Finite-State Machines

A finite-state recognizing machine is described by:

- A finite set of states
- A finite set of input symbols (input alphabet)
- A transition function δ which assigns a new state to every combination of state and input
- A subset of states designated as accepting states
- A state designated as the starting state

The transition function δ defines a new state S_{new} in terms of a current state S_{old} and a current input symbol x.

A finite-state processing machine is a finite-state recognizing machine that exits to a specified routine on reaching the end of its input. Extend the input alphabet of the machine that processes the language by the end-marker symbol ⊣, (note: the input alphabet of the machine that recognizes the language remains unchanged).

State Equivalence

State S in finite-state recognizer M is equivalent to state T in finite-state recognizer N if and only if machine M starting in state S will accept exactly the same sequences as machine N starting in state T.

Finite-state recognizers M and N are said to be equivalent if and only if their starting states are equivalent.

If two states are not equivalent then any sequence which causes one state to make a transition into an accepting state and the other state to go into a rejecting state is called a distinguishing sequence.

Two states are equivalent if and only if they have no distinguishing sequence.

State equivalence is:

Reflexive — each state is equivalent to itself
Symmetric — state S equivalent to state T implies state T equivalent to state S
Transitive — if states S and T are equivalent and states T and U are equivalent then states S and U are equivalent

Extraneous states

States in a finite-state recognizer that can never be reached by any possible input sequence when the machine is initially in its starting state are called extraneous states.

To prepare a list of non-extraneous states for any given finite-state machine:

i. Initialize the list with the starting state.
ii. Add to the list all states which can be reached from the starting state under single inputs.
iii. For every new state on the list add any unlisted state which can be reached from this state.

Reduced (Minimal) Machines

A finite-state machine is reduced if it has no extraneous states and if no two states are equivalent to each other.