Introduction to Computer Science

Géza Horváth

Fourth Lecture

Finite automata – minimization

Let $A = (Q, T, q_0, \delta, F)$ be a deterministic finite automaton such that each of its states is reachable from its initial state (there are no useless states). Then we can construct the minimal deterministic finite automaton that is equivalent to A in the following way:

Let us divide the set of states into two groups obtaining the classification $C_1 = \{F, Q \setminus F\}$. (We denote the class where state q is by $C_1[q]$.)

Then, for $i > 1$ the classification C_i is obtained from C_{i-1}: the states p and q are in the same class by C_i if and only if they are in the same class by C_{i-1} and for every $a \in T$ they behave similarly: $\delta(p, a)$ and $\delta(q, a)$ are in the same class by C_i.

Set Q is finite and, therefore, there is a classification C_m such that it is the same as C_{m+1}.

Finite automata – minimization

Then, we can define the minimal completely defined deterministic automaton that is equivalent to A: its states are the groups of the classification C_m, the initial state is the group containing the initial state of the original automaton, the final states are those groups that are formed from final states of the original automaton, formally:

$$(C_m, T, C_m[q_0], \delta_{C_m}, F_{C_m}),$$

where $\delta_{C_m}(C_m[q], a) = C_m[\delta(q, a)]$ for every $C_m[q] \in C_m,\ a \in T$ and $F_{C_m} = \{C_m[q] \mid q \in F\}$.

Finite automata – minimization example

Example

Let the deterministic finite automaton A be given as follows:

$T \quad Q$	$\rightarrow q_0$	q_1	$\overline{q_2}$	q_3	$\overline{q_4}$	$\overline{q_5}$	$\overline{q_6}$
a	q_2	q_5	q_1	q_1	q_2	q_1	q_0
b	q_1	q_0	q_3	q_4	q_5	q_3	q_2

Give a minimal deterministic finite automaton that is equivalent to A.

Finite automata – minimization example

Example

Solution:

Before applying the algorithm we must check which states can be reached from the initial state: from q_0 one can reach the states $q_0, q_2, q_1, q_3, q_5, q_4$. Observe that the automaton cannot enter state q_6, therefore, this state (column) is deleted. The task is to minimize the following automaton by the algorithm.

$T \quad Q$	$\rightarrow q_0$	q_1	$\circledcirc q_2$	q_3	$\circledcirc q_4$	$\circledcirc q_5$
a	q_2	q_5	q_1	q_1	q_2	q_1
b	q_1	q_0	q_3	q_4	q_5	q_3

Finite automata – minimization example

Example

We perform the first classification of the states $C_1 = \{Q_1, Q_2\}$ *by separating the accepting and non-accepting states.*
$Q_1 = \{q_2, q_4, q_5\}, Q_2 = \{q_0, q_1, q_3\}$ *then we have:*

T \ Q	Q_1			Q_2		
	$(\overline{q_2})$	$(\overline{q_4})$	$(\overline{q_5})$	$\rightarrow q_0$	q_1	q_3
a	Q_2	Q_1	Q_2	Q_1	Q_1	Q_2
b	Q_2	Q_1	Q_2	Q_2	Q_2	Q_1

Finite automata – minimization example

Example

Then $C_2 = \{Q_{11}, Q_{12}, Q_{21}, Q_{22}\}$ *with* $Q_{11} = \{q_2, q_5\}$, $Q_{12} = \{q_4\}$, $Q_{21} = \{q_0, q_1\}$, $Q_{22} = \{q_3\}$. *Then according to this classification we have:*

Q / T	Q_{11}		Q_{12}	Q_{21}		Q_{22}
	$\overline{q_2}$	$\overline{q_5}$	$\overline{q_4}$	$\rightarrow q_0$	q_1	q_3
a	Q_{21}	Q_{21}	Q_{11}	Q_{11}	Q_{11}	Q_{21}
b	Q_{22}	Q_{22}	Q_{11}	Q_{21}	Q_{21}	Q_{12}

Finite automata – minimization example

Example

Since $C_3 = C_2$ we have the solution, the minimal deterministic finite automaton equivalent to A:

T Q	$\subset Q_{11} \supset$	$\subset Q_{12} \supset$	$\rightarrow Q_{21}$	Q_{22}
a	Q_{21}	Q_{11}	Q_{11}	Q_{21}
b	Q_{22}	Q_{11}	Q_{21}	Q_{12}

Mealy automata – definition

We start by giving the formal definition of Mealy automata.

