Chapter 7

Tic-tac-toe

J. W. Myers

Models of Sequential Systems

A Simple Sequential Circuit

Verification/Testing

Logic/Timing Optimization

State Encoding

State Minimization

Simplified FSM Design Flow

State Transition Graph for Previous Circuit
\[
x^*y = (x^s y)^\gamma
\]
\[
x + 1^s = (x^s y) \theta
\]
\[
x^s + x^s y + x^s \gamma = (x^s y)^\gamma
\]

State Assignment

**FSM with Redundant States**

<table>
<thead>
<tr>
<th>State Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
</tr>
</tbody>
</table>

**State Assignment**

<table>
<thead>
<tr>
<th>State Assignment</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
</tr>
</tbody>
</table>

**Finite State Machines**

- **Machines**
  - O is the output function (or choice for output)
  - O is the output alphabet
  - S is the set of initial (reset) states
  - S is the next-state function
  - S is the (null) non-empty set of states

**Finite State Machine**

\[
\begin{cases}
  O & S \\
  I \times S & V
\end{cases}
\]
Example of a State Transition Graph

Table: Representations of FSMs

Example of Incompletely Specified FSM
FSM Minimization for Compeletely Specified Machines

Theorem 7.4.2

Let \( S \) and \( S' \) be the \( x \)-successors of \( s \) and \( s' \), respectively.

**Definition 7.4.1**

Two states \( s \) and \( s' \) are \( x \)-equivalent if and only if \( \forall x \in X, \; \) there does not exist a distinguishing sequence of length \( \leq 1 \).

**Proposition 7.4.3**

We also denote the binary relation \( \equiv_x \) as \( S \times S \subseteq (1/s) \equiv_x (s') \subseteq (1/s') \) by \( \equiv_x \).
The simple FSM after state minimization

The STG of a simple FSM

Flow Table for a Complexly Specified Memory Machine

The STG of an FSM Where All States are Equivalent
A Simple Undirected Graph

Graph Algorithms for FSM Traversal

Shortest paths
Finding strongly connected components (SCC)
Depth first search
Breath first search

A Directed Graph and its SCCs
(1 - γ)n \cdot \beta = f \cdot \text{Post} \cdot \text{Pre}

\text{while} \{ \text{Post} = \text{Post} + 1 \} \{ \text{Pre} = \text{Pre} + 1 \}

\text{Predecessor(Pre)} = \text{Predecessor} \text{ Post}\n
\text{Procedure BFS}(V, E)

(a) C \cup (a \cdot n) \in E \quad = \quad ((a) C, (a \cdot n) \in E)

\text{Neighbor}(a) \quad = \quad \{(a) C \in E \mid (a) C \in E \} \quad \cap \quad a

\text{Example}

BFS
Shortest Path Example

\[
\text{return } G \text{ if } \forall a \in V, \text{ Reachable} = a \text{ or } (d(a', A) \text{ if } A \in v) \text{ and } (\neg a) \text{ hold}
\]

\[
\forall a \in V, \text{ Reachable} = a \text{ or } (d(a', A) \text{ if } A \in v) \text{ and } (\neg a) \text{ hold}
\]

\[
\forall a \in V, \text{ Reachable} = a \text{ or } (d(a', A) \text{ if } A \in v) \text{ and } (\neg a) \text{ hold}
\]