

Inhaltsverzeichnis

1	Kapitelüberschrift	2
1.1	Mealy machines and apartness:.....	2
1.2	The learning Algorythm $L_{\#}$	4
1.2.1	The Teacher	4
1.2.2	the main Loop.....	4
	Tabellenvarianten.....	9
	Tabellenvarianten 2	10

1. Introduction

2. L#

2.1. Mealy machines and apartness:

a mealy machine can be understood as a finite state automaton with inputs and outputs.

Definition 2.1. a mealy machine M is a 6-tuple $M = (Q, q_0, I, O, \delta, \lambda)$ where

- Q is the finite set of states
- q_0 is the initial state
- I/O are the finite sets of input/output symbols
- $\delta : Q \times I \rightarrow Q$ the state transition function
- $\lambda : Q \times I \rightarrow O$ the output function

it is important to note that both functions need to be defined on the same inputs $\delta(x) \downarrow \iff \lambda(x) \downarrow$. We define $\langle \lambda, \delta \rangle : Q \times I \rightarrow O \times Q$ as the output transition function. Throughout this explanation we fix I and O . to differentiate between mealy machines we use superscript M e.g. $M = (Q^M, q_0^M, I, O, \delta^M, \lambda^M)$. to denote that $\langle \lambda, \delta \rangle(q, i) = (q', o)$ we write $q \xrightarrow{i/o} q'$. to use the output transition function with words of length $n \in \mathbb{N}$ we compose $\langle \lambda, \delta \rangle$ n times with itself: we define $\langle \lambda_n, \delta_n \rangle : Q \times I^n \rightarrow O^n \times Q$ by defining $\langle \lambda_0, \delta_0 \rangle = id_Q$ and $\langle \lambda_{n+1}, \delta_{n+1} \rangle = Q \times I^{n+1} \xrightarrow{\langle \lambda_n, \delta_n \rangle \times id_I} O^n \times Q \times I \xrightarrow{id_{O^n} \times \langle \lambda, \delta \rangle} O^{n+1} \times Q$. We call a Mealy machine Complete if δ (and therefore λ) is Total.

intuitively we can see each state q of a mealy machine as a Map from input words to output words.

Definition 2.2. we write $\llbracket q \rrbracket : I^* \times O^*$ where $\llbracket q \rrbracket(\sigma) = \lambda(q, \sigma)$. two states q and q' are equivalent $q \approx q'$ iff $\llbracket q \rrbracket = \llbracket q' \rrbracket$, both states can be from different mealy machines. two Mealy machines are equivalent if their initial states are equivalent $M \approx N \iff q_0^M \approx q_0^N$.

in the learning context we are in we see undefined values in partial functions as lack of knowledge, to show that two mealy machines act the same on the input defined by both machines we define a functional simulation. this helps with expansion of a partial mealy machine as we can expand the functional simulation. formally:

Definition 2.3. for two mealy machines M and N we call $f : M \rightarrow N$ if f is a map $f : Q^M \rightarrow Q^N$ and

$$f(q_0^M) = q_0^N \text{ and } q \xrightarrow{i/o} q' \implies f(q) \xrightarrow{i/o} f(q')$$

as our primary data structure to store learned information we use a Mealy machine that is

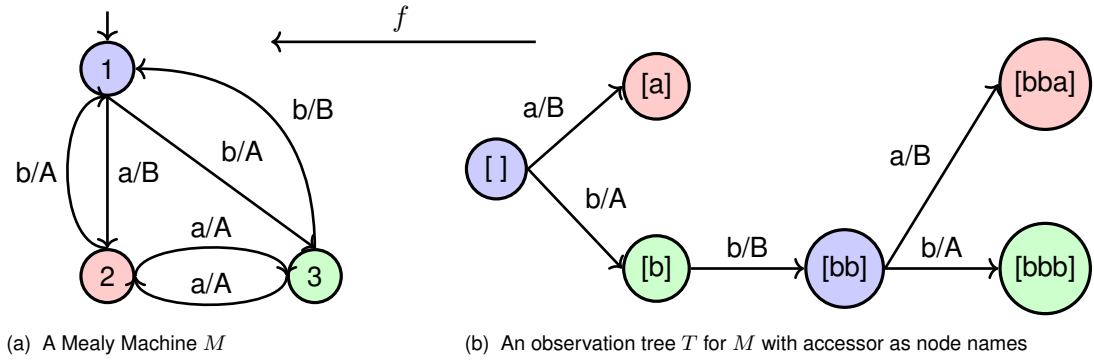


Abbildung 1 an observation tree simulates a Mealy machine

also a tree. this tree gets expanded with new information upon finding out. we call the tree an observation tree. formally we define:

Definition 2.4. A mealy machine T is a Tree if for each $q \in Q^T$ there is a unique accessor $\delta \in I^*$ so that $\sigma^T(q_0^T, \sigma) = q$. we call a Tree T an Observation tree for a mealy machine M if there is a functional simulation $f : T \rightarrow M$.

