

CFL Big Picture

Context Free Languages Conclusion

- We have studied the class of context free languages (CFL)
- We saw many different ways to express a CFL
 1. Context Free Grammars
 2. Context Free Expressions. Things like $(\mu x . a x a)$
 3. 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

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

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 \vdash 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 (Alg 12.7 pg 770)
 2. PDA into Context Free Grammar (Alg 12.8 pp771-772)
- We studied how to transform grammars
 1. To remove ambiguity (layering)
 1. Non-ambiguous languages can have ambiguous grammars
 2. Some languages are inherently ambiguous.
 2. To remove left recursion by factoring

Parsing

- We studied how to accept CFL by using parsing methods based upon context free grammars
 - Top down methods - LL(1)
 - Recursive descent
 - Predictive parsers
 - Bottom up methods – LR(1)

Properties

- We saw that Regular Languages have many properties
- Closure properties
 - Union
 - Kleene – star
 - Intersection
 - Complement
 - Reversal
 - Difference

CFL have fewer properties

- Closure properties
 - Union
 - Kleene – star
 - Concat
- But we do have the intersection between CFL and RL produces a CFL

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. $xu^i y v^i z \in L$

The Regular World

```
data DFA q s =
  DFA { states :: [q],
        symbols :: [s],
        delta :: q -> s -> q,
        start :: q,
        final :: [q] }
```

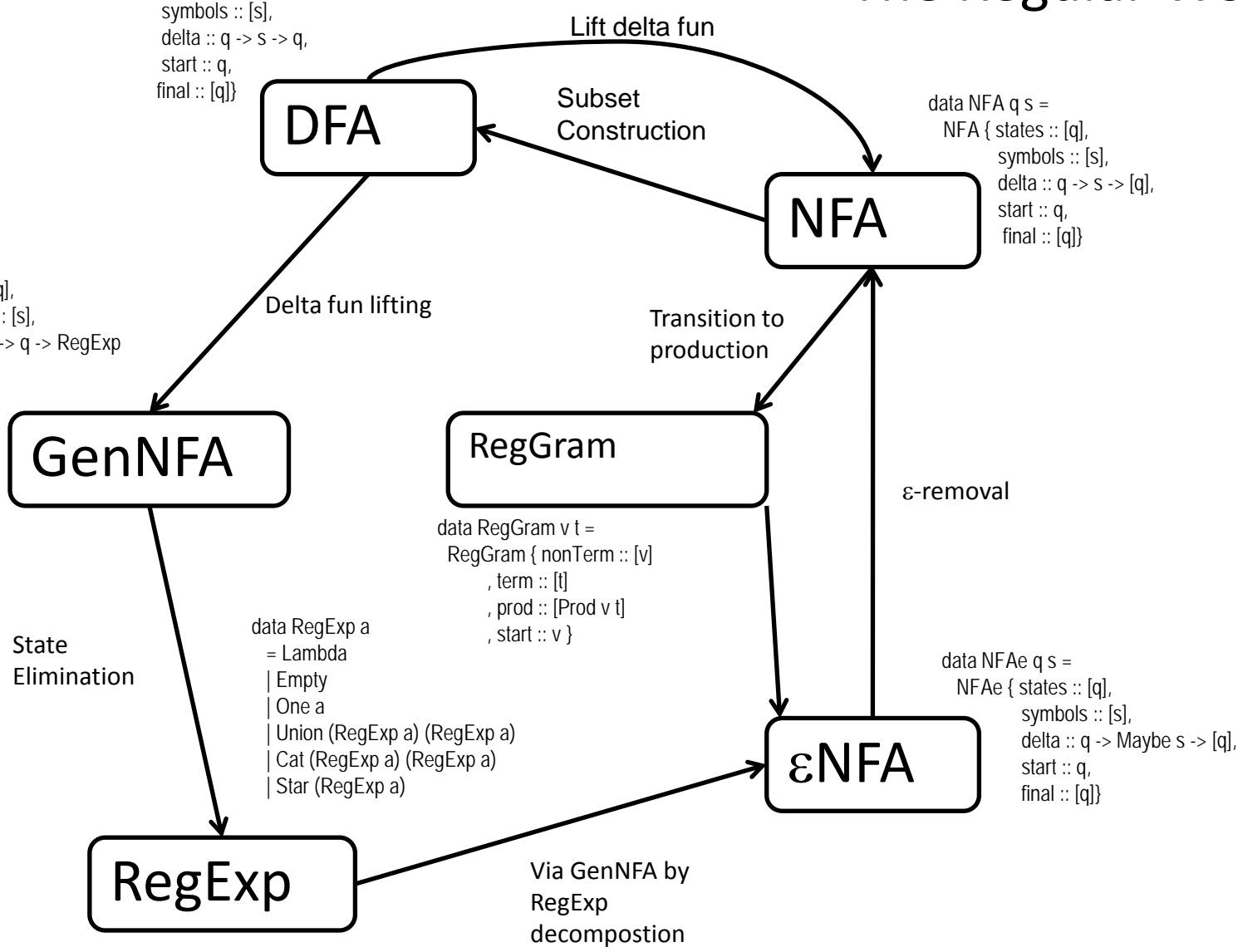
```
data NFA q s =
  NFA { states :: [q],
        symbols :: [s],
        delta :: q -> s -> [q],
        start :: q,
        final :: [q] }
```

```
data GNFA q s =
  GNFA { states :: [q],
         symbols :: [s],
         delta :: q -> q -> RegExp,
         start :: q,
         final :: [q] }
```

```
data RegGram v t =
  RegGram { nonTerm :: [v],
            term :: [t],
            prod :: [Prod v t],
            start :: v }
```

```
data RegExp a =
  Lambda
  | Empty
  | One a
  | Union (RegExp a) (RegExp a)
  | Cat (RegExp a) (RegExp a)
  | Star (RegExp a)
```

```
data NFAs q s =
  NFAs { states :: [q],
         symbols :: [s],
         delta :: q -> Maybe s -> [q],
         start :: q,
         final :: [q] }
```



The Context Free World

Mu instantiation

Mu Abstraction

Context Free Expressions

Context Free Grammars

Deterministic PDA

Non-deterministic PDA

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

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

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

Alg 12.8

Alg 12.7

The Larger World