Definition

A Mealy automaton is an ordered sextuple $A = (Q, T, V, q_0, \delta, \mu)$*, where* Q, T, q_0, δ *are the same as at the completely defined deterministic finite automata, i.e.,* Q *is the finite set of states,* T *is the input alphabet,* $q_0 \in Q$ *is the initial state,* $\delta : Q \times T \to Q$ *is the transition function; and* V *is the output alphabet and* $\mu : Q \times T \to V$ *is the output function.*

Notice that there are no final (or accepting) states.
These automata are not used to accept languages.

Mealy automata – representation

A Mealy automaton can be defined by a Cayley table or by a graph. When a Cayley table is used to describe a Mealy automaton, then both the values $\delta(q, a)$ and $\mu(q, a)$, as pairs are written to the cell identified by the state q and by the letter a. When a graph is used to describe a Mealy automaton, then we can put a/x to an arrow meaning that the transition represented by the arrow is performed by reading an input letter a, while an output letter $x \in V$ is written to the output tape.

Mealy automata representation – Cayley table

Example (Mealy automata – Cayley table)

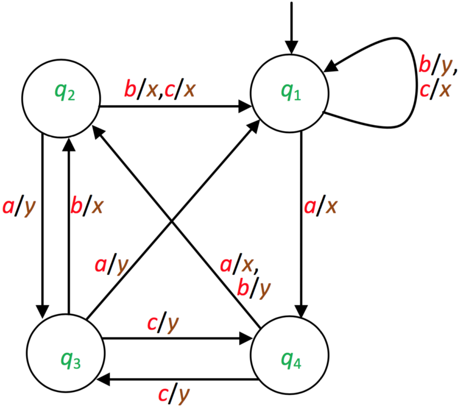
Let the Mealy automaton A be given by the following Cayley table

$T \quad Q$	$\to q_1$	q_2	q_3	q_4
a	(q_4, x)	(q_3, y)	(q_1, y)	(q_2, x)
b	(q_1, y)	(q_1, x)	(q_2, x)	(q_2, y)
c	(q_1, x)	(q_1, x)	(q_4, y)	(q_3, y)

Let us make some sample runs of the automaton:

- Let the input be *abcabc*, then the output is *xyxxyx*.
- Let the input be *aaaabbbb*, then the output is *xxyyyyyy*.
- Let the input be *ccabcabc*, then the output is *xxxyxxyx*.
- Let the input be *abcbbccabac*, then the output is *xyxyyxxxyyy*.

Mealy automata representation – graph



Let us make some sample runs of the automaton:

- Let the input be *abcabc*, then the output is *xyxxyx*.
- Let the input be *aaaabbbb*, then the output is *xxyyyyyy*.

Mealy automata – minimization

Definition

We say that two Mealy automata are equivalent if they assign the same output word for every input word $u \in T^*$.

The number of states of equivalent automata can be various. However, there is particular Mealy automaton for each equivalent class that has a minimal number of states. This automaton can be obtained from any automaton of the class by the minimization algorithm.

Now we present the minimization algorithm for Mealy automata. First, as with the finite state recognizers, we should check which states can be reached from the initial state. The states that cannot be reached can simply be erased from the automaton (table or graph) together with the transitions from them.

Mealy automata – minimization

When we have a Mealy automaton, such that each of its states is reachable from the initial state with some input words (i.e., reading the given input word the automaton arrives to this particular state), then we can start an analogous algorithm that was used for minimizing finite state recognizers.

Only the initial step of the algorithm differs from the one shown previously: since we have no accepting states, the initial classification is done in another way. Let two states p and q be in the same class, i.e., $C_1[p] = C_1[q]$ if and only if for every input letter $a \in T$ the equality $\mu(p, a) = \mu(q, a)$ is fulfilled.

