Today's learning goals  Sipser Ch 1.1, 1.2

• Compare properties of regular and NFA-recognizable languages
• Convert an NFA (with or without epsilon transitions) to a DFA recognizing the same language
Differences between NFA and DFA

- **DFA**: unique computation path for each input
- **NFA**: allow several (or zero) alternative computations on same input
  - $\delta(q,x)$ may specify *more than one* possible next states
  - $\epsilon$ transitions allow the machine to *transition between states spontaneously*, without consuming any input symbols
  - computation can *get stuck* at some state, if there's a missing arrow
Formal definition of NFA

A *nondeterministic finite automaton* is a 5-tuple \((Q, \Sigma, \delta, q_0, F)\) where

1. \(Q\) is a finite set called the states
2. \(\Sigma\) is a finite set called the alphabet
3. \(\delta : Q \times \Sigma \varepsilon \rightarrow \mathcal{P}(Q)\) is the transition function
4. \(q_0 \in Q\) is the start state
5. \(F \subseteq Q\) is the set of accept states.

Which piece of the definition of NFA means there might be **more than one** possible next state from a given state, when reading symbol \(x\) from the alphabet?

A. Line 2, the size of \(\Sigma\)
B. Line 3, the domain of \(\delta\)
C. Line 3, the codomain of \(\delta\)
D. Line 5, that \(F\) is a set
E. I don't know.
Simulating NFA with DFA

For any language recognized by an NFA, there is a DFA that recognizes this language.

**Proof:**

**Given** $A$, a language recognized by $N = (Q, \Sigma, \delta, q_0, F)$ a NFA

**WTS** there is some DFA $M$ with $L(M) = A$

**Construction**

**Correctness**

**Conclusion**
From NFA to DFA

- Transform the following NFA into an equivalent DFA
- Idea: Use subsets of \( \{q_0, q_1, q_2, q_3\} \) as states
Subset construction

Given $A$, a language recognized by $N = (Q, \Sigma, \delta, q_0, F)$ a NFA

WTS there is some DFA $M$ with $L(M) = A$

Construction Define $M = (Q', \Sigma, \delta', q_0', F')$ with

- $Q' = \text{the power set of } Q = \{ X \mid X \text{ is a subset of } Q \}$
- $q_0' = \{ q_0 \}$ \textit{(assuming no } \varepsilon \text{-transitions)}
- $F' = \{ X \mid X \text{ is a subset of } Q \text{ and } (X \cap F) \text{ is nonempty} \}$
- $\delta' (\text{ }) = \text{ }$
Subset construction

**Given** A, a language recognized by $N = (Q, \Sigma, \delta, q_0, F)$ a NFA

**WTS** there is some DFA $M$ with $L(M) = A$

**Construction** Define $M = (Q', \Sigma, \delta', q_0', F')$ with

- $Q' = \text{the power set of } Q = \{ X | X \text{ is a subset of } Q \}$
- $q_0' = \{ q_0 \}$ *(assuming no $\varepsilon$-transitions)*
- $F' = \{ X | X \text{ is a subset of } Q \text{ and } (X \cap F) \text{ is nonempty} \}$
- $\delta'(X, x) = \{ q \in Q | q \text{ is in } \delta(r, x) \text{ for some } r \in X \}$
Subset construction example

How big is $Q'$?

A. 2
B. 4
C. 5
D. 16
E. I don't know
What is the initial state $q_0'$?

A. $q_0$
B. $q_3$
C. $\{q_0,q_1,q_2,q_3\}$
D. $\{q_0\}$
E. I don't know
Subset construction example

NFA

DFA
Simulating NFA with DFA

For any language recognized by an NFA, there is a DFA that recognizes this language.

**Proof:**

**Given** A, a language recognized by N = (Q, Σ, δ, q0, F) a NFA

**WTS** there is some DFA M with L(M) = A

**Construction** Define M = (Q', Σ, δ', q0', F') with Q' = P(Q),...

q0'={q0}, δ' ( X, x ) = { q in Q | q is in δ(r,x) for some r in X }

**Correctness** ??

**Conclusion**
Simulating NFA with DFA

For any language recognized by an NFA, there is a DFA that recognizes this language.

Proof:

Given $A$, a language recognized by $N = (Q, \Sigma, \delta, q_0, F)$ a NFA

WTS there is some DFA $M$ with $L(M) = A$

Construction Define $M = (Q', \Sigma, \delta', q_0', F')$ with ...

Correctness ??

Conclusion

Details, with epsilon transitions: Sipser 55-56
Application

A language $A$ over $\Sigma$ is **regular** if and only if

- it is recognized by a DFA
- it is recognized by a NFA

To prove that the class of regular languages is closed under operation .... :

Let $A$ be a regular language, so recognized by DFA $M$. Build a **NFA** that recognizes the result of .... on $A$. Conclude this result is also a regular language.
The regular operations  Sipser Def 1.23 p. 44

For A, B languages over same alphabet, define:

\[ A \cup B = \{ x | x \in A \text{ or } x \in B \} \]

\[ A \circ B = \{ xy | x \in A \text{ and } y \in B \} \]

\[ A^* = \{ x_1x_2\ldots x_k | k \geq 0 \text{ and each } x_i \in A \} \]

How can we prove that the concatenation of two regular languages is a regular language?
For next time

- Read Sipser 1.2:
  NFA $\rightarrow$ DFA transformation including $\varepsilon$-transitions
- *Haskell 1 due tonight.*