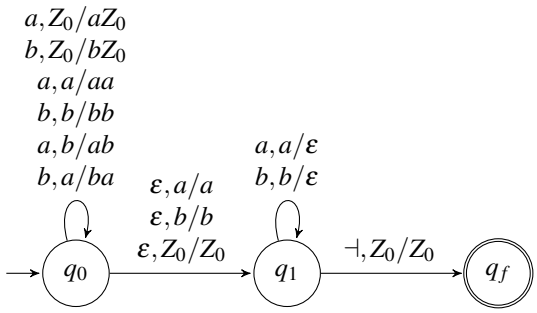
Introduction to Language Theory and Compilation
Solutions

Session 7: Pushdown automata and parsing

Exercises

Exercise 1

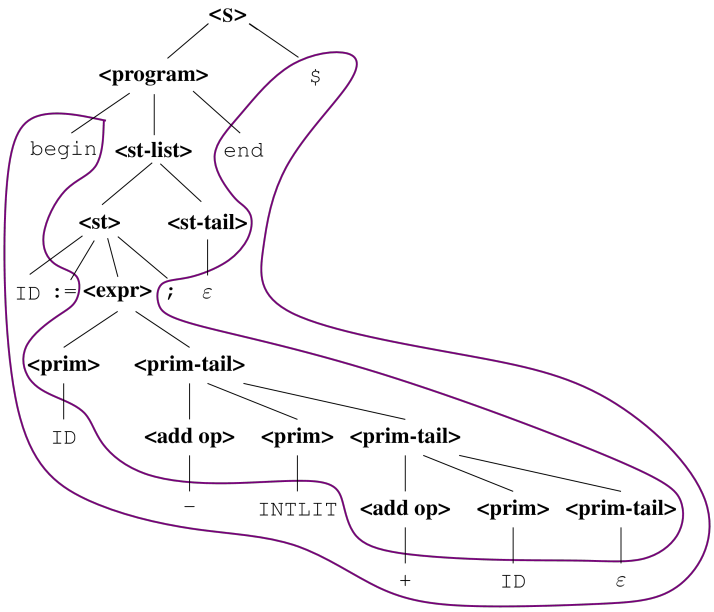
In the case of L_{pal}, we cannot, as we did with $L_{pal\#}$, design a *deterministic* pushdown automaton to accept the language ! We need the power of non-determinism to be able to "guess" properly the middle of the input word and start checking if the remaining input is indeed the mirror image of the first half. In the following, we design a PDA such that $L(P) = L_{pal}$, that is P accepts by *final state*. Of course, it is always possible to design a PDA P' such that $N(P') = L_{pal}$, that is P' accepts by *empty stack*, but we leave that as an exercise. Finally, we choose here to have, as a convention, a distinguished "end of the input word symbol" $\dashv$ to avoid any confusion that could appear while using only ε, but this is not necessary.



Let us now give an *accepting* execution on the input word *abaaaaba*:

$\langle q_0, abaaaaba \dashv, Z_0\rangle \vdash \langle q_0, baaaaba \dashv, aZ_0\rangle \vdash \langle q_0, aaaaba \dashv, baZ_0\rangle \vdash \langle q_0, aaaba \dashv, abaZ_0\rangle \vdash \langle q_0, aaba \dashv, aabaZ_0\rangle \vdash \langle q_1, aaba \dashv, aabaZ_0\rangle\langle q_1, aba \dashv, abaZ_0\rangle \vdash \langle q_1, ba \dashv, baZ_0\rangle \vdash \langle q_1, a \dashv, aZ_0\rangle \vdash \langle q_1, \dashv, Z_0\rangle \vdash \langle q_f, \dashv, Z_0\rangle$