The other steps of the algorithm are similar to the previously described algorithm: based on the previous classification the next classification is obtained by separating those states for which there is an input letter such that the transition function with this letter brings them to different classes.

Mealy automata – minimization example

Example

Let the Mealy automaton A be given as follows:

T Q	$\rightarrow q_1$	q_2	q_3	q_4	q_5	q_6
a	(q_2, x)	(q_1, z)	(q_6, x)	(q_5, z)	(q_4, x)	(q_5, x)
b	(q_3, y)	(q_2, y)	(q_4, y)	(q_4, y)	(q_3, y)	(q_2, y)

Give the minimal Mealy automaton that is equivalent to A.

Mealy automata – minimization example

Example

Solution:

One can easily check that every state can be reached from the initial state. Then classification $C_1 = \{Q_1, Q_2\}$, *where* $Q_1 = \{q_1, q_3, q_5, q_6\}$ *(having output* x *for input* a *and output* y *for input* b*) and* $Q_2 = \{q_2, q_4\}$ *(having output* z *for input* a *and output* y *for input* b*). Then, the transition function reflecting this classification is as follows:*

Q / T	Q_1				Q_2	
	$\to q_1$	q_3	q_5	q_6	q_2	q_4
a	Q_2	Q_1	Q_2	Q_1	Q_1	Q_1
b	Q_1	Q_2	Q_1	Q_2	Q_2	Q_2

Mealy automata – minimization example

Example

Then, Q_1 *is divided into two subgroups in the classification* $C_2 = \{Q_{11}, Q_{12}, Q_2\}$ *with* $Q_{11} = \{q_1, q_5\}$ *and* $Q_{12} = \{q_3, q_6\}$. *(*$Q_2 = \{q_2, q_4\}$ *remains the same group.) The transition function reflecting these groups is as follows:*

T \ Q	Q_{11}		Q_{12}		Q_2	
	$\rightarrow q_1$	q_5	q_3	q_6	q_2	q_4
a	Q_2	Q_2	Q_{11}	Q_{12}	Q_{11}	Q_{11}
b	Q_{12}	Q_{12}	Q_2	Q_2	Q_2	Q_2

Mealy automata – minimization example

Example

Only Q_{12} *is divided (to its elements) and thus*
$C_3 = \{Q_{11}, Q_{121}, Q_{122}, Q_2\}$ *with* $Q_{121} = \{q_3\}$ *and* $Q_{122} = \{q_5\}$.
Then the transitions become:

$T \backslash Q$	Q_{11}: $\to q_1$	q_5	Q_{121}: q_3	Q_{122}: q_6	Q_2: q_2	q_4
a	Q_2	Q_2	Q_{11}	Q_{122}	Q_{11}	Q_{11}
b	Q_{121}	Q_{121}	Q_2	Q_2	Q_2	Q_2

Mealy automata – minimization example

Example

Since $C_4 = C_3$, we can give the minimal Mealy automaton (writing also the values of the output function into the table):

T $\quad Q$	$\rightarrow Q_{11}$	Q_{121}	Q_{122}	Q_2
a	(Q_2, x)	(Q_{11}, x)	(Q_{122}, x)	(Q_{11}, z)
b	(Q_{121}, y)	(Q_2, x)	(Q_2, y)	(Q_2, y)

Moore automata – definition

Definition

A Moore automaton is an ordered sextuple $A = (Q, T, V, q_0, \delta, \eta)$, where Q, T, V, q_0, δ are the same as at the Mealy automata, and $\eta : Q \to V$ is the output function.

Notice that the difference between the Mealy and the Moore automata is due to their output function. While with the Mealy automata the output is produced during the transition (depending on both the state that the automaton was in and on the read input letter), at the Moore automata the output letter is produced after the transition is finished and the output letter depends only on the state the automaton reached by the transition.

Moore automata – representation

The Moore automata can also be defined by Cayley table and by graph. Since with the Moore automata the output depends only on the state the automaton has reached, the output letters are written to the states (above the states, when the states are in the 0th row of table) and inside the circles of the states as pairs containing the state and the output assigned to the state on the graphs.

