Yihan Zou (yz2575)

"Principles and methods of testing finite state machines-a survey"
by D. Lee and M. Yannakakis

Pass 3 on "Adaptive Distinguishing Sequence" (ADS)

1. Definition and existence

- Definition: ADS is a rooted tree *T* where internal nodes are input symbols and edges are output symbols. Tree *T* has n leaves, and each of them is a state of FSM. By applying inputs based on previously observed outputs from root of T, we can identify the initial state when we reach certain leave of T.

- Length of sequence = depth (*T*), best possible of length $(\pi^2/12)n^2$ (proved by Sokolovskii)

- A FSM can have no preset distinguish sequence (PDS) but still have ADS

- An input *a* is valid for a set of state C if $\delta(si, a) \neq \delta(sj, a)$ OR $\lambda(si, a) \neq \lambda(sj, a)$. Difference between PDS and ADS: validity w.r.t. all sets v.s. validity w.r.t. a particular set

- Algorithm for determining existence:

  • Maintain a partition π, representing partition of initial states that can be distinguished

  • Each time consider a valid input *a* for a block *B* in π, split the block if: 1) two states in B has different outputs, or 2) two states has same output but mapped to different blocks. If states can be partitioned into discrete sets, the ADS exist for this FSM.

  Example: $\pi = \{\{s_1, s_2\}, s_3\}$ $B=\{s_1,s_2\}$ if $\lambda(s1, a) = \lambda(s2, a)$ but $\delta(s1, a) = s1 \; and \; \delta(s2, a) = s3$ → split : $\pi = \{\{s_1\} \{s_2\}\{s_3\}\}$ (discrete sets)

  • Complexity: straightforward $O(pn^2)$, can be improved to $O(pnlogn)$ (divide & conquer)

2. Construction of ADS

- Consists of two steps: 1) constructing splitting tree (ST), 2) constructing ADS from ST

- ST: internal nodes u_i = a set of states associated with sequence *p*. Edges are output symbols.

- Three types of valid input for a block *B* in current partition:

  i) At least two states produce different outputs,
  ii) All states produce same output but at least two of them are mapped to (next state belongs to) different blocks,
  iii) All states produce same output and mapped to same block.
- Implication graph G_π => I/O relationship between blocks

Yihan Zou (yz2575)

- Splitting Tree Algorithm:

  • Initialization: root labeled the set *S* of all states.

  • ST Expansion: (best illustrate with example..)

  For type i) input: associate input symbol with current node *u*. Each child of *u* corresponds to a set of states with same output.

  For type ii) input: *v* is lowest node contains the set $\delta(B, a)$, then associate string *a*-str(*v*). Each child *w* of *v*, if label(*w*) and $\delta(B, a)$ have common element => create a child of *u* labeled with B $\cap$ $\delta^{-1}(w, a)$.

  For type iii) input: find a path in G_π to another block has type i) or ii) input and expand the tree as for type ii)

  • Example: (found a mistake?) block u_4 has type ii) input *b* instead of type ii), (page 11)

- ADS algorithm using ST:

  • Initialization: $I = C$

  • Construct ADS as follows until certain leaf is reached: find lowest node *u* whose label contains set C, apply str(*u*) and update I and C.

  • Example: a->aba-> ba