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
Nondeterminism II

\[ \delta \] 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? It can be run by split
A kind of parallel machines
ex: 010110
Fig 1.29
Some NFA examples I

- Example 1.30
  Strings with 1 in 3rd position from the end
  00100, 0100 are accepted, but 0010 is not

Fig 1.31

- DFA and NFA
Some NFA examples II

- They are equivalent. We will explain this later.

Fig 1.32. For this example in fact we are able to design a DFA for this language.
Some NFA examples III
Idea of this diagram: using 8 states to record the past 3 digits so far

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
Some NFA examples V

- For example,

  \[ \_\_0 \xrightarrow{0} \_0 \xrightarrow{1} \_10 \]

- A modification of the NFA

  \[
  q_2 \rightarrow q_3 : 0, 1 \Rightarrow 0, 1, \epsilon \\
  q_3 \rightarrow q_4 : 0, 1 \Rightarrow 0, 1, \epsilon
  \]
What is the language: one of the last three characters is 1

How about DFA for this language?
Except $q_{000}$, all others are in $F$
Consider the following figure
Example 1.33 II
Example 1.33 III

- For this language, $\Sigma = \{0\}$. This is called unary alphabets.
- What is the language?

$$\{0^k \mid k \text{ multiples of } 2 \text{ or } 3\}$$
Example 1.35 I

- Fig 1.36

- Accept
  - $\epsilon$, $a$, $baba$, $baa$ can be accepted
- But babba is rejected
Example 1.35 II

See the tree below

```
    q1    q3
   / \   /  \\
  b   ↓   ↓  ↓
   q2  q2  q3
```

- This example is later used to illustrate the procedure for converting NFA to DFA