Theory of Computation

MIEIC, 2nd Year

João M. P. Cardoso
Email: jmpc@acm.org

Outline

- Non-Deterministic Finite Automata (NFAs)
- Conversion between FAs (from NFA to DFA)

Starting Example

- Draw a DFA to recognize strings over {0,1} with a ‘1’ in the second from last position.

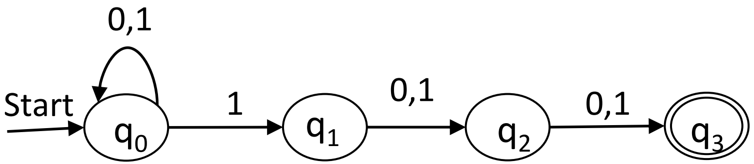
Example 1

- Let's consider a DFA to recognize strings over {0,1} with a '1' in the third from last position.
- even more difficult!

Non-Deterministic Finite Automata (NFAs)

- A Non-Deterministic Finite Automaton (NFA)
  - It can be in more than one state at the same time (we don't know which one, all the possibilities are open)
  - From a state, with an input, it can go to various states
  - In the end, it is enough that one of the states reached be an accept state
- Example 1, now using an NFA

recognize strings over {0,1} with a '1' in the third from last position

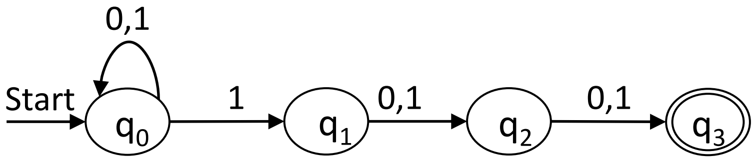


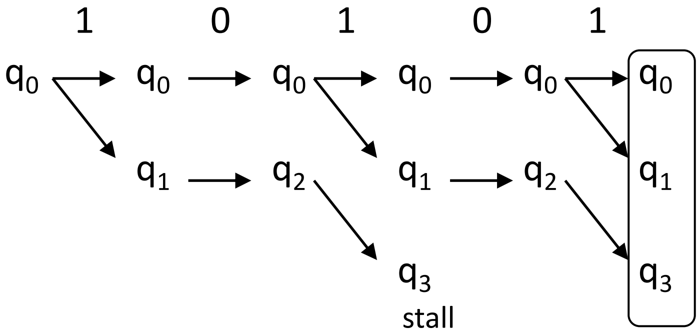
- To opt about the transition to follow in q_0, when arrives a 1, it would be necessary to guess the rest of the input chain

Processing in an NFA

- Considering the input 10101
  - In order to avoid guessing, we analyze all the alternatives in parallel
  - Simpler FA but requiring a higher computing complexity

recognize strings over {0,1} with a '1' in the third from last position





– String of the language because q_3 is an accept state

Definition of an NFA

- NFA A = $(Q, \Sigma, \delta, q_0, F)$
  - Equal to DFA, except that the state transition function δ returns a subset of Q, instead of a single state
- Example 1, NFA
  - A= $(\{q_0, q_1, q_2, q_3\}, \{0,1\}, \delta, q_0, \{q_3\})$
- Transition table
  - Uses sets of states

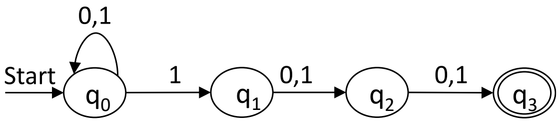
	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0,q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
$*q_3$	$\varnothing$	$\varnothing$

Extended Transition Function $\hat{\delta}$

- New definition inductive in |w| dealing with composable states
  - **Basis**: $\hat{\delta}(q,\varepsilon) = \{q\}$
  - **Induction**: let w=*xa* (*x* is a string and *a* represents a symbol), supposing

  $\hat{\delta}(q,x) = \{p_1, \ldots, p_k\}$ then we have

$$\hat{\delta}(q,w) = \bigcup_{i=1}^{k} \delta(p_i, a) = \{r_1, r_2, \ldots, r_m\}$$



