Prefix DFA

Prefix DFA

Overview

Constructing Complex FA

=
Correctness of Proof

If $A$ accepts any string, there must be a path from the start to an accept state in $M$. That same path is in $N$, so $N$ must accept $\varepsilon$.

If $A$ does not accept any string, there is no path from start to an accept state in $M$. So there is no path in $N$, so $N$ does not accept anything.

Epsilon Transitions

Let $L = \{ \varepsilon \}$ if $A$ is not empty, and $\emptyset$ otherwise.

We know that either way, $L$ is a regular language - in either case, it is easy to build a DFA for $L$.

Build a machine $N$ for $L$ based on $M$, with the same set of states.

But let's say that we want to build a FA for $L$ based on $N$.

In real, all finite languages are regular.

Let $L$ be a FA that recognizes $A$ is a regular language, and $N$ a DFA that recognizes $\varepsilon$.

$\{ b \in \{ a, b \} \mid \text{there is } a \in \{ a, b \} \text{ s.t. } \delta_M(q, a) = q' \}$

Note that $M$ is now a NFA, but that is fine.
More formally

- State of $P$ is $N$, start state, and accept states are $N$, accept states, and $N$, accept states
- $\delta_P(\cdot, 0) = \{ q' \}$ for all corresponding states $q$ and $q'$. Also has the following transitions:
- $\delta_P$ has all of the transitions from $N$ and $M$
- Call $N$, states $u \ldots v$, and $N$, states $u \ldots w$
- States of $P$ are the union of states of $M$ and $N$ (not cross product)
- $\delta$ function:
  - $f = \{(a \cdot b) \in \delta_N | b \}$
- $g = \{(a \cdot b) \in \delta_M | b \}$
- Construct $N$, similar to $N$, same states, same start state, same accept

Second Way to Show Prefix is Regular

$L = \{ w | w \in A \}$ where $A$ is regular
- Let $A$ be a DFA for $A$
- Make a new machine $P$ that joins $M$ and $N$ together
- Make a copy of $N$, call it $N'$
- Have $N'$'s transition be same as $N$, but replace $\epsilon$ instead of characters
- Add transitions from states of $M$ to corresponding states of $N$ on reading $\epsilon$
- $\delta_P$ also has the following transitions:
- $\delta_P(\cdot, 0) = \{ q' \}$ for all corresponding states $q$ and $q'$. Also has the following transitions
Common Prefix

Let \( M \) be a DFA that recognizes \( A \) and \( N \) be a DFA that recognizes \( B \).

- Build machine \( P \) in the same way we built a DFA that does intersection

  - Turn all non-accept states into accept states

  - Build machine in the same way we built a DFA that does intersection

More formally:

- Create \( P \) as follows
  - \( P \) has union of states of \( M \) and \( N \)
  - Start state of \( P \) is \( M \)'s start state
  - Accept states of \( P \) are \( N \)'s accept states
  - Transitions of \( P \) are union of \( M \)'s transitions and \( N \)'s transitions along with \( \delta(q, \epsilon) \rightarrow q' \) for all of \( q \) in \( M \)'s states and \( q' \) in \( N \)'s states

Splicing

Let \( M \) be a DFA where \( A \) and \( B \) are both regular.

- \( L = \{ x \in \Sigma^* | \exists n \in \mathbb{N} \text{ such that } m \cdot x \in \Sigma^* \} \)