

Programs for DFAs: 1

Michael R. Hansen

`mrh@imm.dtu.dk`

Informatics and Mathematical Modelling
Technical University of Denmark

Overview

- SML programs accepting the languages of DFAs
 - structure of the programs follows the structure of the DFAs
- systematic (automatic) derivation of programs from DFAs
 - correctness proof using structural induction on lists

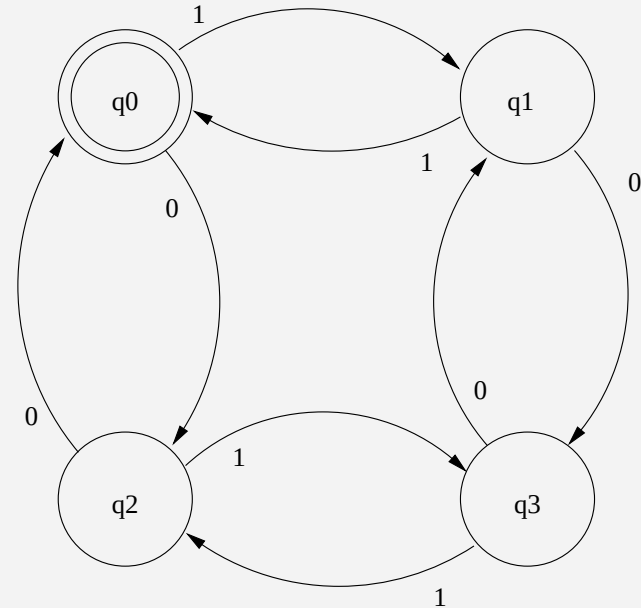
An Example: Even number of 0s and 1s

```
fun q0(0::xs) = q2 xs
  | q0(1::xs) = q1 xs
  | q0 []      = true
```

```
and q1(0::xs) = q3 xs
  | q1(1::xs) = q0 xs
  | q1 []      = false
```

```
and q2(0::xs) = q0 xs
  | q2(1::xs) = q3 xs
  | q2 []      = false
```

```
and q3(0::xs) = q1 xs
  | q3(1::xs) = q2 xs
  | q3 []      = false
```



Answer from the SML system:

```
val q0 = fn: int list -> bool
val q1 = fn: int list -> bool
val q2 = fn: int list -> bool
val q3 = fn: int list -> bool
```

Sample runs

```
- q0 [1, 0, 1, 0, 1, 0, 1, 1, 0];
```

```
> val it = false : bool
```

```
- q0 [];
```

```
> val it = true : bool
```

```
- q0 [0, 0, 1, 1, 0, 0];
```

```
> val it = true : bool
```

```
- q1 [0,1,0,1,0,1,1,0];
```

```
> val it = false : bool
```

For which strings xs does q evaluate to true, i.e.

$q\ xs \rightsquigarrow \text{true} \quad ?$

Constructing programs from DFAs: senario

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$.

The task is to declare mutually recursive functions:

$$q : \Sigma^* \rightarrow \text{bool}, \quad \text{for } q \in Q$$

satisfying that for all $xs \in \Sigma^*$:

$$\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)$$

Constructing programs from DFAs: senario

Given DFA $M = (Q, \Sigma, \delta, q_0, F)$.

The task is to declare mutually recursive functions:

$$q : \Sigma^* \rightarrow \text{bool}, \quad \text{for } q \in Q$$

satisfying that for all $xs \in \Sigma^*$:

$$\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)$$

In particular we have

$$q_0 \text{ } xs \rightsquigarrow \text{true} \iff xs \in L(M)$$

The construction for $M = (Q, \Sigma, \delta, q_0, F)$

Let $\Sigma = \{a_1, \dots, a_k\}$, $Q = \{q_0, \dots, q_n\}$. The program has the form:

fun	q_0	$\dots$	$k + 1$ clauses for q_0
and	q_1	$\dots$	$k + 1$ clauses for q_1
	$\vdots$		
and	q_n	$\dots$	$k + 1$ clauses for q_n

The construction for $M = (Q, \Sigma, \delta, q_0, F)$

Let $\Sigma = \{a_1, \dots, a_k\}$, $Q = \{q_0, \dots, q_n\}$. The program has the form:

```
fun  q0  ...      k + 1 clauses for q0
and  q1  ...      k + 1 clauses for q1
  ⋮
and  qn  ...      k + 1 clauses for qn
```

where the $k + 1$ clauses for $q \in Q$ are

```
q(a1 :: xs) = p1 xs      where p1 = δ(q, a1)
  ⋮
| q(an :: xs) = pn xs      where pn = δ(q, an)
| q []         = b           where b = { true   if q ∈ F
                                         false  otherwise
```

