Büchi acceptance.

Thus an LSC is also software.

We will give a

•

for LSCs.

σ is a set of computation paths.

σ₁ α₁ α₂ is

T is a set of states and transitions.

The Big Picture

Cuts and Firedsets, automaton construction

•

Excursion: automata accepting infinite words

Formalise this positive scenario/anti-scenario/requirement using LSCs.

Which are the cuts and firedsets of this LSC?

Capabilities for following tasks/questions.

Educational Objectives:
LSC Semantics: TBA Construction

Example

Run of TBA

From Finite Automata to Symbolic Büchi Automata
Definition. Let \((L, \preceq, \sim)\), \(I\), \(\text{Msg}\), \(\text{Cond}\), \(\text{LocInv}\), \(\Theta\) be an LSC body. A non-empty set \(\emptyset \neq C \subseteq L\) is called a cut of the LSC body iff

- \(C\) is downward closed, i.e. \(\forall l, l' \in L \cdot l' \in C \land l \preceq l' \implies l \in C\),
- \(C\) is closed under simultaneity, i.e. \(\forall l, l' \in L \cdot l' \in C \land l \sim l' \implies l \in C\), and
- \(C\) comprises at least one location per instance line, i.e. \(\forall I \in I \cdot C \cap I \neq \emptyset\).

The temperature function is extended to cuts as follows:

\[\Theta(C) = \begin{cases} \text{hot} & \text{if } \exists l \in C \cdot (\nexists l' \in C \cdot l \prec l') \land \Theta(l) = \text{hot} \\ \text{cold} & \text{otherwise} \end{cases}\]

This is equivalent to saying that \(C\) is hot if and only if at least one of its maximal elements is hot.

A Successor Relation on Cuts

The partial order "\(\preceq\)" and the simultaneity relation "\(\sim\)" of locations induce a direct successor relation on cuts of \(L\) as follows:

Definition. Let \(C \subseteq L\) be a cut of an LSC body \((L, \preceq, \sim, I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)\). A set \(\emptyset \neq F \subseteq L\) is called a fired-set of \(C\) if and only if

- \(C \cap F = \emptyset\) and \(C \cup F\) is a cut, i.e. \(F\) is closed under simultaneity,
- all locations in \(F\) are direct \(\prec\)-successors of the front of \(C\), i.e. \(\forall l \in F \exists l' \in C \cdot l' \prec l \land (\nexists l'' \in C \cdot l' \prec l'')\),
- locations in \(F\), that lie on the same instance line, are pairwise unordered, i.e. \(\forall l \neq l' \in F \cdot (\exists I \in I \cdot \{l, l'\} \subseteq I) \implies l \not\preceq l' \land l' \not\preceq l\),
- for each asynchronous message reception in \(F\), the corresponding sending is already in \(C\), i.e. \(\forall (l, E, l') \in \text{Msg} \cdot l' \in F \implies l \in C\).

The cut \(C' = C \cup F\) is called the direct successor of \(C\) via \(F\), denoted by \(C \Rightarrow F\).
A full LSC $L = ((L, \preceq, \sim), I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)$ consists of:

- a body $(L, \preceq, \sim, I, \text{Msg}, \text{Cond}, \text{LocInv}, \Theta)$,
- an activation condition $ac_0 \in \Phi(C)$,
- a strictness flag $\text{strict}$ (otherwise called permissive)
- an activation mode $am \in \{\text{initial}, \text{invariant}\}$,
- a chart mode $\Theta L = \text{cold}$ or $\text{hot}$.

Concrete syntax:

$LSC: L_1 AC: c_1 AM: \text{initial} I: \text{permissive}
I_1 I_2 I_3 E F G$

References: