Excursion: Büchi Automata

From Finite Automata to Symbolic Büchi Automata

\[ A : \Sigma = \{0, 1\} \]

\[ B : \Sigma = \{0, 1\} \]

\[ B' : \Sigma = \{0, 1\} \]

\[ q_1 \xrightarrow{0,1} q_2 \]

\[ even(x) \]

\[ odd(x) \]

\[ A_{sym} : \Sigma = \{x \rightarrow N\} \]

\[ B_{sym} : \Sigma = \{x \rightarrow N\} \]

Symbolic Büchi Automata

**Definition.** A Symbolic Büchi Automaton (SBA) is a tuple \( B = (\text{Expr}^B(X), X, Q, q_{ini}, \rightarrow, Q_F) \) where
- \( X \) is a set of logical variables,
- \( \text{Expr}^B(X) \) is a set of Boolean expressions over \( X \),
- \( Q \) is a finite set of states,
- \( q_{ini} \in Q \) is the initial state,
- \( \rightarrow \subseteq Q \times \text{Expr}^B(X) \times Q \) is the transition relation. Transitions \((q, \psi, q')\) from \( q \) to \( q' \) are labelled with an expression \( \psi \in \text{Expr}^B(X) \).
- \( Q_F \subseteq Q \) is the set of fair (or accepting) states.

**Definition.** Let \( X \) be a set of logical variables and let \( \text{Expr}^B(X) \) be a set of Boolean expressions over \( X \). A set \((\Sigma, \cdot |\cdot = \cdot \cdot)\) is called an alphabet for \( \text{Expr}^B(X) \) if and only if
- for each \( \sigma \in \Sigma \),
- for each expression \( \text{expr} \in \text{Expr}^B(X) \), and
- for each valuation \( \beta : X \rightarrow D(X) \) of logical variables,
  either \( \sigma \mid\cdot =\beta \text{expr} \) or \( \sigma \not\mid\cdot =\beta \text{expr} \).

An infinite sequence \( w = (\sigma_i)_{i \in \mathbb{N}} \in \Sigma^\omega \) over \((\Sigma, \cdot |\cdot = \cdot \cdot)\) is called word (for \( \text{Expr}^B(X) \)).
Concrete syntax: Live Sequence Charts — Full LSC Semantics

Note: Activation Condition
Universal LSC: Example

LSC: buy water
AC: true
AM: invariant
I: strict

User CoinValidator

cp: ChoicePanel
Dispenser

C 50pWATER
¬ (C50 ! ∨ E1 ! ∨ pSOFT ! ∨ pTEA ! ∨ pFILLUP !)

cp -> water_in_stock
dWATER
OK

Universal LSC: Example

LSC: buy softdrink
AC: true
AM: invariant
I: permissive

User V end. Ma.
E 1 pSOFT

LSC: get change
AC: true
AM: invariant
I: permissive

User V end. Ma.
C 50 E 1 pSOFT

¬ C50 ! ∧ ¬ E1 !
false

• Existential LSCs
∗ may hint at test-cases for the acceptance test

\(\ast\): as well as (positive) scenarios in general, like use-cases
Activation mode: Universion is accepted by the constructed TBA, at least on word (of the model).

Existential accepts An LSC

Mathematics (N, E, f)

$\mathbb{S} = (\mathbb{S}, \epsilon, \pi, 0, \mathbb{S})$

Fully infinitely often

if there

$\mathbb{S} = (\Sigma, \mathbb{M}, \mathbb{N}, \mathbb{E}, \mathbb{F}, \mathbb{A})$

accept

signal / attribute expressions

Annotations use

Locations induce a partial order on cuts.

The semantics in this model is:

(iv) define language

LSC: buy soft drink

AC: get change

AM: invariant

I: permissive

LSC: get change

AC: get change

AM: invariant

I: permissive

Note: Sequence Diagrams and (Acceptance) Test