Figure 1 shows a complete mealy machine for $I^M = a, b$ as well as an observation tree for M . the nodes of the observation tree are labeled with the accessor of the actual node, e.g. $\sigma^T(q_0^T, bb) = [bb]$. f is a functional simulation from T to M $f : T \rightarrow M$ and each node in the observation tree has the color of the node that it corresponds with in M e.g. $f([a]) = 2$. throughout the learning algorithm the observation tree gets expanded, but as the learner only learns via queries and has no further information about the underlying Mealy machine the functional simulation is unknown. to be able to produce a Mealy machine that could be matching the hidden Mealy machine the algorithm only infers that two states can not be the same state of the hidden machine, e.g. even if we dont know what state $[b]$ represents, we can infer that $f([b]) \neq f([])$ because $\lambda^T([b], b) = B$ and $\lambda^T([], b) = A$. we call this form of inequality apartness. formally:

Definition 2.5. For a Mealy machine M , we say that the two states $p, q \in Q^M$ are apart(written $p \# q$) if there is some $\sigma \in I^*$ such that $\llbracket p \rrbracket(\sigma) \downarrow \wedge \llbracket q \rrbracket(\sigma) \downarrow$ and $\llbracket p \rrbracket(\sigma) \neq \llbracket q \rrbracket(\sigma)$. we call σ a witness for of $p \# q$ and write $\sigma \vdash p \# q$.

as the apartness relation notes a difference in the semantics of two states, two apart states can not be mapped to the same state in a functional simulation $f : T \rightarrow M$

$$p \# q, p, q \in Q^T \implies f(q) \not\approx f(p)$$

thus whenever the learner knows that two states are apart in the observation tree, it knows that these are corresponding to different states in the hidden Mealy machine.

2.2. The learning Algorithm $L_{\#}$

2.2.1. The Teacher

with the invention of automata Learning Dana Angluin also proposed the minimal adequate teacher Framework. this framework allows the teacher to answer to types of queries: a membership query, that allows to check if a string is in a hidden set. the second query allows the learner to ask if their description of a set produces the same set as the hidden set, if the answer is false the teacher also responds with a counterexample why the sets are not the same.[?,]

for our purposes of learning a hidden mealy machine we formulate the two queries a bit different.

Definition 2.6. in the learning game between a teacher and a learner where the teacher has knowledge of a hidden mealy machine M we expect the teacher to answer the following queries correctly:

OutputQuery(σ) : for $\sigma \in I^*$ the teacher replies with the output sequence $\lambda^M(q_0^M, \sigma) \in O^*$

EquivQuery(H) : for a Complete Mealy Machine H the teacher replies with yes if $H \approx M$ or no and a counterexample $\sigma \in I^*$ with $\lambda^M(q_0^M, \sigma) \neq \lambda^H(q_0^H, \sigma)$

our output query lets us expand our observation tree along a chosen route σ , and if we build a hypothesis as to how the mealy machine looks we can verify it with the equivalence query or obtain a counterexample that eliminates the hypothesis we build from the ones we could create.

2.2.2. the main Loop

Passen Sie gegebenenfalls die Ränder an die Vorgaben Ihres Prüfers an und beachten Sie dabei, dass das Logo der TUM sich oben rechts innerhalb der Ränder, auf der Titelseite befindet. Für die Titelseiten stehen separate Vorlagen zur Verfügung.

Zur Definition von Abk. (Abkürzungen) erstellen Sie für die gewünschte Abkürzung einen Eintrag in der Datei `Abkuerzungen.tex` und referenzieren sie ihn mittels `\gls`; diese tauchen nach einem Lauf mit `latexmk` im Abkürzungsverzeichnis auf. Beispiel:

Definition in `Abkuerzungen.tex`: `\newacronym{abk}{Abk.}{Abkürzungen}`

Referenzierung: `\gls{abk}`

Für weitere Informationen zu Glossaren und Abkürzungen siehe die Dokumentation des Pakets `glossaries` und die entsprechenden Abschnitte in den Vorlagendateien.

2.2.3. Aufzählungen

- Dies ist die Standardaufzählung
 - Dies ist die nächste Ebene der Aufzählung

2.2.4. Nummerierungen

1. Erster Punkt der Nummerierungen
 - a. Unterpunkt der Nummerierungen

Abbildungsverzeichnis

Abbildung 1 an observation tree simulates a Mealy machine	3
---	---

Abkürzungsverzeichnis

A

Abk. - Abkürzungen	4
--------------------------	---

Tabellenverzeichnis

Tabelle 1 Beschreibung.....	9
Tabelle 2	9
Tabelle 3	9
Tabelle 4	10
Tabelle 5	10
Tabelle 6	10

Tabellenvarianten

Überschrift Tabelle 1

Spalte 1	Spalte 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 1 Beschreibung

Überschrift Tabelle 2

Spalte 1	Spalte 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 2

Überschrift Tabelle 3

Spalte 1	Spalte 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 3

Tabellenvarianten 2

Überschrift Tabelle 1

Spalte 1	Spalte 2
Nummer 1, mehrzeilig in Schriftgröße 9 pt	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 4

Überschrift Tabelle 2

Spalte 1	Spalte 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 5

Überschrift Tabelle 3

Spalte 1	Spalte 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2
Nummer 1	Nummer 2

Tabelle 6