SAND2017-4962PE

Compiling Statecharts into Why3

Rob Armstrong, Jon Aytac, Geoff Hulette, Jackson Mayo,
Karla Morris

Sandia National Laboratories is a multimission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC., a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energys National Nuclear Security Administration under contract DE-NA0003525.

Compiling Statecharts into Why3

- The subcategory $\mathcal{E}$ of reactive hierarchical statecharts allowing communication strictly between consecutive layers in the hierarchy may, following Argos, be given a semantics $F : \mathcal{E} \to \mathcal{M}$ in terms of Mealy machines $\mathcal{M}$ and operations (parallel composition, encapsulation, and refinement) thereon.
- Given an alphabet $\mathcal{A}$ with formulae $\mathcal{B}(\mathcal{A})$, Mealy Machines are tuples(S, s_0, I, O, T) of states S, initial states s_0, inputs $I \subset \mathcal{A}$, outputs $O \subset \mathcal{V}$, and transitions $T \subset S \times \mathcal{B}(I) \times \mathbb{B}^O \times S$
- The indicator function of the set of transitions T gives a Boolean predicate, so that the Mealy Machine may be readily expressed as a theory in Why3.

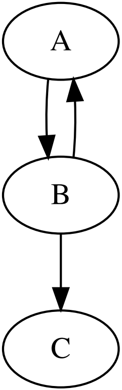
Mealy Machines in Why3

- The indicator function of T gives a Boolean predicate P_T, so that the Mealy Machine may be readily expressed as a theory in Why3. Allowing non-determinism, let $\text{next} : S \to \mathcal{P}(S)$ a map from the set of states to the set of sets of states. Then

$$P_T = \bigvee_{s \in S} \bigvee_{(s,b,\bullet,s')} \big(s \wedge b \wedge \big(\text{next} \ni s'\big)\big)$$

Mealy Machines in Why3

- Consider the simple example



- the why3 listing for this simple chart in the follows:

```
theory KarlaTypeDefs
  use import set.Set as S
  type states = AA | BB | CC
  function next states : S.set states
end
theory KarlaTransitions
  use import KarlaTypeDefs
  axiom nextStep: forall s:states.
  ((s=AA) /\ (next s = (add BB empty)))
      \/ ((s=BB) /\ (next s = (add CC (add AA empty))))
      \/ ((s=CC) /\ (next s = (add CC empty)))
end
theory KarlaPropery
  use import KarlaTransitions
  predicate ccFixedPoint =
    ((next CC) = (add CC empty))
  goal ccFixedPointProperty =
    ccFixedPoint
end
```

Moving Forward

- Argos semantics gives a map[1] into Mealy Machines
- Compiling Mealy Machines as Boolean constraints into Why3 is fairly straightforward
- When we later augment Argos Semantics with variables, WhyML allows great flexibility, extending FOL while preserving tractability

[1] a monoidal functor, even