Regular Language Topics (Section 11.4)

Regular Grammars

- In Section 3.3 we talked about grammars. The grammars we talked about are capable of representing regular languages, as well as other kinds of languages. By adding some additional constraints on our grammars, we can limit them to only defining regular languages. Grammars with these limitations are called regular grammars.
- A regular grammar is a grammar whose productions take the following form, where w is a string of terminals:
  - $A \to wB$
  - $A \to w$
- *Example:* A regular grammar for the language $a^* b^*$ is:
  - $S \to \Lambda | aS | T$
  - $T \to b | bT$

Example

- Write a regular grammar for $\{ab, acb, accb, \ldots, ac^n b, \ldots\}$.

Example

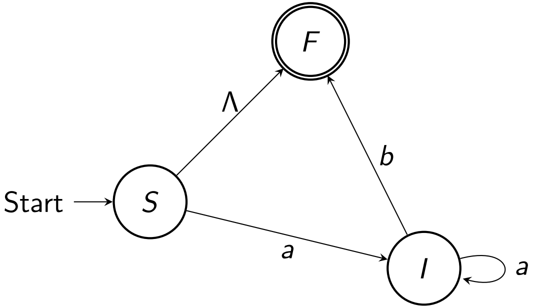
- Write a regular grammar for $\{ab, acb, accb, \ldots, ac^nb, \ldots\}$.
- *Solution:* A regular expression for the language is ac^*b. A regular grammar is:
    - $S \to aT$
    - $T \to b|cT$

NFA to Regular Grammar

1. State names become the nonterminals. So rename them to uppercase letters.
2. The start state becomes the start symbol of the grammar.
3. For each state transition from I to J labeled with x construct a production $I \to xJ$.
4. For each final state F contstruct a production $F \to \Lambda$.

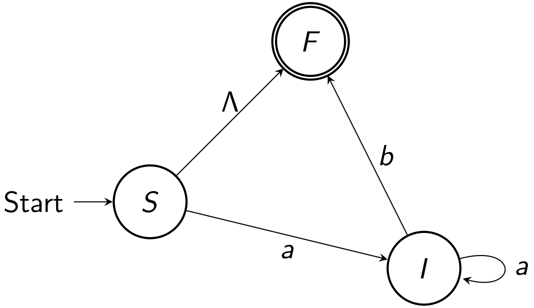
Example

Transform the following NFA into a regular grammar:



Example

Transform the following NFA into a regular grammar:



Solution:

- $S \rightarrow aI | F$
- $I \rightarrow aI | bF$
- $F \rightarrow \Lambda$

Simplification

The previous grammar can be simplified to:

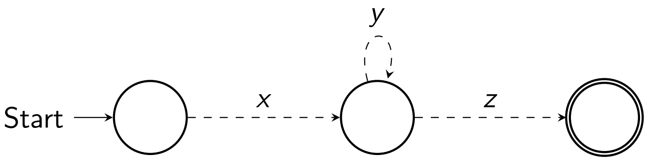
- $S \to aI|\Lambda$
- $I \to aI|b$

Opposite direction

There is also a straightforward procedure for transforming a regular grammar to an NFA. We are not going to discuss it.

Properties of Regular Languages

- Using some properties of regular languages will help us to argue that certain languages are not regular.
- The Pumping Lemma: If L is an infinite regular language, then it is recognized by a DFA with, say, m states. If s in L and $|s| \geq m$, then an acceptance path for s must pass through some state twice.



Pumping Lemma continued

- The dotted arrows represent the path to acceptance for s and the letters x, y, and z represent the concatenation of the letters along the edges of the path. So $s = xyz$ and $y \neq \Lambda$. Assume that the middle state is the first repeated state on the path. So $|xy| \leq m$. Since the loop can be traversed any number of times, we have the *pumping property:* $xy^k z \in L$ for all $k \in \mathbb{N}$.

Pumping Lemma example

- *Example:* The language $L = \{a^n b^n | n \in \mathbb{N}\}$ is not regular. (This is probably the canonical example of a non-regular language.)
- *Proof:* Assume, BWOC (by way of contradiction), that L is regular. Since L is infinite, the pumping lemma applies. Choose $s = a^m b^m$. Then $s = xyz$, where $y \neq \Lambda, |xy| \leq m$, and $xy^k z \in L$, for all $k \in \mathbb{N}$. Since $|xy| \leq m$ and $s = a^m b^m = xyz$, it follows that xy is a string of a's. Since $y \neq \Lambda$, it must be that $y = a^i$ for some $i > 0$. We'll try for a contradiction with $k = 2$. The pumping property implies $xy^2 z \in L$. But $xy^2 z = a^{m+i} b^m$, which is not in L because $i > 0$. This contradiction implies that L is not regular. QED.