Exercise 2



Exercise 3

Stack	Input	Action
<S> ⊣	`begin A := BB - 314 + A ; end $`	P1
<program> $ ⊣	`begin A := BB - 314 + A ; end $`	P2
begin <st-list> end $ ⊣	`begin A := BB - 314 + A ; end $`	M
<st-list> end $ ⊣	`A := BB - 314 + A ; end $`	P3
<st> <st-tail> end $ ⊣	`A := BB - 314 + A ; end $`	P6
ID := <expression> ; <st-tail> end $ ⊣	`A := BB - 314 + A ; end $`	M
:= <expression> ; <st-tail> end $ ⊣	`:= BB - 314 + A ; end $`	M
<expression> ; <st-tail> end $ ⊣	`BB - 314 + A ; end $`	P15
<primary> <primary-tail> ; <st-tail> end $ ⊣	`BB - 314 + A ; end $`	P19
ID <primary-tail> ; <st-tail> end $ ⊣	`BB - 314 + A ; end $`	M
<primary-tail> ; <st-tail> end $ ⊣	`- 314 + A ; end $`	P16
<addop> <primary> <primary-tail> ; <st-tail> end $ ⊣	`- 314 + A ; end $`	P22
- <primary> <primary-tail> ; <st-tail> end $ ⊣	`- 314 + A ; end $`	M
<primary> <primary-tail> ; <st-tail> end $ ⊣	`314 + A ; end $`	P20
INTLIT <primary-tail> ; <st-tail> end $ ⊣	`314 + A ; end $`	M
<primary-tail> ; <st-tail> end $ ⊣	`+ A ; end $`	P16
<addop> <primary> <primary-tail> ; <st-tail> end $ ⊣	`+ A ; end $`	P21
+ <primary> <primary-tail> ; <st-tail> end $ ⊣	`+ A ; end $`	M
<primary> <primary-tail> ; <st-tail> end $ ⊣	`A ; end $`	P19
ID <primary-tail> ; <st-tail> end $ ⊣	`A ; end $`	M
hline <primary-tail> ; <st-tail> end $ ⊣	`; end $`	P17
; <st-tail> end $ ⊣	`; end $`	M
<st-tail> end $ ⊣	`end $`	P5
end $ ⊣	`end $`	M
$ ⊣	`$`	ACCEPT

Exercise 4

Stack	Input	Action
⊢	`begin A := BB - 314 + A ; end $`	S
⊢ begin	`A := BB - 314 + A ; end $`	S
⊢ begin ID	`:= BB - 314 + A ; end $`	S
⊢ begin ID :=	`BB - 314 + A ; end $`	S
⊢ begin ID := ID	`- 314 + A ; end $`	R19
⊢ begin ID := <prim>	`- 314 + A ; end $`	S
⊢ begin ID := <prim> -	`314 + A ; end $`	R22
⊢ begin ID := <prim> <addop>	`314 + A ; end $`	S
⊢ begin ID := <prim> <addop> INTLIT	`+ A ; end $`	R20
⊢ begin ID := <prim> <addop> <prim>	`+ A ; end $`	S
⊢ begin ID := <prim> <addop> <prim> +	`A ; end $`	R21
⊢ begin ID := <prim> <addop> <prim> <addop>	`A ; end $`	S
⊢ begin ID := <prim> <addop> <prim> <addop> ID	`; end $`	R19
⊢ begin ID := <prim> <addop> <prim> <addop> <primary>	`; end $`	R17
⊢ begin ID := <prim> <addop> <prim> <addop> <prim> <p-tail>	`; end $`	R16
⊢ begin ID := <prim> <addop> <prim> <p-tail>	`; end $`	R16
⊢ begin ID := <prim> <prim-tail>	`; end $`	R15
⊢ begin ID := <expression>	`; end $`	S
⊢ begin ID := <expression> ;	`end $`	R6
⊢ begin <st>	`end $`	R5
⊢ begin <st> <st-tail>	`end $`	R3
⊢ begin <st-list>	`end $`	S
⊢ begin <st-list> end	`$`	R2
⊢ <program>	`$`	S
⊢ <program> $		R1
⊢ <S>		ACCEPT