CFL Big Picture

Context Free Languages Conclusion

- We have studied the class of context free languages (CFL)
- We saw two different ways to express a CFL
  1. Context Free Grammar
  2. Push Down Automata

- We showed that some were equally expressive
  - We need non-deterministic PDA to express Context Free Grammars
  - Recall the construction of the PDA had only one state, and possible several transitions on the same Non-terminal.
- Some were easier to use than others to describe some languages

Acceptance

- Context free grammars

  The *language of the CFG* , G, is the set

  L(G) = {w$\in$T* | S $\Rightarrow^*$ w}   where

  S is the start symbol of G

  $\Rightarrow$ is the single step relation between derivations

- Push down automata
  - Use of instantaneous descriptions (IDs) and the relation |- between IDs
  - Acceptance by final state
  - Acceptance by empty stack

Algorithms

- We studied algorithms to transform one description into another
  1. Context Free Grammar to PDA (Theorem 2.21 pg 115)
  2. PDA into Context Free Grammar (Lemma 2.27 pg 119)
- We studied how to transform grammars
  1. To remove ambiguity (layering)
     1. Non-ambiguous languages can have ambiguous grammars
  2. To transform into Chomsky Normal Form

Properties

- We saw that Regular Languages have many properties

- Closure properties
  - Union
  - Kleene – star
  - Intersection
  - Complement
  - Reversal
  - Difference
  - Prefix

CFL Languages have fewer properties

- Closure properties
  - Union
  - Kleene – star
  - Concat

- But we do have the intersection between CFL and RL produces a CFL

Closure Properties of CFL's

- The class of context-free languages is closed under these three operations: Union, Concatenation, Kleene Star

- Assumptions:
- Let $G_1=(V_1,T_1,P_1,S_1)$ and $G_2=(V_2,T_2,P_2,S_2)$
- be two CF grammars. Assume the sets of variables, V_1 and V_2 are disjoint.

Unio
n

- A grammar for the union $L(G_1) \cup L(G_2)$ is
- $G=(\{S\} \cup V_1 \cup V_2, T_1 \cup T_2, P, S)$
- where P consists of productions in P_1 and P_2 together with $S \rightarrow S_1 \mid S_2$

Concatenation

- A grammar for the concatenation $L(G_1)L(G_2)$ is
- $G=(\{S\} \cup V_1 \cup V_2, T_1 \cup T_2, P, S)$
- where P consists of productions in
- P_1 and P_2 together with $S \rightarrow S_1S_2$.

Kleene Star

- A grammar for $L(G_1)^*$ is
- $G=(\{S\} \cup V_1, T_1, P, S)$
- where P consists of productions in P_1 together with $S \rightarrow \Lambda \mid SS_1$
- qed

Negative result for Complement, Intersection

- The class of context-free languages is *not* closed under these two operations: Complement, Intersection

- **Proof.** The language
- $L_1 = \{a^i b^i c^j \mid i,j \geq 0\} = \{ a^i b^i \mid i \geq 0\} \bullet c^*$
- being the concatenation of two CFL's is CFL itself.
- Similarly, $L_2 = \{ a^j b^i c^i \mid i,j \geq 0 \}$ is a CFL.
- However, $L_1 \cap L_2 = \{a^i b^i c^i \mid i \geq 0\}$ is not a CFL, as we saw last time.

- Since the intersection can be expressed in terms of union and complementation $A \cap B = Comp(Comp(A) \cup Comp(B))$, it follows that the class of CFL's is not closed under complementation.

Mixtures of CFL and RE

- **Theorem**. Intersection of any context-free language with any regular language is context-free.

- *Proof Idea*. Product construction. Take a PDA for the first language and a DFA for the second. Construct a PDA for the intersection by taking for its states the set of all pairs of states of the first two automata. Etc.
- qed

- Note that there is no sensible definition of the product of two PDA's: we cannot combine two stacks into one.

