
A	$\bullet$	Choose $A \to \bullet$
$\bullet$	$\bullet$	Match
ϵ	ϵ	

Exercise 4.9

(←) Stack	Input	Rule
E	$0 * 0 + 0 \bullet$	Choose $E \to AT$
AT	$0 * 0 + 0 \bullet$	Choose $A \to 0$
$0T$	$0 * 0 + 0 \bullet$	Match
T	$*0 + 0 \bullet$	Choose $T \to *E$
$*E$	$*0 + 0 \bullet$	Match
E	$0 + 0 \bullet$	Choose $E \to AT$
AT	$0 + 0 \bullet$	Choose $A \to 0$
$0T$	$0 + 0 \bullet$	Match
T	$+0 \bullet$	Choose $T \to +E$
$+E$	$+0 \bullet$	Match
E	$0 \bullet$	Choose $E \to AT$
AT	$0 \bullet$	Choose $A \to 0$
$0T$	$0 \bullet$	Match
T	$\bullet$	Choose $T \to \bullet$
$\bullet$	$\bullet$	Match
ϵ	ϵ	

Exercise 4.10 (1)

$$\begin{array}{c}
\epsilon, P/A\,\texttt{;}\,P \\
\epsilon, P/A\,\texttt{;}\,P \\
\epsilon, P/A\,\texttt{;}\,P \\
\epsilon, P/\texttt{stop} \\
\epsilon, P/(P) \\
\epsilon, A/\texttt{ping} \\
\epsilon, A/\texttt{pong} \\
\texttt{;}, \texttt{;}/\epsilon \\
\texttt{or}, \texttt{or}/\epsilon \\
\texttt{and}, \texttt{and}/\epsilon \\
\texttt{stop}, \texttt{stop}/\epsilon \\
\texttt{ping}, \texttt{ping}/\epsilon \\
\texttt{pong}, \texttt{pong}/\epsilon \\
\texttt{(}, \texttt{(}/\epsilon \\
\texttt{)}, \texttt{)}/\epsilon \\
\boxed{q}
\end{array}$$

Exercise 4.10 (2)

$$\begin{array}{ll}
(q, \texttt{ping and pong ; stop}, & P) \vdash \\
(q, \texttt{ping and pong ; stop}, & A \texttt{ and } P) \vdash \\
(q, \texttt{ping and pong ; stop}, & \texttt{ping and } P) \vdash \\
(q, \texttt{and pong ; stop}, & \texttt{and } P) \vdash \\
(q, \texttt{pong ; stop}, & P) \vdash \\
(q, \texttt{pong ; stop}, & A\texttt{;}P) \vdash \\
(q, \texttt{pong ; stop}, & \texttt{pong;}P) \vdash \\
(q, \texttt{; stop}, & \texttt{;}P) \vdash \\
(q, \texttt{stop}, & P) \vdash \\
(q, \texttt{stop}, & \texttt{stop}) \vdash \\
(q, \epsilon, & \epsilon)
\end{array}$$

Exercise 4.10 (3) The original grammar is

$$\begin{array}{lcl}
P & \rightarrow & A \texttt{ ; } P \mid A \texttt{ or } P \mid A \texttt{ and } P \mid \texttt{stop} \mid (P) \\
A & \rightarrow & \texttt{ping} \mid \texttt{pong}
\end{array}$$

The factorised grammar is

$$\begin{array}{lcl}
P & \rightarrow & AQ \mid \texttt{stop} \mid (P) \\
Q & \rightarrow & \texttt{ ; } P \mid \texttt{ or } P \mid \texttt{ and } P \\
A & \rightarrow & \texttt{ping} \mid \texttt{pong}
\end{array}$$

Exercise 4.10 (4)

	`stop`	(	)	;	`or`	`and`	`ping`	`pong`
P	$P \to$ `stop`	$P \to (P)$					$P \to AQ$	$P \to AQ$
Q				$Q \to$; P	$Q \to$ `or` P	$Q \to$ `and` P		
A							$A \to$ `ping`	$A \to$ `pong`

Exercise 4.10 (5)

$(\leftarrow)$ Stack	Input	Rule
P	`ping and pong ; stop`	CHOOSE $P \to AQ$
AQ	`ping and pong ; stop`	CHOOSE $A \to$ `ping`
`ping`Q	`ping and pong ; stop`	MATCH
Q	`and pong ; stop`	CHOOSE $Q \to$ `and` P
`and`P	`and pong ; stop`	MATCH
P	`pong ; stop`	CHOOSE $P \to AQ$
AQ	`pong ; stop`	CHOOSE $A \to$ `pong`
`pong`Q	`pong ; stop`	MATCH
Q	`; stop`	CHOOSE $Q \to$; P
;P	`; stop`	MATCH
P	`stop`	CHOOSE $P \to$ `stop`
`stop`	`stop`	MATCH
ϵ	ϵ	