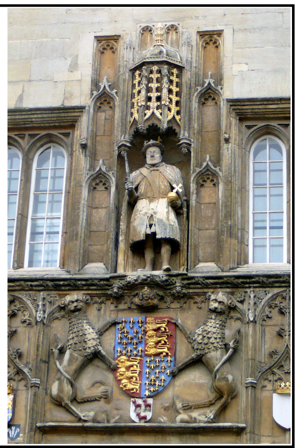
Lecture 10: Context-Free Languages Contextually

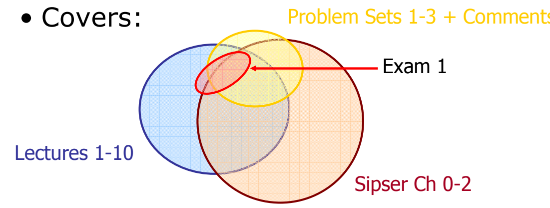


David Evans
http://www.cs.virginia.edu/evans

cs302: Theory of Computation
University of Virginia
Computer Science

Exam 1

- In class, next Thursday, Feb 28
- Covers:



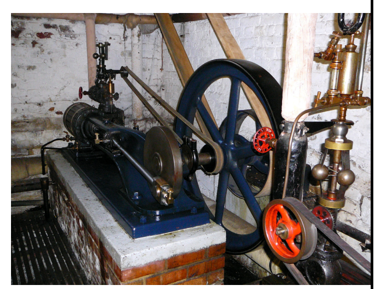
Note: unlike nearly all other sets we draw in this class, all of these sets are finite, and the size represents the relative size.

Exam 1 Notesheet

- For Exam 1, you may not use anything other than
  - Your own brain and body
  - A single page (one side) of notes that you create
- You can work with others to create your notes page

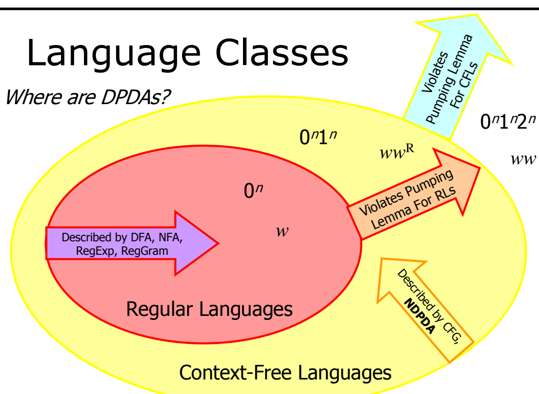
Menu

- Are DPDAs equivalent to NDPDAs?
- Properties of CFLs
- Equivalence of CFGs and NDPDAs



Language Classes

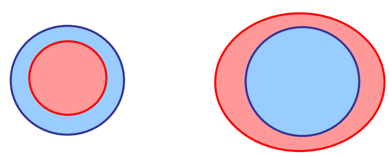
Where are DPDAs?



Proving Set Equivalence

$$A = B \Leftrightarrow A \subseteq B \text{ and } B \supseteq A$$

Sets A and B are equivalent if A is a subset of B and B is a subset of A.



Proving Formalism Equivalence

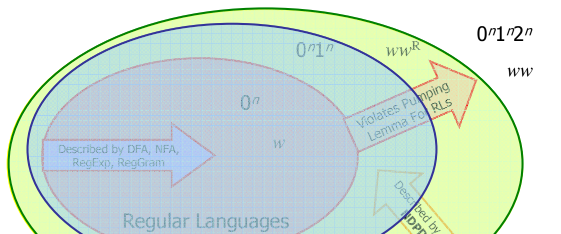
$LR(M)$ = the set of languages that can be recognized by some M
= $\{ l \mid l \in P(\Sigma^*)$ and there is some $m \in M$ such that $L(m) = l \}$

$A = B \Leftrightarrow LR(A) \subseteq LR(B)$ and $LR(B) \supseteq LR(A)$

Proving Formalism **Non**-Equivalence

$LR(M)$ = the set of languages that can be recognized by some M
= $\{ l \mid l \in P(\Sigma^*)$ and there is some $m \in M$ such that $L(m) = l \}$

$A \neq B \Leftrightarrow$ There is an $l \in P(\Sigma^*)$ that is in $LR(A)$ but not in $LR(B)$ or there is an l in LR(B) but not in $LR(A)$



