CS310

Chomsky Normal Form

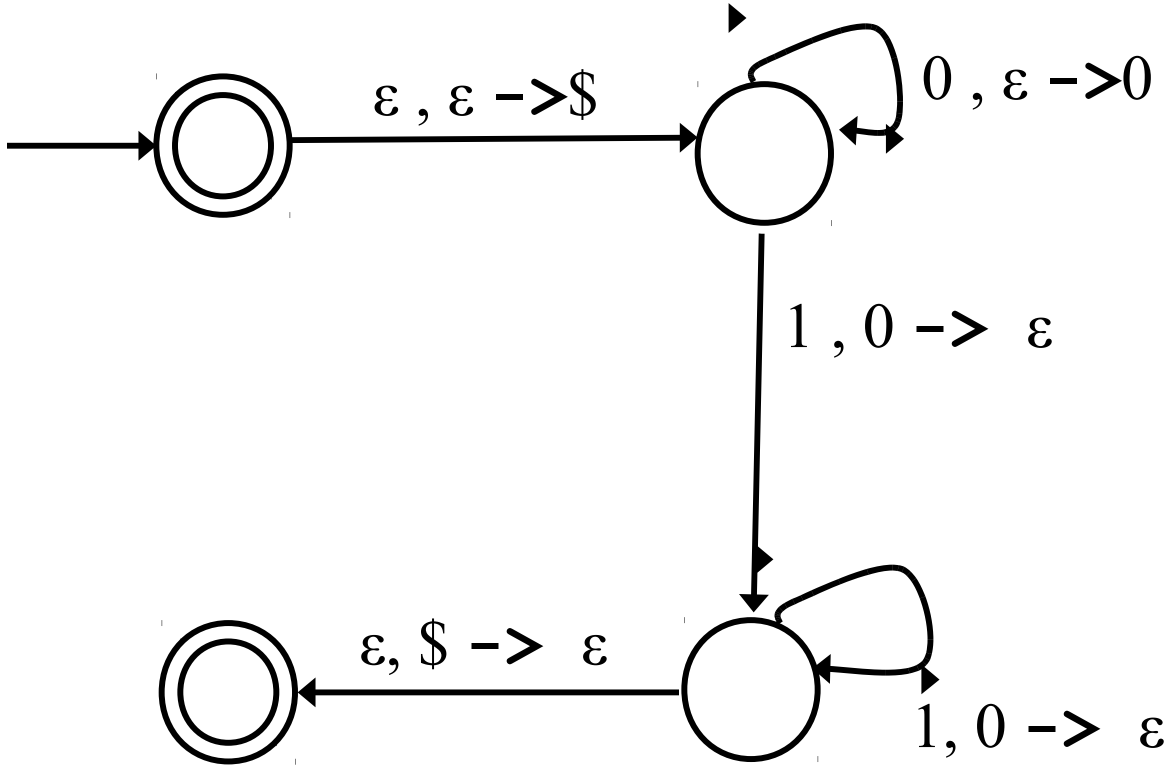
Section: 2.1 page 106

PDA → CFG

Section 2.2 page 115

October 18, 2010

Quick Review



a , b ->c

Read a from input, read b from stack, push c onto stack to take this transition

a = ε, read no input
b = ε, don't pop data from stack
c = ε, don't push data onto stack

Example

- $L = \{ a^i b^j c^k \mid i, j, k \geq 0, i = j \text{ or } i = k \}$

Theorem

- A Language is context free if and only if there exists a PDA that recognizes it.

- Lemma:
  - If a language is context free, then some PDA recognizes it
  - Show: a CFG can be transformed into a PDA
- Lemma:
  - If a PDA recognizes a language, then it is context free
  - *Program a PDA to run a CFG*

Construct PDA from CFG p 116

- $L = \{a^n b b^n \mid n \geq 0\}$

  CFG?

1) Place $, start variable on stack

2) Repeat:

   a) if variable A is on top of stack, use replacement rule A *(pop)* **->**w *(push)*

   b) if terminal on top, read input, compare. If match, repeat, else die

   c) if $ on top, enter accept, die if there's more input

Construct CFG from PDA

- Read p 119 – 122 for next time, we'll discuss the proof

Chomsky Normal Form

- CNF presents a grammar in a standard, simplified form:

  A-> BC

  A -> *a*

  S -> ε

  - Where A,B,C are variables and B and C are not the start variable
  - *a* is a terminal
  - The rule S -> ε is allowed so the language can generate the empty string (optional)

CNF Benefits

- Easier to prove statements about CFG's when in CNF

- Any CFG can be converted to CNF
- Remove productions:

A -> ε	to empty
A -> B	Unit rule
A -> s, s contains a terminal and |s| > 1	
A -> s, |s| > 2	

$s \in \{ V \cup \Sigma \}^*$

Removing A -> ε

S -> UAV

A -> ε

- A variable A is ***nullable*** if A-*> ε

  Find all nullable variables

  Remove all ε transitions

  If T -> $s_1 A s_2$ and A nullable

  then add T -> $s_1 s_2$

Example

S -> TU

T -> AB

A -> aA | ε

B -> bB | ε

U -> ccA | B

Nullable variables?

Productions removed?

Productions added?

Removing A -> B (Unit Productions)

A-> B

B-> s

$s \in \{ V \cup \Sigma \}^*$

- A variable B is A-derivable if A-*>B

  Find all A-derivable variables for each A

  Remove all unit transitions

  If B -> s and B is A-derivable

  then add A -> s

Example

S -> TU | T | U

T -> AB | A | B

A -> aA | a

B -> bB | b

U -> ccA | B | cc

S-derivable:

T-derivable:

U-derivable:

Productions removed:

Productions added:

Remove $A \rightarrow S_1 a S_2$

$A \rightarrow S_1 a S_2$

$a \in \sum$, S_1 and S_2 strings, at least one is not empty

Create

$X_a \rightarrow a$

$A \rightarrow S_1 X_a S_2$

Then fix up $A \rightarrow S_1 X_a S_2$

- why? what rule is violated?
- how?

Remove $A \rightarrow S_1X_aS_2$

$A \rightarrow S_1X_aS_2$

$A \rightarrow$

Put in to CNF

$S \rightarrow ASA \mid aB$

$A \rightarrow B \mid S$

$B \rightarrow b \mid \varepsilon$