Nondeterminism I

- Deterministic algorithm:
  Given current state and current input, next step is known

- Nondeterministic algorithm:
  Several choices are possible
  They will be respectively called
  DFA: deterministic finite automata
  and
  NFA: non-deterministic finite automata

- Fig 1.27
$\delta$ is not a function any more: $\delta(q_1, 1) = q_1$ or $q_2$

$\epsilon$ between $q_2$ and $q_3$: $q_2$ can move to $q_3$ without any input
How to run a string? We can separately consider different situations

It’s like we have a kind of parallel machines

ex: 010110

Fig 1.29
Nondeterminism IV

from $q_2$ via $\varepsilon$
After processing the string, if one path reaches an accept state, then the string is accepted.

Note that we handle the \( \epsilon \) edge immediately.

So each layer of the tree is the collection of states that can be reached up to the current input character.
Strings with 1 in 3rd position from the end
Strings 00100, 0100 are accepted, but 0010 is not
Fig 1.31

The only nondeterministic place is at $q_1$
Example 1.30 II

- At $q_1$ we nondeterministically guess if we are already at the third position from the end.
- Another difference is that at $q_4$, there are no out-links.
  This is crucial. It ensures that at $q_4$ we have 1 in the 3rd position from the end and can accept the string.
- Can we recognize this language by a DFA?
- An interesting issue is about the relationship between DFA and NFA.
- They are equivalent. We will formally explain this later.
Example 1.30 III

- For this example we can directly design a DFA for the language
- Fig 1.32
Example 1.30 IV

\[
\begin{align*}
q_{000} & \quad 0 \quad 0 \quad 0 \quad 0 \\
q_{100} & \quad 1 \quad 1 \quad 1 \\
q_{010} & \quad 0 \\
q_{110} & \quad 0 \\
q_{001} & \quad 1 \\
q_{101} & \quad 1 \\
q_{011} & \quad 1 \\
q_{111} & \quad 1
\end{align*}
\]
Example 1.30 V

- Idea of this diagram: using 8 states to record the past 3 digits so far
- Accept states: any $q_{1xx}$ can be an accept state
- The idea is simple. But why can we use 000 as the start state?
- Looks like we need other nodes:

  \[\_, \_, 0, \_, 1, \_, 01, \_, 10, \_, 00, \_, 11\]

- Then we see that the path is the same as if we start from 000
For example,

\[ ___ \rightarrow _0 \rightarrow _0^1 \rightarrow _0^1 \]
Consider a modification of the NFA in example 1.30

$q_2 \rightarrow q_3 : 0, 1 \Rightarrow 0, 1, \epsilon$

$q_3 \rightarrow q_4 : 0, 1 \Rightarrow 0, 1, \epsilon$
A modification of example 1.30 II

What is the language: at least one of the last three characters is 1

How about DFA for this language?
Except $q_{000}$, all others are in $F$