CSCE 355
3/14/22

Push-down Automata

Def'n: A pushdown automaton (PDA) is a tuple $\langle Q, \Sigma, \Gamma, \delta, q_0, z_0, F \rangle$ where

- Q is a finite set (elements called states)
- Σ, Γ are alphabets. Σ is the input alphabet
  Γ is the stack alphabet
- $\delta : Q \times (\Sigma \cup \{\varepsilon\}) \times \Gamma \to 2^{Q \times \Gamma^*} \cap (\text{finite sets})$

  Given state $q \in Q$, $a \in \Sigma \cup \{\varepsilon\}$, $t \in \Gamma$

  $\delta(q, a, t)$ contains (r, γ) $[r \in Q, \gamma \in \Gamma^*]$

  means in this step, can change state to r, and consume a on the input and replace t on top of the stack with γ (pop t and push each symbol of γ onto the stack right to left, so that 1st symbol of γ is on top of the stack)
- $q_0 \in Q$ (the start state)
- $z_0 \in \Gamma$ (the bottom stack marker)

Initially, Z_0 is on the stack and nothing else.

- $F \subseteq Q$ (elements are called <u>accepting states</u>)

<u>Ex:</u> $L = \{0^n 1^n : n \geq 0\}$ (not regular)

PDA P recognizing L:

$Q = \{p, q, r\}$ p start state

$\Sigma = \{0, 1\}$ Z_0 bottom stack marker

$\Gamma = \{Z_0, +\}$

$F = \{r\}$

$\delta(p, 0, Z_0) = \{(p, +Z_0)\}$ // push +

$\delta(p, 0, +) = \{(p, ++)\}$ // push +

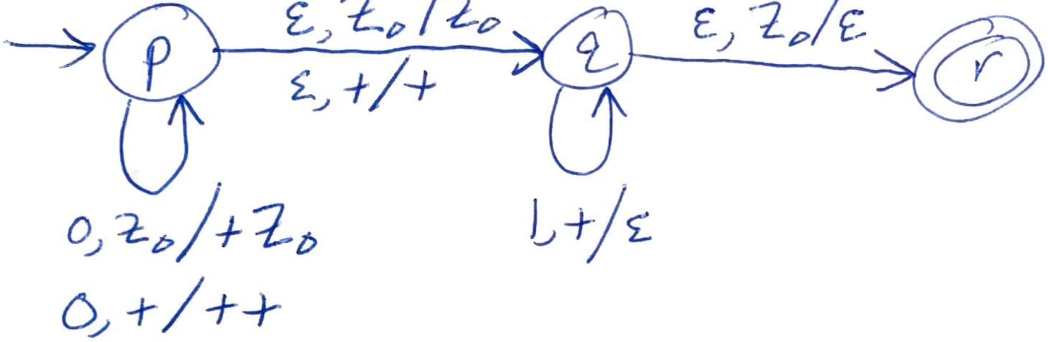
$\delta(p, \varepsilon, Z_0) = \{(q, Z_0)\}$ } don't change stack

$\delta(p, \varepsilon, +) = \{(q, +)\}$ } don't change stack

$\delta(q, 1, +) = \{(q, \varepsilon)\}$ // pop +

$\delta(q, \varepsilon, Z_0) = \{(r, \varepsilon)\}$

③ Transition diagram:



Ex: Properly nested delimiters ( ) and [ ]

$\Sigma = \{ '(', ')', '[', ']' \}$ "([ )]" not accepted!

PDA to accept all strings of well-balanced delimiters '(', ')', '[', ']' (and nothing else)

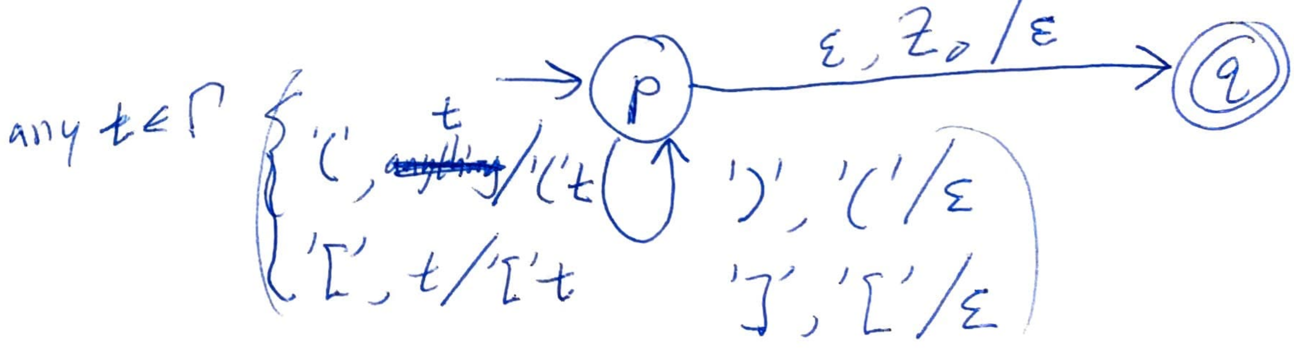
$Q = \{p, q\}$ p start state

Z_0 bottom stack marker

$F = \{q\}$

$\Gamma = \{ '(', '[', Z_0 \}$

Transition diagram



④ PDA to recognize

$$L = \{ w \in \{0,1\}^* \mid w \text{ has an equal number of 0's and 1's} \}$$

Exercise! (not to hand in)

Formal semantics

Def: Let $P = \langle Q, \Sigma, \Gamma, \delta, q_0, Z_0, F \rangle$ be a PDA and let $w \in \Sigma^*$ be a string.

An instantaneous description (ID, also called a configuration) is a ~~pair (x, γ)~~ triple (q, x, γ) where $q \in Q$, $x \in \Sigma^*$, $\gamma \in \Gamma^*$.

(x is a suffix of w)

Idea: Complete snapshot of the state of P's computation on input w at any point in time.

- q is P's current state
- x (suffix of w) is the portion of the input yet to be consumed
- γ current stack contents (left → right means top → bottom)

(5) **Def**: Let ID_1 and ID_2 be ~~two~~ ID's of P

$$ID_1 = (q, x, \gamma)$$

$$ID_2 = (r, y, \eta)$$

Say that ID_2 *succeeds* ID_1 (is a *successor* of ID_1)
(written $ID_1 \vdash ID_2$) if

– $x = ay$ for some $a \in \Sigma \cup \{\varepsilon\}$

and

– $\gamma \neq \varepsilon$ and has first symbol t
and rest of the stack is $\alpha \in \Gamma^*$
$(\gamma = t\alpha)$

and

– $(r, \beta) \in \delta(q, a, t)$

and

– $\eta = \beta\alpha$.

The *initial ID* on input w is
~~(w, Z_0)~~. (q_0, w, Z_0)