Finite Automata

Alphabets and Languages

Def

$\Sigma^0 = \{\mathcal{E}\}$ contains only the empty word

Σ - alphabet (finite set)

$\Sigma^n = \{a_1 a_2 .. a_n \mid a_i \in \Sigma\}$ is the set of words of length n

$\Sigma^* = \{w \mid \exists n \in \mathbb{N}.\ \exists\ a_1, a_2, .., a_n \in \Sigma.\ w = a_1 a_2 .. a_n\}$ is the set of all words over Σ

A language L over Σ is a subset $L \subseteq \Sigma^*$

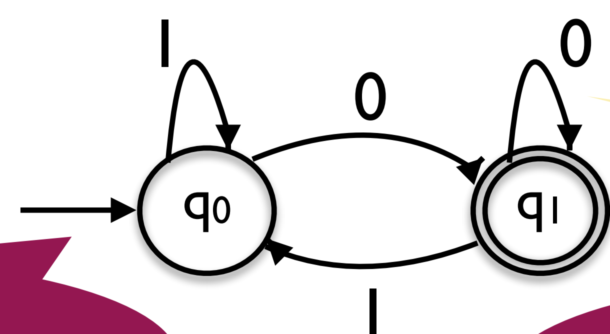
Deterministic Automata (DFA)

Informal example

alphabet

$\Sigma = \{0,1\}$

M_1:



q_0, q_1 are states

transitions, labelled by alphabet symbols

q_0 is initial

q_1 is final

Accepts the language $L(M_1) = \{w \in \Sigma^* \mid w \text{ ends with a } 0\} = \Sigma^*0$

regular language

regular expression

DFA

Definition

A deterministic automaton M is a tuple $M = (Q, \Sigma, \delta, q_0, F)$ where

Q is a finite set of states
Σ is a finite alphabet
$\delta: Q \times \Sigma \longrightarrow Q$ is the transition function
q_0 is the initial state, $q_0 \in Q$
F is a set of final states, $F \subseteq Q$

In the example M_1

$M_1 = (Q, \Sigma, \delta, q_0, F)$ for

$Q = \{q_0, q_1\}$ $F = \{q_1\}$

$\Sigma = \{0, 1\}$

$\delta(q_0, 0) = q_1$, $\delta(q_0, 1) = q_0$
$\delta(q_1, 0) = q_1$, $\delta(q_1, 1) = q_0$

DFA

The extended transition function

Given $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma \longrightarrow Q$ to

$\delta^*: Q \times \Sigma^* \longrightarrow Q$

In M_1, $\delta^*(q_0, 110010) = q_1$

inductively, by:

$\delta^*(q, \varepsilon) = q$ and $\delta^*(q, wa) = \delta(\delta^*(q, w), a)$

Definition

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$

The language recognised / accepted by a deterministic finite automaton $M = (Q, \Sigma, \delta, q_0, F)$ is

$$L(M) = \{w \in \Sigma^* | \ \delta^*(q_0, w) \in F\}$$

Regular languages and operations

$L(M_1) = \{w0 | w \in \{0,1\}^*\}$
is regular

Definition

Let Σ be an alphabet. A language L over Σ ($L \subseteq \Sigma^*$) is regular iff it is recognised by a DFA.

Regular operations

Let L, L_1, L_2 be languages over Σ. Then $L_1 \cup L_2$, $L_1 \cdot L_2$, and L^* are languages, where

$$L_1 \cdot L_2 = \{w_1 \cdot w_2 \mid w_1 \in L_1, w_2 \in L_2\}$$

$$L^* = \{w \mid \exists n \in \mathbb{N}.\ \exists\, w_1, w_2, .., w_n \in L.\ w = w_1 w_2 .. w_n\}$$

$\varepsilon \in L^*$ always

Closure under regular operations

also under intersection

Theorem C1

The class of regular languages is closed under union

We can already prove these!