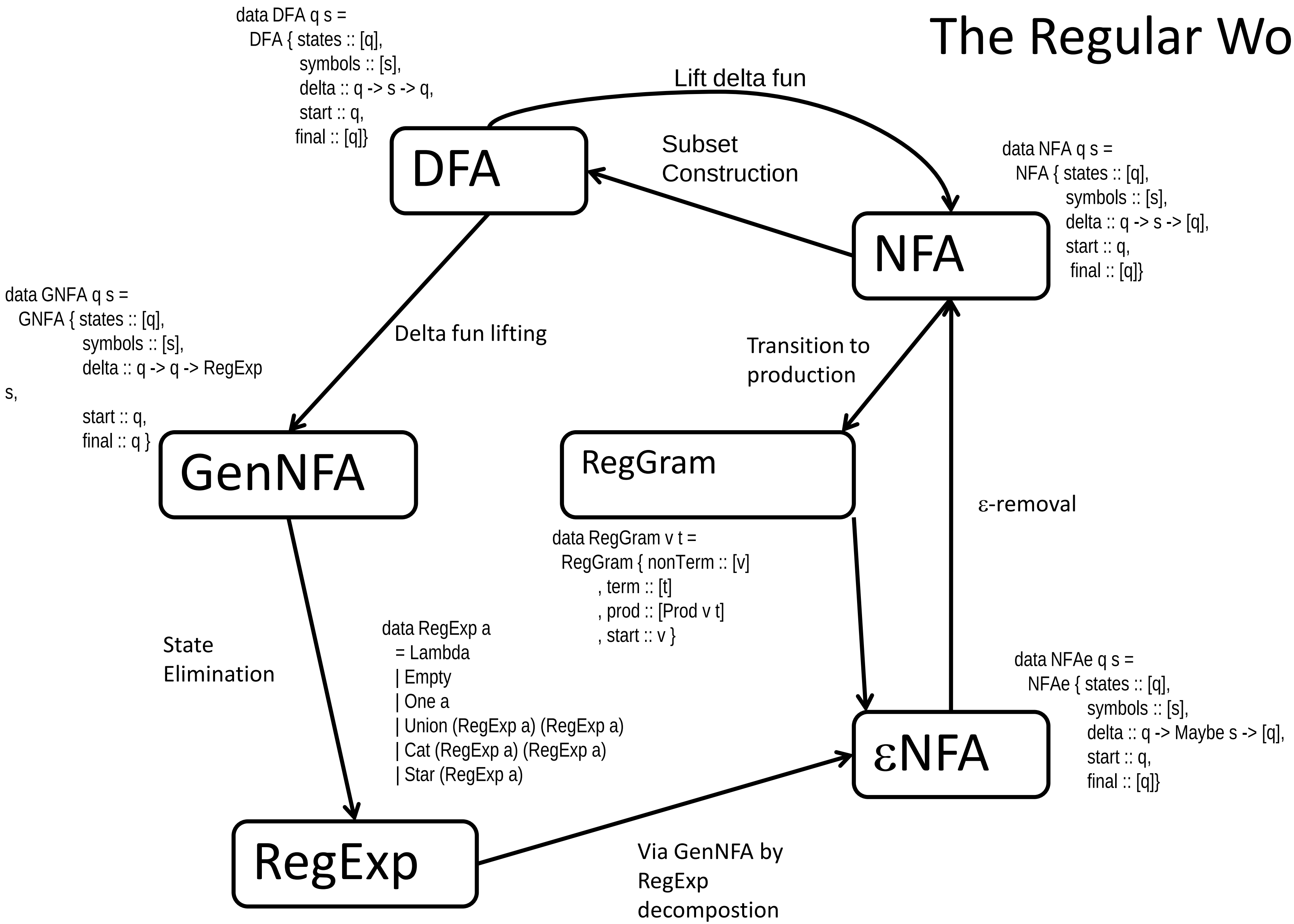
Proving some language is not CF

- Pumping lemma for CF languages
- Let L be a CFL. Then there exists a number n (depending on L) such that every string w in L of length greater than n contains a CFL pump.

