18.404/6.840 Lecture 4

Last time:

- Finite automata → regular expressions
- Proving languages aren't regular
- Context free grammars

Today:

- Context free grammars (CFGs) – definition
- Context free languages (CFLs)
- Pushdown automata (PDA)
- Converting CFGs to PDAs

- Problem Set 2 will be posted by tomorrow on the homepage.
- Confirm your checkins are recorded on Canvas/grades.
- The 12:00 and 2pm recitations will be in 2-190, but have space for 20 people only.

Context Free Grammars (CFGs)

S 0S1
S R
R

Shorthand:

S 0S1 | R
R

Recall that a CFG has terminals, variables, and rules.

Grammars generate strings

1. Write down start variable
2. Replace any variable according to a rule
   Repeat until only terminals remain
3. Result is the generated string
4. is the language of all generated strings
5. We call a Context Free Language.

Example of generating a string

Tree of
substitutions
"parse tree"

S

S

Resulting
string

CFG – Formal Definition

Defn: A <u>Context Free Grammar</u> (CFG) is a 4-tuple

finite set of variables

finite set of terminal symbols

finite set of rules **(**rule form: **)**

start variable

For write

1) if can go from to with one substitution step in

2) if can go from to with some number of substitution step

is called a derivation of from .

If then it is a <u>derivation</u> of .

and

Defn: is a <u>Context Free Language</u> (CFL) if for some CFG .

Check-in 4.1

Which of these are valid CFGs?

:	B 0B1 |	:	S 0S | S1
	B1 1B		R RR
	0B 0B		

a) only

b) only

c) Both and

d) Neither

Check-in 4.1

CFG – Example

E E+T | T

T T×F | F

F (E) | a

Parse tree

E

E

Resulting string

{E, T, F}
{+, ×, (,), a}
the 6 rules above
E

Generates a+a×a

Observe that the parse tree contains additional informati
such as the precedence of × over + .

If a string has two different parse trees then it is derived a
and we say that the grammar is ambiguous.

Check-in 4.2

How many reasonable distinct meanings does the following English sentence have?

The boy saw the girl with the mirror.

(a) 1

(b) 2

(c) 3 or more

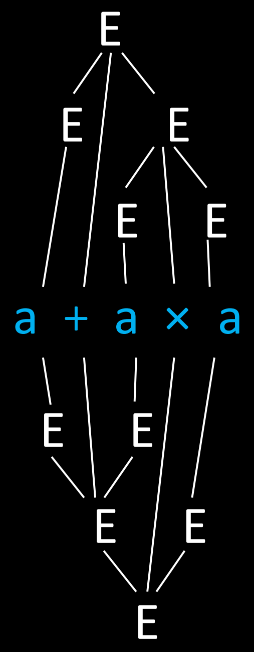
Check-in 4.2

Ambiguity

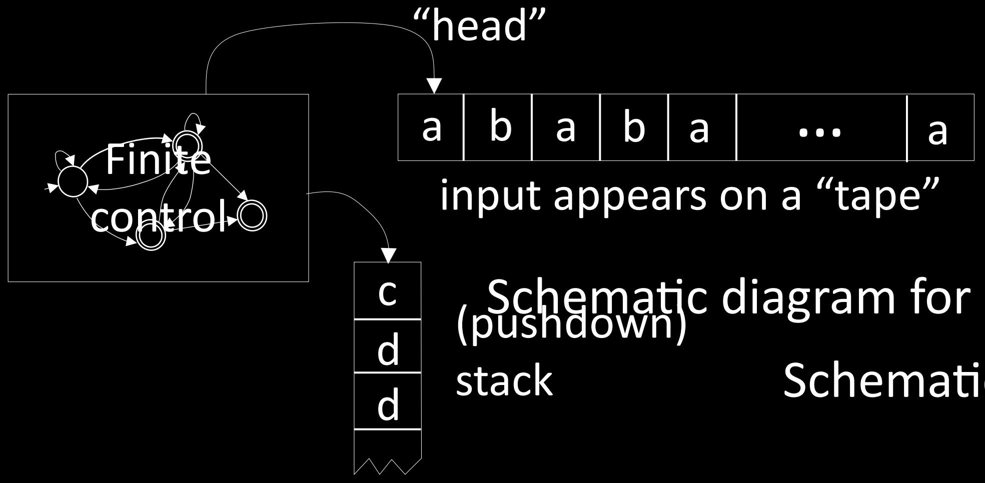
E E+T | T
T T×F | F
F (E) | a

E E+E | E×E | (E) | a

Both and recognize the same language, i.e.,
However is an unambiguous CFG and is ambiguous.