- Example 1, NFA and w=10101:
  - $\hat{\delta}(q_0,1010) = \{q_0, q_2\}$ and $\hat{\delta}(q_0,10101) = \delta(q_0,1) \cup \delta(q_2,1) = \{q_0, q_1, q_3\}$
  - I.e., $\hat{\delta}(q_0,10101) = \{q_0, q_1, q_3\}$

Language of an NFA A:

- $L(A) = \{w \mid \hat{\delta}\,(q_0,w) \cap F \neq \emptyset\}$
- Set of strings w such that $\hat{\delta}(q_0,w)$ contains at least an accept/final state

NFA – DFA Equivalence

- To *convert* an NFA N = (Q_N, Σ , δ_N, q_0, F_N) into a DFA D = (Q_D, Σ , δ_D, {q_0}, F_D) we use the subset construction technique
- Q_D is the set of the subsets of Q_N
  - $Q_D = \wp(Q_N)$
  - If Q_N has *n* states, Q_D has 2^n, but many might be eliminated because they are unreachable
- F_D is the set of the subsets S of Q_N such that $S \cap F_N \neq \emptyset$
- For each $S \subseteq Q_N$ and each $a \in \Sigma$

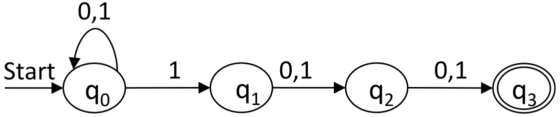
$$\delta_D(S,a) = \bigcup_{p \in S} \delta_N(p,a)$$

Construction of Subsets

NFA:

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0, q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
$*q_3$	$\varnothing$	$\varnothing$

recognize strings over {0,1} with a '1' in the third from last position



Construction of Subsets

NFA:

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0,q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
$*q_3$	$\varnothing$	$\varnothing$

DFA:

	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\rightarrow\{q_0\}$	$\{q_0\}$	$\{q_0,q_1\}$
$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
$\{q_2\}$	$\{q_3\}$	$\{q_3\}$
$*\{q_3\}$	$\varnothing$	$\varnothing$
$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0,q_3\}$	$\{q_0,q_1,q_3\}$
$*\{q_0,q_3\}$	$\{q_0\}$	$\{q_0,q_1\}$

	0	1
$\{q_1,q_2\}$	$\{q_2,q_3\}$	$\{q_2,q_3\}$
$*\{q_1,q_3\}$	$\{q_2\}$	$\{q_2\}$
$*\{q_2,q_3\}$	$\{q_3\}$	$\{q_3\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$*\{q_0,q_1,q_3\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$*\{q_0,q_2,q_3\}$	$\{q_0, q_3\}$	$\{q_0,q_1,q_3\}$
$*\{q_1,q_2,q_3\}$	$\{q_2, q_3\}$	$\{q_2,q_3\}$
$*\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$

State $\varnothing$ (dead state) is essential to guarantee that the resultant DFA all the alphabet symbols have a transition from each of the DFA states.

Construction of Subsets (cont.)

NFA:

	0	1
$\rightarrow q_0$	$\{q_0\}$	$\{q_0,q_1\}$
q_1	$\{q_2\}$	$\{q_2\}$
q_2	$\{q_3\}$	$\{q_3\}$
$*q_3$	$\varnothing$	$\varnothing$

DFA:

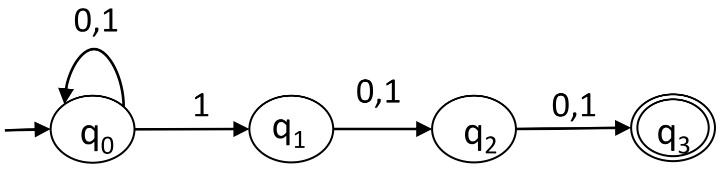
	0	1
$\varnothing$	$\varnothing$	$\varnothing$
$\rightarrow\{q_0\}$	$\{q_0\}$	$\{q_0,q_1\}$
$\{q_1\}$	$\{q_2\}$	$\{q_2\}$
$\{q_2\}$	$\{q_3\}$	$\{q_3\}$
$*\{q_3\}$	$\varnothing$	$\varnothing$
$\{q_0,q_1\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$\{q_0,q_2\}$	$\{q_0,q_3\}$	$\{q_0,q_1,q_3\}$
$*\{q_0,q_3\}$	$\{q_0\}$	$\{q_0,q_1\}$