To eliminate **=**, we need to find some language L that can be recognized by an NDPDA and prove it cannot be recognized by a DPDA

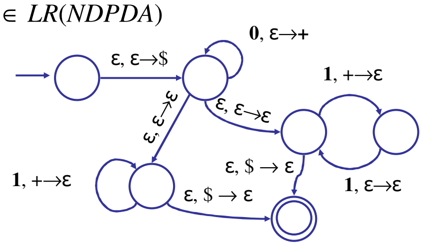
Sensible option 1: LR(NDPDA) $\supset$ LR(DPDA) $\supset$ LR(NFA) = LR(DFA)

~~Sensible option 2: LR(NDPDA) = LR(DPDA) $\supset$ LR(NFA) = LR(DFA)~~

$LR(\text{NDPDA}) \supset LR(\text{DPDA})$

$A = \{ 0^i1^j \mid i \geq 0, j = i \text{ or } j = 2i \}$

$A \in LR(NDPDA)$

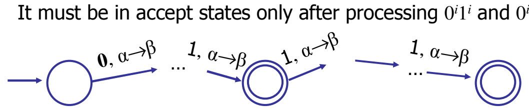


$LR(\text{NDPDA}) \supset LR(\text{DPDA})$

$A = \{ 0^i1^j \mid i \geq 0, j = i \text{ or } j = 2i \}$

$A \notin LR(DPDA)$

Proof by contradiction.
Suppose there is a DPDA P that recognizes A.
It must be in accept states only after processing 0^i1^i and 0^i1^{2i}



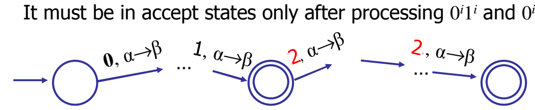
$2i$ transitions, consuming 0^i1^i i transitions, consuming 1^i

$LR(\text{NDPDA}) \supset LR(\text{DPDA})$

$A = \{ 0^i1^j \mid i \geq 0, j = i \text{ or } j = 2i \}$

$A \notin LR(DPDA)$

Proof by contradiction.
Suppose there is a DPDA P that recognizes A.
It must be in accept states only after processing 0^i1^i and 0^i1^{2i}



$2i$ transitions, consuming 0^i1^i i transitions, consuming 2^i

$L(P') = \{ 0^i1^i2^i \mid i \geq 0 \}$

$LR(\text{NDPDA}) \supset LR(\text{DPDA})$

$A = \{ 0^i 1^j \mid i \geq 0, j = i \text{ or } j = 2i \}$

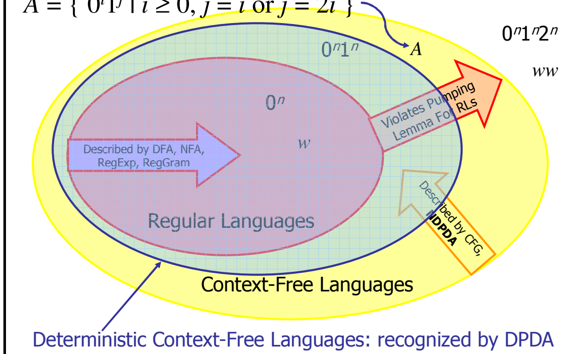
$A \notin LR(DPDA)$

Proof by contradiction. If there is a DPDA P that recognizes A, we could construct a DPDA P' that recognizes

$$A' = L(P') = \{ 0^i 1^i 2^i \mid i \geq 0 \}$$

But, we know A' is not a CFL! (Prove using pumping lemma)
So, there is no NDPDA that can recognize A', so there is no DPDA that can recognize A', so P' must not exist.
Hence, P must not exist. This means there is no DPDA that can recognize A.

$A = \{ 0^i 1^j \mid i \geq 0, j = i \text{ or } j = 2i \}$



$LR(\text{NDPDA}) \supset LR(\text{DPDA}) \supset LR(\text{NFA}) = LR(\text{DFA})$

Closure Properties of RLs

If A and B are regular languages then:

A^R is a regular language
  Construct the reverse NFA

A^* is a regular language
  Add a transition from accept states to start

$\overline{A}$ is a regular language (complement)
  $F' = Q - F$

$A \cup B$ is a regular language
  Construct an NFA that combines DFAs

$A \cap B$ is a regular language
  Construct an DFA combining DFAs that accepts if both accept

Closure Properties of CFLs

If A and B are *context free* languages then:

A^R is a context-free language ?