Pushdown Automata (PDA)



Schematic diagram for DFA or NFA

Schematic diagram for PDA

Operates like an NFA except can write-add or read-remove symbols from the top of stack.

push

pop

Example: PDA for

1) Read 0s from input, push onto stack until read 1.
2) Read 1s from input, while popping 0s from stack.
3) Enter accept state if stack is empty. (note: acceptance only at end of input)

Coffee
Break

PDA – Formal Definition

Defn: A Pushdown Automaton (PDA) is a 6-tuple

input alphabet

stack alphabet

Accept if some thread is in the accept state at the end of the input string.

Example: PDA for

Sample input:

0	1	1	1	1	0

1) Read and push input symbols.
   Nondeterministically either repeat or go to (2).
2) Read input symbols and pop stack symbols, compare.
   If ever then thread rejects.
3) Enter accept state if stack is empty. (do in “software”)

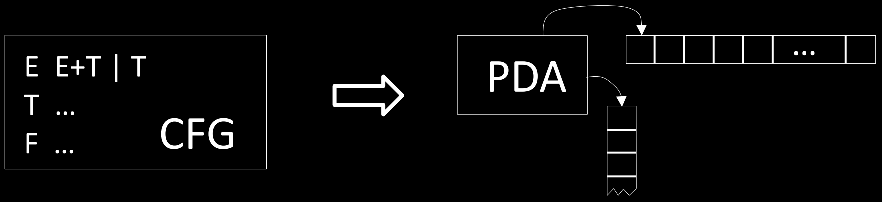
The nondeterministic forks replicate the stack.

This language requires nondeterminism.
Our PDA model is nondeterministic.

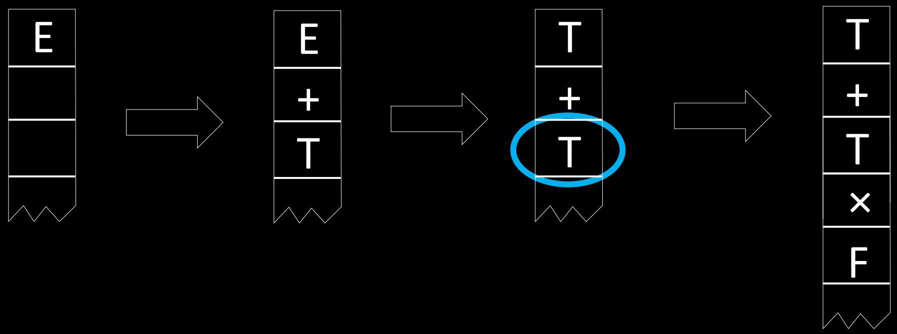
Converting CFGs to PDAs

Theorem: If is a CFL then some PDA recognizes

Proof: Convert ’s CFG to a PDA



IDEA: PDA begins with starting variable and guesses substitutions.
It keeps intermediate generated strings on stack. When done, compare with input.

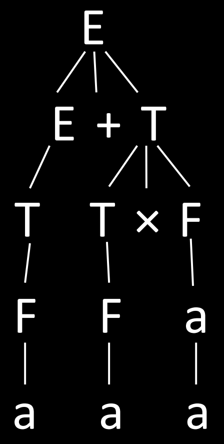


Input:

a	+	a	×	a

E E+T | T
T T×F | F
F (E) | a

E
E+T
T+T×F
F+F×a
a+a×a



Problem! Access below the top of stack is cheating!

Instead, only substitute variables when on the top of stack.

If a terminal is on the top of stack, pop it and compare with input. Reject if .

