Trace generation problem

Clafer transition system TS is a tuple $(S, \rightarrow)$, where

- S - All top level Clafers and their configurations
- $\rightarrow$ - Transition Relation

Problem: Given temporal formula φ find traces of TS that satisfy φ.

Input and output

Input

- S - all possible configurations constrained by structural constraints
- $\rightarrow$ - All $S \times S$

Output

- Relation of allowed transitions (subset of $S \times S$)
- Traces of possible executions $s_0 s_1 \ldots \models \varphi$

Using Amir's library to get traces

- Set up *State* signature that captures transition system states.
- Include *temporal_logics/ctl* (or *ctlfc*) library
- Define a model checking assertion that checks for $\neg\varphi$
- Execute and collect counterexample execution

Model checking finds counterexamples $\sigma \not\models \neg\varphi$. So counterexample itself is a valid trace $\sigma \models \varphi$.

Simple pacemaker in Alloy

```
open util/integer
open temporal_logics/ctl[State]
sig ID {}
sig PM {id : ID, s : lone  PMStatus}{one this[pm]}
abstract sig PMStatus  {}
one sig ASensingTimeout , APace, ASense, ARecovery, SensingAPulse extends PMSta
sig State { pm : one PM }
fact TransitionRelation {
-- all s, s' : State | s' in nextState[s]
}
fact { no disj s, s': State | some s.pm & s'.pm}
assert MC{
CTL_MC[not_ctl[ AG[implies_ctl[pm.s.SensingAPulse, or_ctl[AX[pm.s.ASense],  AX[
 CTL_MC[not_ctl[ AG[implies_ctl[pm.s.ASensingTimeout, AX[pm.s.APace] ] ] ]  ]
 CTL_MC[not_ctl[ AG[implies_ctl[pm.s.APace, AX[pm.s.ARecovery] ] ] ]  ]
 CTL_MC[not_ctl[ AG[implies_ctl[pm.s.ASense, AX[pm.s.ARecovery] ] ] ]  ]
}

check MC for 10 State, 10 PM, 4 ID, 10 PMStatus
```

Problems

Capturing the trace of the counterexample.