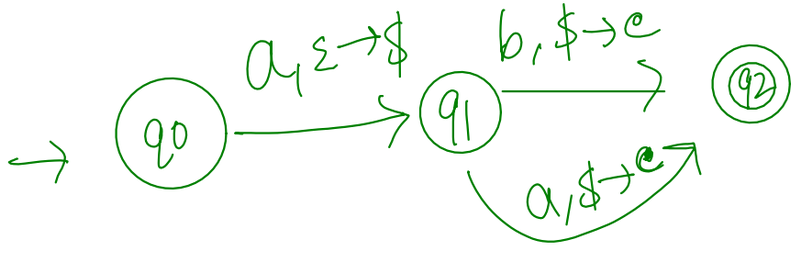
CSE322 Theory of Computation (L12)

Today

PDA to CFG



aa is accepted.

q0 q1 q2 , aa = a.a

s0 = ε, s1 = $, s2 = ε

s1 = $.ε (b t) (q1, $) ∈ δ(q0, a, ε)

s0 = ε.ε (a t)

s2 = ε.ε (b t) (q2, ε) ∈ δ(q1, a, $)

s1 = $.ε (a t)

for_stmt: 'for' exprlist 'in' testlist ':' suite ['else' ':' suite]
testlist: test (',' test)* [',']
test: or_test ['if' or_test 'else' test] | lambdef
or_test: and_test ('or' and_test)*
and_test: not_test ('and' not_test)*
not_test: 'not' not_test | comparison
comparison: expr (comp_op expr)*
comp_op: '<'|'>'|'=='|'>='|'<='|'<>'|'!='|'in'|'not' 'in'|'is'|'is' 'not'
...

pointer : '*' type_qualifier_list | '*' | '*' type_qualifier_list pointer | '*' pointer
type_qualifier_list : type_qualifier | type_qualifier_list type_qualifier
type_qualifier : 'const' | 'volatile'

Is this valid? const int * const ptr;

CFG

$$\langle EXPR \rangle \rightarrow \langle EXPR \rangle + \langle TERM \rangle \mid \langle TERM \rangle$$
$$\langle TERM \rangle \rightarrow \langle TERM \rangle \times \langle FACTOR \rangle \mid \langle FACTOR \rangle$$
$$\langle FACTOR \rangle \rightarrow (\langle EXPR \rangle) \mid a$$

A CFG is (V, Σ, R, S)

- V
- Σ
- R ($v \in V, (V \cup \Sigma)^*$)
- S

T → T

Is "(axa)+a" present in language of the above?

<EXPR> => <EXPR> + <TERM> => <FACTOR> + <TERM>
=> <FACTOR> + <FACTOR>
=> <FACTOR> + a => (<EXPR>) + a
=> ... => (a x a) + a

EXPR $\Rightarrow_4$ F + a

Derivation:

→ u,v,w: string over variables & terminals $u \in V \cup \Sigma$, v, w

A : variable

FACTOR a a (TERM ( ) a TERM ⇒ FACTOR a a (FACTOR ( ) a T)

o uAv => uwv (uAv yields uwv) if A -> w rule exists

o u =>* v (u derives v) if

u = v, or (base case)

u => v1 => v2 => ... => v for some v1, v2, ... ∃ v' s.t. u ⇒ v_1, $v_1 \Rightarrow^* v$

Language (G) = { w over terminals | S =>* w}

S =>n w : S derives w in n derivation steps.

Definition

$\Sigma_\epsilon = \Sigma \cup \{\epsilon\}$ $\Gamma_\epsilon = \Gamma \cup \{\epsilon\}$

$s_3 = bt, b \in \Gamma_\epsilon$

$s_2 = at$ where $a \in \Gamma_\epsilon, t \in \Gamma_\epsilon^*$

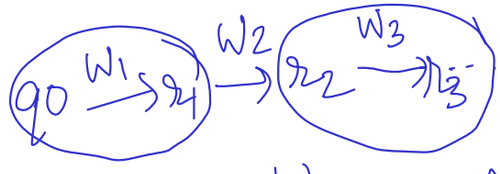
$q \xrightarrow{a \text{ from input}}$ c from top of stack

NPDA $P = \langle Q, \Sigma, \Gamma, \delta, q0, F \rangle$

* Q : set of states
* Σ : set of input symbols
* Γ : set of stack symbols
* $\delta : Q \times \Sigma_\epsilon \times \Gamma_\epsilon \to P(Q \times \Gamma_\epsilon)$ transition fn.
* $q0$: start state
* $F \subseteq Q$: set of accept states

$r_2 \xrightarrow{w_3} r_3$

s_2   s_3



$\cdots \to r_n \in F$

$w = w_1 \cdots w_m$ where $w_i \in \Sigma_\epsilon$

P accepts w if

* w can be written as a sequence of input symbols and e: w=w1 ... wm
* r0 ... rm is a sequence of states
* s0 ... sm is a sequence of strings over stack symbols

left end indicates top of the stack.

s.t.