	0	1
$\{q_1,q_2\}$	$\{q_2,q_3\}$	$\{q_2,q_3\}$
$*\{q_1,q_3\}$	$\{q_2\}$	$\{q_2\}$
$*\{q_2,q_3\}$	$\{q_3\}$	$\{q_3\}$
$\{q_0,q_1,q_2\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$
$*\{q_0,q_1,q_3\}$	$\{q_0,q_2\}$	$\{q_0,q_1,q_2\}$
$*\{q_0,q_2,q_3\}$	$\{q_0, q_3\}$	$\{q_0,q_1,q_3\}$
$*\{q_1,q_2,q_3\}$	$\{q_2, q_3\}$	$\{q_2,q_3\}$
$*\{q_0,q_1,q_2,q_3\}$	$\{q_0,q_2,q_3\}$	$\{q_0,q_1,q_2,q_3\}$

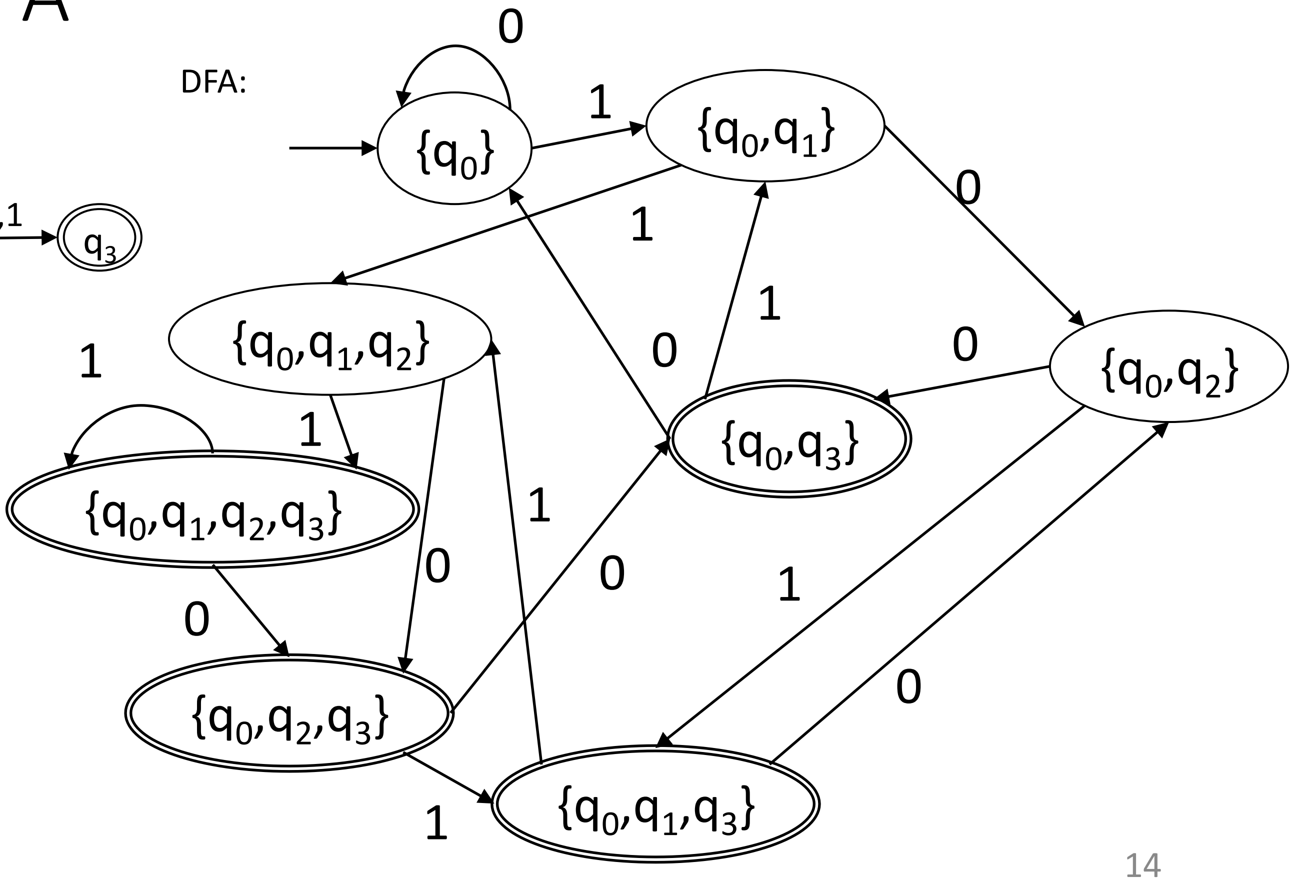
Highlighted states are the reachable states