Converting CFGs to PDAs (contd)

E E+T | T

T T×F | F

F (E) | a

Theorem: If is a CFL then some PDA recognizes

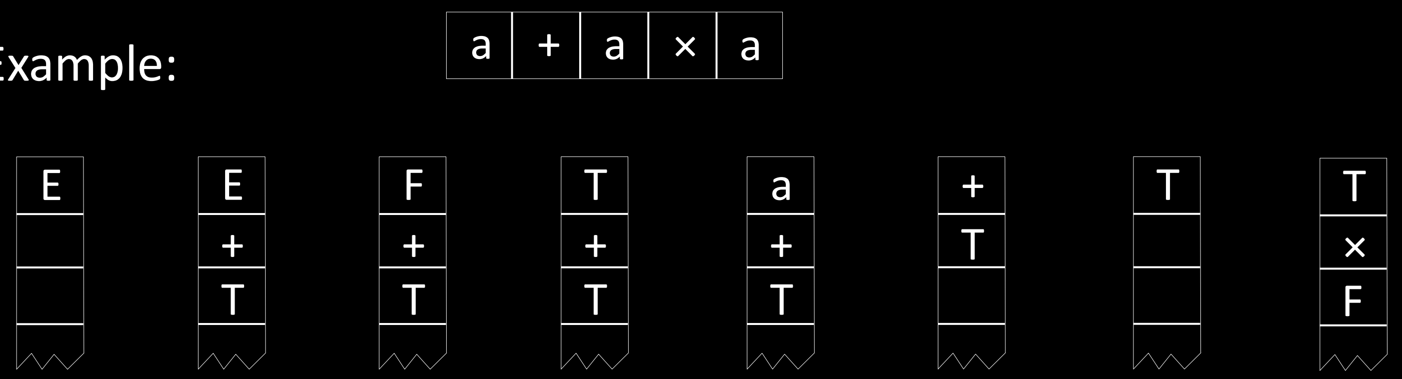
Proof construction: Convert the CFG for to the following PDA.

1) Push the start symbol on the stack.
2) If the top of stack is

   Variable: replace with right hand side of rule (nondet choice).

   Terminal: pop it and match with next input symbol.
3) If the stack is empty, *accept.*

Example:



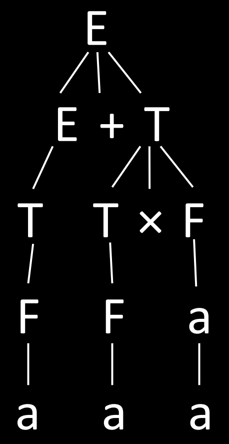
E

E+T

T+T×F

F+F×a

a+a×a



Equivalence of CFGs and PDAs

Theorem: is a CFL iff* some PDA recognizes

Done: ⟷

In book. You are responsible for knowing it is true, but not for knowing the proof.

* “iff” = “if an only if” means the implication goes both ways.
So we need to prove both directions: forward () and reverse ().

Check-in 4.3

Is every Regular Language also a Context Free Language?

(a) Yes

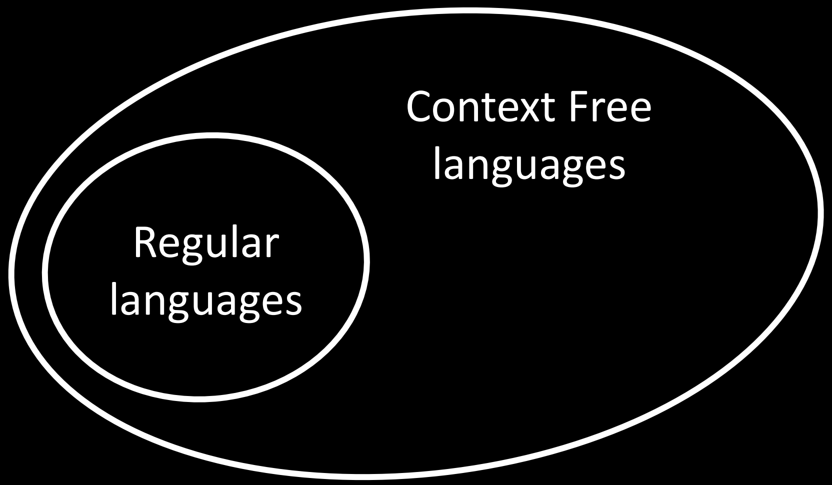
(b) No

(c) Not sure

Check-in 4.3

Recap

	Recognizer	Generator
Regular language	DFA or NFA	Regular expression
Context Free language	PDA	Context Free Grammar



Quick review of today

1. Defined Context Free Grammars (CFGs) and Context Free Languages (CFLs)
2. Defined Pushdown Automata(PDAs)
3. Gave conversion of CFGs to PDAs.