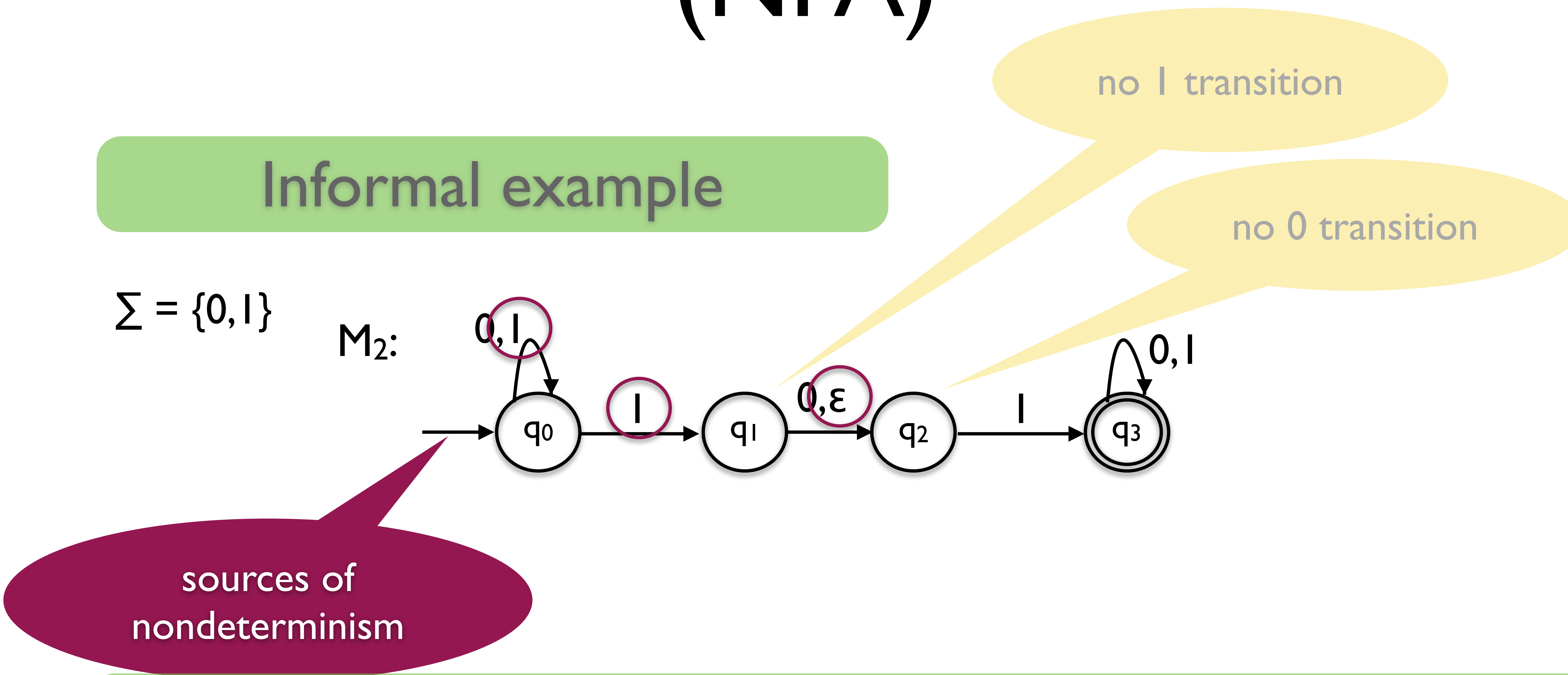
Nondeterministic Automata (NFA)

Informal example

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



Accepts a word iff there exists an accepting run

NFA

Definition

A nondeterministic 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
- $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ is the transition function
- q_0 is the initial state, $q_0 \in Q$
- F is a set of final states, $F \subseteq Q$

$\Sigma_\varepsilon = \Sigma \cup \{\varepsilon\}$

In the example M

$M_2 = (Q, \Sigma, \delta, q_0, F)$ for

$Q = \{q_0, q_1, q_2, q_3\}$

$\Sigma = \{0, 1\}$ $F = \{q_3\}$

$\delta(q_0, 0) = \{q_0\}$

$\delta(q_0, 1) = \{q_0, q_1\}$

$\delta(q_0, \varepsilon) = \varnothing$

.....

NFA

ε-closure of q, all states reachable by ε-transitions from q

$E(q) = \{q' \mid q' = q \vee \exists n \in \mathbb{N}^+ . \exists q_0, .., q_n \in Q . q_0 = q, q_n = q', q_{i+1} \in \delta(q_i, \varepsilon), \text{ for } i = 0, .., n-1\}$

The extended transition function

Given an NFA $M = (Q, \Sigma, \delta, q_0, F)$ we can extend $\delta: Q \times \Sigma_\varepsilon \longrightarrow \mathcal{P}(Q)$ to

$\delta^*: Q \times \Sigma^* \longrightarrow \mathcal{P}(Q)$

$E(X) = \bigcup_{x \in X} E(x)$

In M_2, $\delta^*(q_0, 0110) = \{q_0, q_2, q_3\}$

inductively, by:

$\delta^*(q, \varepsilon) = E(q)$ and $\delta^*(q, wa) = E(\bigcup_{q' \in \delta^*(q,w)} \delta(q', a))$

Definition

$L(M_2) = \{u101w \mid u,w \in \{0,1\}^*\}$ $\cup$ $\{u11w \mid u,w \in \{0,1\}^*\}$

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

$$L(M) = \{w \in \Sigma^* \mid \delta^*(q_0, w) \cap F \neq \varnothing\}$$

Equivalence of automata

Definition

Two automata M_1 and M_2 are equivalent if $L(M_1) = L(M_2)$

Theorem NFA ~ DFA

Every NFA has an equivalent DFA

Proof via the "powerset construction" / determinization

Corollary

A language is regular iff it is recognised by a NFA

Closure under regular operations

Theorem C1

The class of regular languages is closed under union

Theorem C2

The class of regular languages is closed under complement

Theorem C3

The class of regular languages is closed under concatenation

Theorem C4

The class of regular languages is closed under Kleene star

Now we can prove these too

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. $\varnothing$
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

$L(a) = \{a\}$
$L(\varepsilon) = \{\varepsilon\}$
$L(\varnothing) = \varnothing$
$L(R_1 \cup R_2) = L(R_1) \cup L(R_2)$
$L(R_1 \cdot R_2) = L(R_1) \cdot L(R_2)$
$L(R_1^*) = L(R_1)^*$

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.

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

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