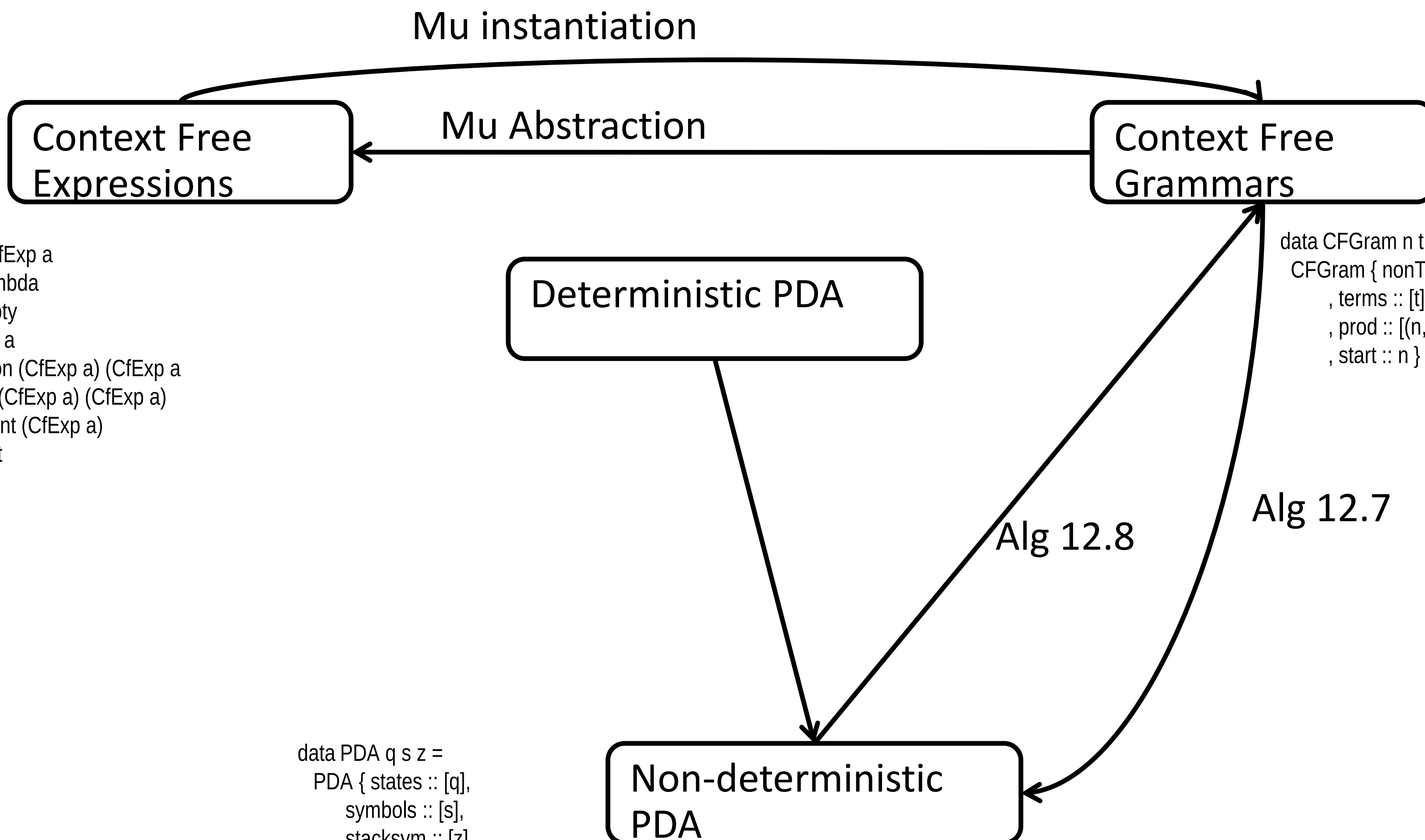
Context Free Pump

- A *CFL pump* consists of two non-overlapping substrings that can be pumped simultaneously while staying in the language.
- Precisely, two substrings u and v constitute a CFL pump for a string w of L ($|w| > m$) when
  1. $uv \neq \Lambda$ (which means that at least one of u or v is not empty)
  2. And we can write $w=xuyvz$, so that for every $i \geq 0$
  3. $xu^iyv^iz \in L$

The Regular World



The Context Free World




```
data CfExp a
  = Lambda
  | Empty
  | One a
  | Union (CfExp a) (CfExp a
  | Cat (CfExp a) (CfExp a)
  | Mu Int (CfExp a)
  | V Int
```

```
data CFGram n t =
  CFGram { nonTerm :: [n]
         , terms :: [t]
         , prod :: [(n,[Sym n t])]
         , start :: n }
```

```
data PDA q s z =
  PDA { states :: [q],
        symbols :: [s],
        stacksym :: [z],
        delta ::  [(q,Maybe s,z,[(q,[z])])],
        start :: q,
        final :: [q]}
```

The Larger World