A^* is a context-free language ?

$\overline{A}$ is a context-free language (complement)?

$A \cup B$ is a context-free language ?

$A \cap B$ is a context-free language ?

Some of these are true. Some of them are false.

CFLs Closed Under Reverse

Given a CFL A, is A^R a CFL?

Since A is a CFL, there is some CFG G that recognizes A.

Proof-by-construction:
There is a CFG G^R that recognizes A^R.

$G = (V, \Sigma, R, S)$

$G^R = (V, \Sigma, R^R, S)$
$R^R = \{ A \to \alpha^R \mid A \to \alpha \in R \}$

CFLs Closed Under *

Given a CFL A, is A^* a CFL?

Since A is a CFL, there is some CFG G that recognizes A.

Proof-by-construction:
There is a CFG G^* that recognizes A^*.

$G = (V, \Sigma, R, S)$

$G^* = (V \cup \{S_0\}, \Sigma, R^*, S_0)$
$R^* = R \cup \{ S_0 \to S \} \cup \{ S_0 \to S_0 S_0 \} \cup \{ S_0 \to \varepsilon \}$

Closure Properties of CFLs

If A and B are *context free* languages then:

A^R is a context-free language **TRUE**

A^* is a context-free language **TRUE**

$\overline{A}$ is a context-free language (complement)?

$A \cup B$ is a context-free language ?

$A \cap B$ is a context-free language ?

CFLs Closed Under Union

Given two CFLs A and B is $A \cup B$ a CFL?

Proof-by-construction:
There is a CFG G_{AUB} that recognizes $A \cup B$.
Since A and B are CFLs, there are CFGs $G_A = (V_A, \Sigma_A, R_A, S_A)$ and $G_B = (V_B, \Sigma_B, R_B, S_B)$ that generate A and B.

$$G_{AUB} = (V_A \cup V_B, \Sigma_A \cup \Sigma_B, R_{AUB}, S_0)$$
$$R_{AUB} = R_A \cup R_B \cup \{ S_0 \rightarrow S_A \} \cup \{ S_0 \rightarrow S_B \}$$

Assumes V_A and V_B are disjoint (easy to arrange this by changing variable names.)

Closure Properties of CFLs

If A and B are *context free* languages then:

A^R is a context-free language **TRUE**

A^* is a context-free language **TRUE**

$\overline{A}$ is a context-free language (complement)?

$A \cup B$ is a context-free language **TRUE**

$A \cap B$ is a context-free language ?

CFLs Closed Under Complement?

- Try to find a counter-example

$\{\mathbf{0}^i\mathbf{1}^i \mid i \geq 0\}$ is a CFL.

Is its complement?

Yes. We can make a DPDA that recognizes it: swap accepting states of DPDA that recognizes $\mathbf{0}^i\mathbf{1}^i$.

Not a counterexample…but not a proof either.

Complementing Non-CFLs

$\{ww \mid w \in \Sigma^*\}$ is **not** a CFL.

Is its complement?

Yes. This CFG recognizes is:

~~$S \rightarrow \mathbf{0}S\mathbf{0} \mid \mathbf{1}S\mathbf{1} \mid \mathbf{0}X\mathbf{1} \mid$~~

~~$X \rightarrow \mathbf{0}X\mathbf{0} \mid \mathbf{1}X\mathbf{1} \mid \mathbf{0}X\mathbf{1}$~~

Opps! As Danni pointed out in class, this is badly broken. We will fix it next class…

So, we have found a pair: P, $\overline{P}$ where one is a CFL and the other is not. Thus, CFLs are **not** closed under complement.

Closure Properties of CFLs

If A and B are *context free* languages then:

A^R is a context-free language **TRUE**

A^* is a context-free language **TRUE**

$\overline{A}$ is **not necessarily** a context-free language (complement)

$A \cup B$ is a context-free language **TRUE**

$A \cap B$ is a context-free language ?

Left for you to solve (possibly on Exam 1)

Charge

- Thursday and Tuesday: some interesting applications of CFGs