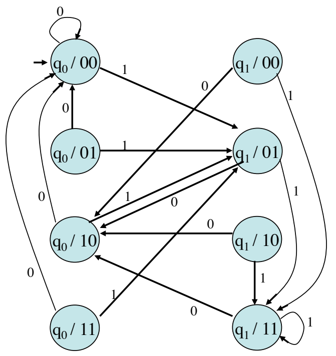
SAPIENZA
Università di Roma
Dipartimento di Informatica

Automata Minimization

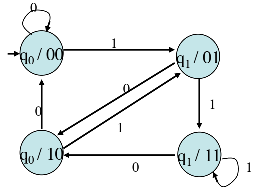
Prof. Daniele Gorla

Two Moore automata for the reminders MOD 4

SAPIENZA
Università di Roma
Dipartimento di Informatica



≠



????

Minimal Automaton

SAPIENZA
Università di Roma
Dipartimento di Informatica

Def.: two automata (of the same kind) are *equivalent* if, for every possible input sequence, they produce the same output sequence.

REMARK: two two previous Moore automata are equivalent (with the same input they give the same output), but the first one is more complex!

The minimal automaton equivalent to *M* is the automaton equivalent to *M* but with the minimum number of states (and transitions)

REMARK: it can be proved that such an automaton is *unique*, up to states renaming.

As we intuitively saw, an automaton is the abstract model of a sequential net. Hence, minimizing an automaton implies a reduction on the number of memory components needed in the corresponding circuit.

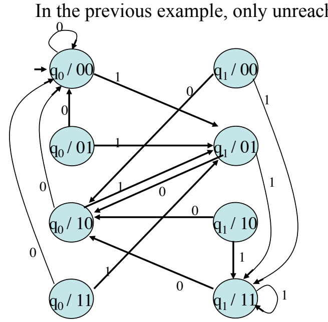
Unreachable States

SAPIENZA
Università di Roma
Dipartimento di Informatica

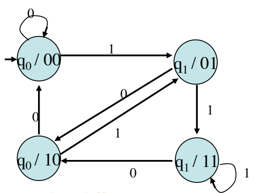
The first step in the minimization is canceling *unreachable states*, i.e. those states q such that, for every input sequence $a_1 \ldots a_n$, we have that

$$q_0 \xrightarrow{a_1} q_1 \xrightarrow{a_2} q_2 \ldots \xrightarrow{a_n} q_n \neq q$$

In the previous example, only unreachable states come into play:



=

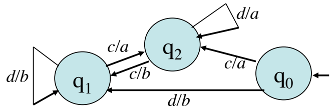


REMARK: with a different choice of the initial state, also the minimal automaton would have been different!

Indistinguishable States (1)

…but canceling unreachable states is usually not enough:

Ex.:

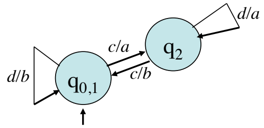


States q_0 and q_1 behave the same: with the same input, they produce the same output and reach the same state.

→ it is useless having both q_0 and q_1 → WE MERGE THEM

By contrast, q_2 behaves differently and so it must remain distinguished

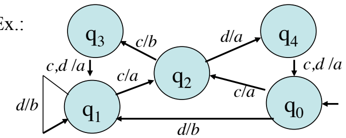
Minimal automaton:



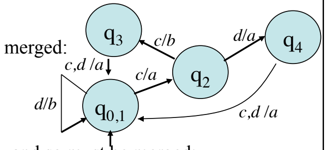
Indistinguishable States (2)

…indistinguishability is an evolution of this notion:

Ex.:



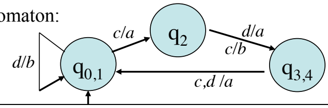
q_0 and q_1 are indistinguishable and must be merged:



This makes also q_3 and q_4 indistinguishable, and so must be merged;

By contrast, q_2 behaves differently and remains different.

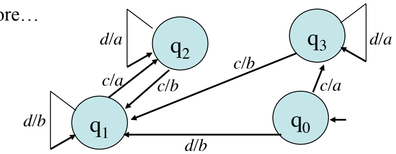
Minimal automaton:



Indistinguishable states (3)

…but there is more…

Ex.:



q_0 and q_1 always return the same output, but with input *c* don't reach the same state.

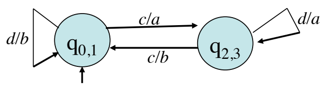
→ if q_2 and q_3 are indistinguishable, also q_0 and q_1 are (see previous slide)

q_2 and q_3 always return the same output, but with input *d* don't reach the same state.

→ indistinguishability of q_2 and q_3 depends from indistinguishability of themselves

→ if indistinguishable means "not distinguishable" (see later), we can merge!

Minimal automaton:

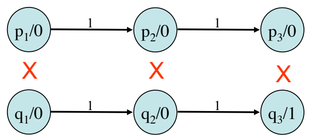


Distinguishability (Moore)

In a Moore automaton, two states are *distinguishable* if

1. The associated outputs are different, or
2. There is an outgoing stransition labeled with the same input that leads to distinguishable states.