1. r0 = q0, s0 = e
2. rm is in F
3. for i=0 ... (m-1), there exists a,b,t s.t. si=at and s(i+1)=bt and ...

$(r_{i+1}, b) \in \delta(r_i, w_{i+1}, a)$

$a \in \Gamma_\epsilon$ $t \in \Gamma_\epsilon^*$

PDA to CFG

$A_{pq} \Rightarrow^* w$ s.t. $p \overset{w}{\rightsquigarrow} q$

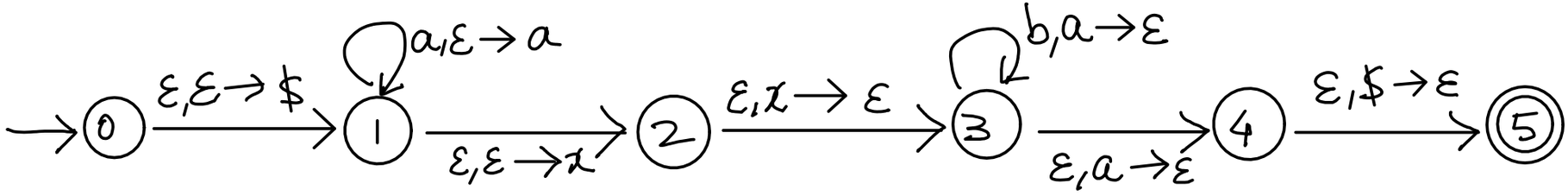
Construct G from PDA: PDA accets w iff G generates w.

Modify PDA:

* One accepting state qa.
* Stack is empty at beginning and at end.
* Each transition either pushes or pops but not both.

Given P construct P' s.t. these hold.

$a/\varepsilon \to x$ $a/x \to \varepsilon$ ~~$a/x \to y$~~



Variables

A_{00} A_{01} ⋯ A_{05} ⋯ A_{50} ⋯ A_{55} (start var.)

A_{pq} $\forall p,q \in Q$

Rules

$A_{00} \to \varepsilon$ $A_{11} \to \varepsilon$ ⋯ $A_{55} \to \varepsilon$ $A_{qq} \to \varepsilon$ $\forall q$

$A_{05} \to A_{01} A_{15}$ $A_{05} \to A_{02} A_{25}$ ⋯ $A_{55} \to A_{50} A_{05}$

$A_{pq} \to A_{pr} A_{rq}$ $\forall p,q,r$

$A_{05} \to \varepsilon A_{14} \varepsilon$ (p', x) in d(p, a, e) &
$A_{13} \to a A_{13} b$ (q, e) in d(q', b, x) for some x

$A_{pq} \to a A_{p'q'} b$ if

PDA Transitions

$\vdash$ yields $(q, w', z') \vdash (q', w'', z'')$

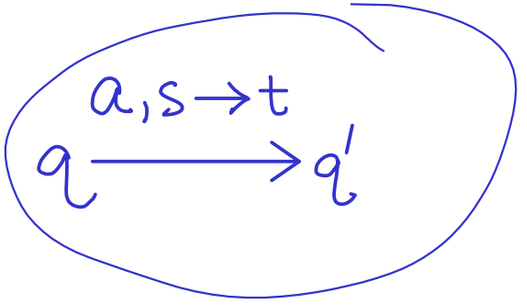
Instantaneous Description

ID of a PDA : (state, remaining input, stack contents (top-to-bottom))

Notation: $(q, aw, sZ) \vdash (q', w, tZ)$ means $\delta(q,a,s) = \{(q',t), \ldots\}$

($aw \in \Sigma_\varepsilon^*$, $sZ \in \Gamma_\varepsilon^*$; $\vdash$ yields)

Notation: $\vdash^*$ denotes multiple moves (Σ_ε, Γ_ε)



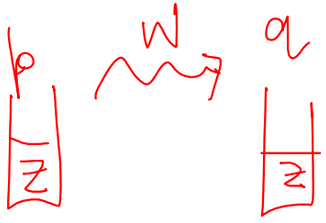
Suppose $\delta(q,a,s)$ contains (q',t).

Then $\forall z, w$ $(q, aw, sz) \vdash (q', w, tz)$

For multiple moves, define $\vdash^*$

$I \vdash^* I$ $\forall I$, $I \vdash^* J$ if $I \vdash K$ and $K \vdash^* J$

initial ID: (q_0, w, ε)



PDA P accepts w if $(q_0, w, \varepsilon) \vdash^* (q_{accept}, \varepsilon, Z)$ for some $Z \in \Gamma_\varepsilon^*$

(initial ID; some final ID)

" " $q_{accept} \in F$

Lemma: $\forall z \in \Gamma_\varepsilon^*$ $(p,w,\varepsilon) \vdash^* (q,\varepsilon,\varepsilon)$ and stack is not empty at any intermediate stage $\Rightarrow (p,w,z) \vdash^* (q,\varepsilon,z)$ stack is not empty at any intermediate stage

How to prove this?

Why is this important!

Exercise: Is the converse true?