Moore automata representation – Cayley table

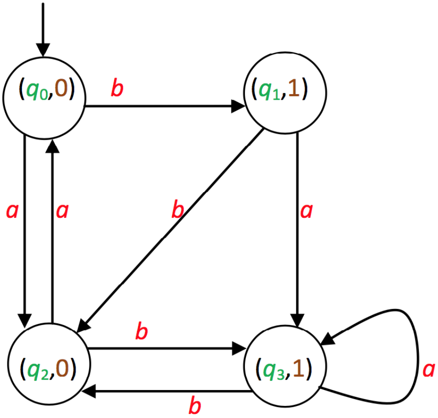
Example

T \ V / Q	0 / $\to q_0$	1 / q_1	0 / q_2	1 / q_3
a	q_2	q_3	q_0	q_3
b	q_1	q_2	q_3	q_2

The example runs give the outputs as follows:

- For input *aabb* the output is 0010.
- For input *baabaa* the output is 111000.
- For input *abaababb* the output is 01110010.

Moore automata representation – graph



Let us make some sample runs of the automaton:

- For input *aabb* the output is 0010.
- For input *baabaa* the output is 111000.

Moore automata – minimization

Definition

Now we can generalize the equivalence relation between finite transducers: a Mealy/Moore automaton A is equivalent to a Mealy/Moore automaton A' if and only if for every input string $u \in T^*$ they produce the same output string.

The Moore automata can also be minimized and its algorithm is very similar to the previously described minimization algorithms. The only difference is that in this case the first classification is done based on the output letters assigned to the states, i.e., the states p and q are in the same class by classification C_1 if and only if $\eta(p) = \eta(q)$.

Moore automata – minimization example

Example

The Moore automata A is defined by its Cayley table as follows:

V	x	y	y	y	x	x	x
T Q	$\rightarrow q_1$	q_2	q_3	q_4	q_5	q_6	q_7
a	q_2	q_5	q_1	q_5	q_2	q_3	q_4
b	q_7	q_4	q_6	q_2	q_4	q_6	q_1

Find a minimal Moore automaton that is equivalent to A.

Moore automata – minimization example

Example

Solution:

First we check if every state is reachable from the initial state. It can be seen that states q_3 and q_6 are not reachable, therefore we erase them. We need to minimize the automaton

T \ V / Q	x / $\rightarrow q_1$	y / q_2	y / q_4	x / q_5	x / q_7
a	q_2	q_5	q_5	q_2	q_4
b	q_7	q_4	q_2	q_4	q_1

Moore automata – minimization example

Example

Classification $C_1 = \{Q_1, Q_2\}$ *is based on the output function:* $Q_1 = \{q_1, q_5, q_7\}$ *(having output* x*) and* $Q_2 = \{q_2, q_4\}$ *(having output* y*). Then, the transitions using the classification become:*

$T \backslash Q$	Q_1			Q_2	
	$\to q_1$	q_5	q_7	q_2	q_4
a	Q_2	Q_2	Q_2	Q_1	Q_1
b	Q_1	Q_2	Q_1	Q_2	Q_2

Moore automata – minimization example

Example

Then, Q_1 *is divided into two subclasses, therefore* $C_2 = \{Q_{11}, Q_{12}, Q_2\}$ *where* $Q_{11} = \{q_1, q_7\}$ *and* $Q_{12} = \{q_5\}$. *Then, we have:*

Q / T	Q_{11}		Q_{12}	Q_2	
	$\rightarrow q_1$	q_7	q_5	q_2	q_4
a	Q_2	Q_2	Q_2	Q_{12}	Q_{12}
b	Q_{11}	Q_{11}	Q_2	Q_2	Q_2

Moore automata – minimization example

Example

Thus $C_3 = C_2$, and we can describe the minimal Moore automaton as follows:

T \ V, Q	x, $\rightarrow Q_{11}$	x, Q_{12}	y, Q_2
a	Q_2	Q_2	Q_{12}
b	Q_{11}	Q_2	Q_2

References

This work was supported by the construction EFOP-3.4.3-16-2016-00021. The project was supported by the European Union, co-financed by the European Social Fund.