CSE 105
THEORY OF COMPUTATION

Spring 2018

http://cseweb.ucsd.edu/classes/sp18/cse105-ab/

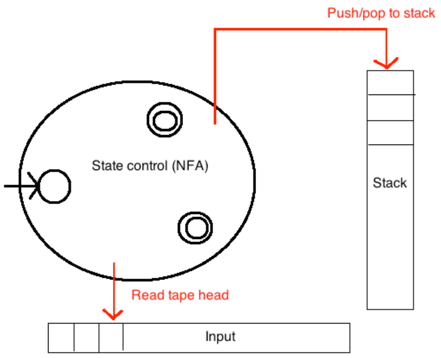
Today's learning goals

Sipser Section 2.1

- Identify the components of a formal definition of a CFG
- Derive strings in the language of a given CFG
- Determine the language of a given CFG
- Design a CFG generating a given language

PDA: NFA as ??: RegExp

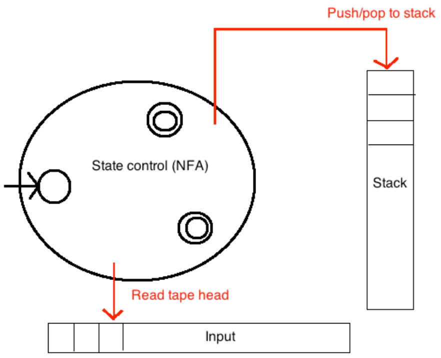
- Automata
  - String is read by the machine one character at a time, from left to right.
  - Determine if computation is successful by checking if entire string was read, and if land at an accept state.



Note: PDA based on **N**FA; can't always be determinized.

PDA: NFA as ??: RegExp

- Automata
  - String is read by the machine one character at a time, from left to right.
  - Determine if computation is successful by checking if entire string was read, and if land at an accept state.
- Regular expressions and ??
  - Derive all strings in the language by following rules for required patterns.



Context-free grammar

Informally, a collection of rules used to *create* string.

CFGs *generate* languages.

Some sample rules:

$$S \to aTb$$
$$T \to aT$$
$$T \to bTS$$
$$S \to \varepsilon$$

More formally...

Context-free grammar

Sipser Def 2.2, page 102

(V, Σ, R, S)

Variables: finite set of (usually upper case) variables **V**

Terminals: finite set of alphabet symbols **Σ**

$V \cap \Sigma = \emptyset$

Rules/Productions: finite set of allowed transformations **R**

$A \to u$ $\quad A \in V, u \in (V \cup \Sigma)^*$

Start variable: origination of each derivation **S**

Derivation

Set of variables

Set of terminals

$G = (\{S\}, \{0\}, R, S)$

with the rules

$R = \{S \rightarrow 0S, S \rightarrow 0\}$

Sample derivation:

$S \Rightarrow 0S \Rightarrow 00S \Rightarrow 000$

Start variable

One-step application of rule

String of terminals

Context-free language

Sipser p. 104

The **language generated by a CFG** (V, Σ, R, S) is

{ w in Σ^* | starting with the Start variable and applying sequence of rules, can derive w on RHS}

If G = (V, Σ, R, S) the language generated by G is denoted L(G).

Notation:

$$S \Longrightarrow {}^* w$$

Terminology: sequence of rule applications is **derivation**

Context-free language

Sipser p. 104

The language generated by CFG (V, Σ, R, S) is

{ w in Σ^* | starting with the Start variable and applying sequence of rules, can derive w on RHS}.

What is the language generated by the CFG ({S}, {0}, R, S) with the rules R = {S → 0S, S → 0} ?

A. {0}
B. {0, 0S}
C. {0, 00, 000, …}
D. {ε, 0, 00, 000, …}
E. I don't know.

Context-free language

Sipser p. 104

What is the language of the CFG ({S}, {0,1}, R, S) with rules

$$S \rightarrow 0S$$
$$S \rightarrow 1S$$
$$S \rightarrow \varepsilon$$

$S \rightarrow 0S \mid 1S \mid \varepsilon$

A. $L(0^*1^*)$
B. $L(0^* \cup 1^*)$
C. $L((0 \cup 1)^*)$
D. $L((0^*1^*)^*)$
E. I don't know.

Designing a CFG

$L = \{ abba \}$

Which CFG generates L?

A. ({S, T, V, W} , {a, b}, {S → aT , T → bV , V → bW , W → a }, S)
B. ({Q}, {a,b}, { Q → abba } , Q)
C. ({X, Y} , {a,b}, { X → aYa , Y → bb } , X)
D. All of the above
E. None of the above

Context-free languages

- L (00*)
- L((0U1)*)
- { abba }

Is any nonregular set context-free?

What about the languages that are recognized by PDAs?

Designing a CFG

We know this set is not regular!

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

Designing a CFG

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

One approach:

- what is shortest string in the language?
- how do we go from shorter strings to longer ones?

Context-free languages

- L (00*)
- L((0U1)*)
- { abba }
- $\{ a^n b^n \mid n \geq 0 \}$

Ex: $\{ 0^i 1^j \mid j \geq i \geq 0 \}$ recognizable by a PDA

PDAs and CFGs are equally expressive

Theorem 2.20: A language is context-free if and only if some nondeterministic PDA recognizes it.

Consequences

- Quick proof that every regular language is context free
- To prove closure of class of CFLs under a given operation, can choose two modes of proof (via CFGs or PDAs) depending on which is easier