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

But not yet these two…

Theorem C4

The class of regular languages is closed under Kleene star finite representation of infinite languages

Regular expressions

inductive

example: $(ab \cup a)^*$

Definition

Let Σ be an alphabet. The following are regular expressions

1. a for $a \in \Sigma$
2. ε
3. $\emptyset$
4. $(R_1 \cup R_2)$ for R_1, R_2 regular expressions
5. $(R_1 \cdot R_2)$ for R_1, R_2 regular expressions
6. $(R_1)^*$ for R_1 regular expression

corresponding languages

$L(a) = \{a\}$

$L(\varepsilon) = \{\varepsilon\}$

$L(\emptyset) = \emptyset$

$L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$

$L(R_1 \cdot R_2) = L(R_1) \cdot L(R_2)$

$L(R_1^*) = L(R_1)^*$

Equivalence of regular expressions and regular languages

Theorem (Kleene)

A language is regular (i.e., recognised by a finite automaton) iff it is the language of a regular expression.

needs nondeterminism

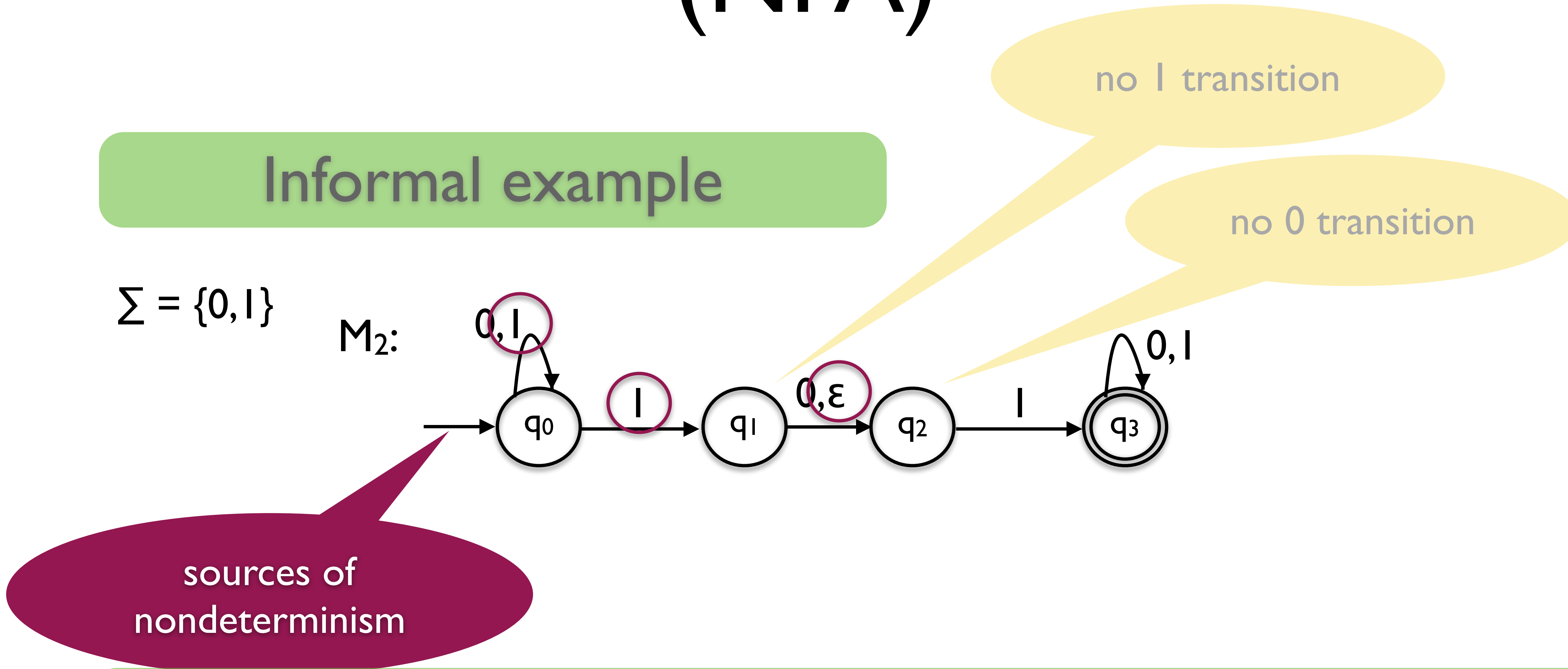
Proof $\Leftarrow$ easy, as the constructions for the closure properties,

$\Rightarrow$ not so easy, we'll skip it for now…

Nondeterministic Automata (NFA)

Informal example

$\Sigma = \{0,1\}$



Accepts a word iff there exists an accepting run