Correctness of the construction

We must prove that for all $xs \in \Sigma^*$:

$$\underbrace{\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)}_{P(xs)}$$

Correctness of the construction

We must prove that for all $xs \in \Sigma^*$:

$$\underbrace{\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)}_{P(xs)}$$

Method: Use induction on the structure of lists, i.e. establish

1. $P([])$ base case
2. $P(xs) \implies P(a :: xs)$, for every $a \in \Sigma$ induction step

Correctness of the construction

We must prove that for all $xs \in \Sigma^*$:

$$\underbrace{\forall q \in Q. (q \text{ xs } \rightsquigarrow \text{true} \iff \hat{\delta}(q, \text{xs}) \in F)}_{P(xs)}$$

Method: Use induction on the structure of lists, i.e. establish

1. $P([])$ base case
2. $P(xs) \implies P(a :: xs)$, for every $a \in \Sigma$ induction step

Specialization of well-founded induction using **suffix ordering**, i.e.

$$xs \prec ys \iff \exists s_1, \dots, s_n. (s_1 :: \dots :: s_n :: xs = ys)$$

Proof of base case: $P([])$

We must establish $\forall q \in Q. (q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F)$.

Consider arbitrary $q \in Q$.

We must show $q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F$.

Proof of base case: $P([])$

We must establish $\forall q \in Q. (q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F)$.

Consider arbitrary $q \in Q$.

We must show $q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F$.

Two cases

$q \in F$:

- $q [] \rightsquigarrow \text{true}$, as $q [] = \text{true}$ is a clause in the program
- $\hat{\delta}(q, []) = q \in F$, by def. of $\hat{\delta}$

$q \notin F$:

Proof of base case: $P([])$

We must establish $\forall q \in Q. (q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F)$.

Consider arbitrary $q \in Q$.

We must show $q [] \rightsquigarrow \text{true} \iff \hat{\delta}(q, []) \in F$.

Two cases

$q \in F$:

- $q [] \rightsquigarrow \text{true}$, as $q [] = \text{true}$ is a clause in the program
- $\hat{\delta}(q, []) = q \in F$, by def. of $\hat{\delta}$

$q \notin F$:

- $q [] \rightsquigarrow \text{false}$, as $q [] = \text{false}$ is a clause in the program
- $\hat{\delta}(q, []) = q \notin F$, by def. of $\hat{\delta}$

Proof of inductive step

For arbitrary $a \in \Sigma, xs \in \Sigma^*, q \in Q$, we must establish

$$P(xs) \implies (q(a :: xs) \rightsquigarrow \text{true} \iff \hat{\delta}(q, a :: xs) \in F)$$

Assume the induction hypothesis $P(xs)$:

$$\underline{\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)}$$

Proof of inductive step

For arbitrary $a \in \Sigma, xs \in \Sigma^*, q \in Q$, we must establish

$$P(xs) \implies (q(a :: xs) \rightsquigarrow \text{true} \iff \hat{\delta}(q, a :: xs) \in F)$$

Assume the induction hypothesis $P(xs)$:

$$\underline{\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)}$$

Let $p = \delta(q, a)$. We have

- $q(a :: xs) \rightsquigarrow p \text{ } xs$, as $q(a :: xs) = p \text{ } xs$ is a program clause
- $\hat{\delta}(q, a :: xs) = \hat{\delta}(p, xs)$, by def. and properties of $\hat{\delta}$

Proof of inductive step

For arbitrary $a \in \Sigma, xs \in \Sigma^*, q \in Q$, we must establish

$$P(xs) \implies (q(a :: xs) \rightsquigarrow \text{true} \iff \hat{\delta}(q, a :: xs) \in F)$$

Assume the induction hypothesis $P(xs)$:

$$\underline{\forall q \in Q. (q \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, xs) \in F)}$$

Let $p = \delta(q, a)$. We have

- $q(a :: xs) \rightsquigarrow p \text{ } xs$, as $q(a :: xs) = p \text{ } xs$ is a program clause
- $\hat{\delta}(q, a :: xs) = \hat{\delta}(p, xs)$, by def. and properties of $\hat{\delta}$

Using the induction hypothesis (q as p) we obtain the desired result

$$q(a :: xs) \rightsquigarrow p \text{ } xs \rightsquigarrow \text{true} \iff \hat{\delta}(q, a :: xs) = \hat{\delta}(p, xs) \in F$$