Equivalent DFA

NFA:



DFA:



NFA to DFA using Subset Construction

- The worst case for the subset construction is when we need an exponential number of states for the DFA:
  - 2^n, including the dead state, for an NFA with n states
- However, in many cases the number of states of the DFA is not too higher than the number of states of the NFA

☞ We can apply the NFA to DFA conversion starting by the start state and considering only the reachable states (*as presented in the white board*)

Dead States

- A dead state is a non-accepting state with self transitions for all the symbols of the alphabet
  - It is used to capture errors in a DFA
  - If the automaton has a maximum of one transition for each state/alphabet symbol, even if it is not complete, can be considered a DFA (sometimes referred as an incomplete DFA): to be a DFA it is only needed to add the dead state (w/ self transitions) to where all the missing transitions will go

Theorem NFA – DFA

- **Theorem**: if $D = (Q_D, \Sigma , \delta_D, \{q_0\}, F_D)$ for a DFA built from NFA $N = (Q_N, \Sigma , \delta_N, q_0, F_N)$ by the subset constructions techniques then $L(D) = L(N)$.
- **Proof**: start by proving by induction on $|w|$ that $\hat{\delta}_D(\{q_0\},w) = \hat{\delta}_N(q_0,w)$
  - Both functions return sets of states, although one of them interpret them as simple/single states
  - **Basis step**: $|w|=0$, $w=\varepsilon$; by the basic rule of the definitions of $\hat{\delta}$ in both cases the result is $\{q_0\}$, $\hat{\delta}_D(\{q_0\}, \varepsilon) = \hat{\delta}_N(q_0, \varepsilon)=\{q_0\}$
  - **Induction step**: let $|w|=n+1$; break w up as $w=xa$
  - By the hypothesis: $\hat{\delta}_D(\{q_0\},x) = \hat{\delta}_N(q_0,x)$ and let both these sets of K states be $\{p_1, \ldots, p_k\}$

Theorem NFA – DFA (cont.)

- The inductive part of the $\hat{\delta}$ for NFAs says:

$$\hat{\delta}_N(q_0, w) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$$

- The subset construction defines:

$$\delta_D(\{p_1, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a)$$

- Using the equation above and the fact that: $\hat{\delta}_D(\{q_0\},x) = \{p_1, \ldots, p_k\}$

$$\hat{\delta}_D(\{q_0\}, w) = \delta_D(\hat{\delta}_D(\{q_0\}, x), a) = \delta_D(\{p_1, \ldots, p_k\}, a) = \bigcup_{i=1}^{k} \delta_N(p_i, a) = \hat{\delta}_N(q_0, w)$$

Theorem NFA – DFA (cont.)

- Then when we observe that D and N both accept w **iff** $\hat{\delta}_D(\{q_0\},w)$ or $\hat{\delta}_N(q_0,w)$, respectively, contains a state in F_N, we have a complete proof that L(D) = L(N).

Theorem of the DFA/NFA Language

- **Theorem**: the language L is accepted by a DFA, **iff** L is accepted by an NFA.
- **Proof**:
  - The **if** part is the subset construction and the previous theorem
  - The **only if** part is based on the recognition that a DFA can be thought as an NFA with only an option

Exercise 1

- Convert the following NFA to a DFA

	0	1
→p	{p,q}	{p}
q	{r}	{r}
r	{s}	∅
*s	{s}	{s}

Exercise 2

- Obtain an NFA, using as much as possible the non-determinism, to accept the language of the strings over the alphabet {0, …, 9} such that the last digit has appeared before.

Summary

- Non-Deterministic Automata (NFAs)
- Conversion of NFAs to DFAs
- Languages of DFAs and NFAs