Ex.:



In: ε, Out_{p3}: 0, Out_{q3}: 1

In: 1, Out_{p2}: 00, Out_{q2}: 01

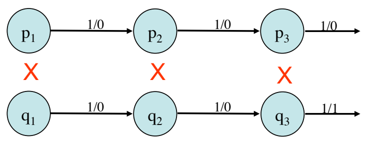
In: 11, Out_{p1}: 000, Out_{q1}: 001

Distinguishability (Mealy)

In a Mealy automaton, two states are *distinguishable* if they have an outgoing transition labeled with the same input but

1. The two transitions return different outputs, or
2. Lead to distinguishable states.

Ex.:



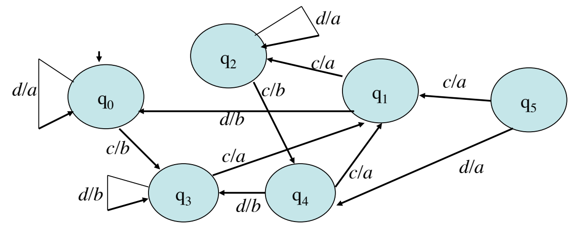
In: 1, Out_{p3}: 0, Out_{q3}: 1

In: 11, Out_{p2}: 00, Out_{q2}: 01

In: 111, Out_{p1}: 000, Out_{q1}: 001

Algorithm for the minimal automaton (step 0: unreachable states)

First cancel all unreachable states:



In this example, since the starting state is q0, state q5 is unreachable

REMARK: unreachability strongly depends on the starting state!

→ without knowing which is the starting state, we cannot deem any state unreachable!!

Algorithm for the minimal automaton (step 1: triangular table)

We have to consider every pair of states, to check whether they are distinguishable or not.

q0					
q1					
q2					
q3					
q4					
	q0	q1	q2	q3	q4

Useless because we already consider the symmetric pair in the lower part of the table

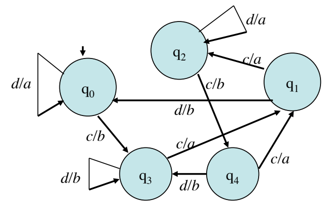
Useless because surely not distinguishable

What remains is an lower triangular table where we can record the comparison between every pair of states (that can be distinguishable).

Algorithm for the minimal automaton (step 2: immediate distinguishability)

- Check all cells one after the other
- In a cell put an X if the states are distinguishable
  - Moore: states with different output;
  - Mealy: states with an outgoing transition with same input but different output

q1	X			
q2		X		
q3	X		X	
q4	X		X	
	q0	q1	q2	q3

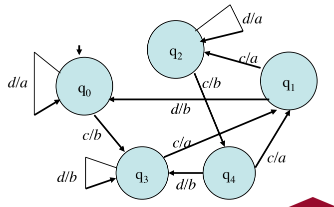


Algorithm for the minimal automaton (step 3: propagated distinguishability)

Check one after the other all non-marked cells; in a cell put

- X, if with the same input the two states reach a pair already marked with X;
- O, if, for every possible input, you reach either the same state or a pair of states already marked with O or in the pair associated to the cell itself;
- If none of the previous two conditions hold, mark the cell with all the pairs not yet marked and different from the pair associated that you can reach with the same input.

q1	X			
q2	(3,4)	X		
q3	X	X	X	
q4	X	X	X	O
	q0	q1	q2	q3



Algorithm for the minimal automaton (step 4: final marking of the table)

SAPIENZA
Università di Roma
Dipartimento di Informatica

For all cells that point to at least another pair of states, examine all the pointed pairs

- If they are marked with X, mark with X also the original cell;
- If they are marked with O, cancel the link to such pairs;
- Otherwise, leave the link into the original cell.

At the end, every cell will either be marked with X or contain mo more pointer to any other pair; in this case it will be marked with O.

q1	X			
q2	O (3,4)	X		
q3	X	X	X	
q4	X	X	X	O
	q0	q1	q2	q3

Algorithm for the minimal automaton (step 5: minimal automaton)

SAPIENZA
Università di Roma
Dipartimento di Informatica

At the end, ALL equivalent pairs will be marked with O

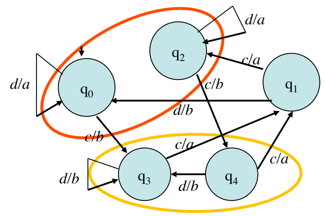
So, from the triangular table we can derive the *indistinguishability classes*: two states are in the same class (and so are equivalent) if and only if the pair associated to them is marked with O in the table.

The minimal automaton has

- States formed by the indistinguishability classes;
- The initial state is the class that contains the old initial state;
- The transition function is obtained by putting
  (class(q) , *a*) → class(q') whenever (q,*a*) → q' ∈ δ
- The output function is obtained by giving
  - The output of q to the class of q (Moore)
  - The output of (q,*a*) → q' to the transition (class(q),*a*) → class(q') (Mealy)

Example

SAPIENZA
Università di Roma
Dipartimento di Informatica



q1	X			
q2	O	X		
q3	X	X	X	
q4	X	X	X	O
	Q0	q1	q2	q3