

Modeling TSAFE in Alloy

Gregory Dennis
October 26, 2001

Domains

sig Tick{nextTick: option Tick}

"clock ticks" - instances in time

sig Point { }

3D points in space

sig Path {points: set Point}

3D "tube" in space

sig Trajectory {location: Tick ->! Point}

4D cueyeach maps time to Points

sig Flight {plannedPath: Path}

every flight has a static planned path

sig State {

state of TSAFE at a given tick

time0, time1: Tick,

time of state, time of next state

position: Flight ->! Point,

positions of all flights

headingPosition: Flight ->! Point,

positions flights are heading

probingTraj: Flight -> Trajectory,

traj's to probe along for each flight

conflicts: Flight -> Flight

maps conflicting flights to one another

}

Functions

```
fun onHeadingPosition(p: Path, pos: Point) : Point
```

returns the point a flight at Point pos would need to be heading to rejoin the Path p

```
fun State.onHeading(f: Flight)
```

true if the flight's headingPosition equals the onHeadingPosition

```
fun State.onTrack(f: Flight)
```

true if the flight's position is in its planned path

```
fun State.drTrajSynth(f: Flight) : Trajectory
```

returns a trajectory whose first two points are the flight's position and headingPosition respectively

```
fun State.nomTrajSynth(f: Flight) : Trajectory
```

same as drTrajSynth except this trajectory rejoins plannedPath at some point

```
fun State.conformanceMonitor() : (Flight -> Trajectory)
```

assigns probing traj's to flights according to whether they are onTrack and onHeading

```
fun State.conflictProbe() : (Flight -> Flight)
```

searches for intersections between flights' probing trajectories

Assertion

- Static set of flights in the air
- Model simulates an infinite look-ahead time.
- Can new conflicts appear from one state to the next?

```
fun nextState(s, s' : State) {  
  s'.time0 = s.time1  
  s'.position = s.headingPosition  
}
```

- Transitions one state from the next
- Next state exists at the next clock tick
- Flights move to their headingPosition

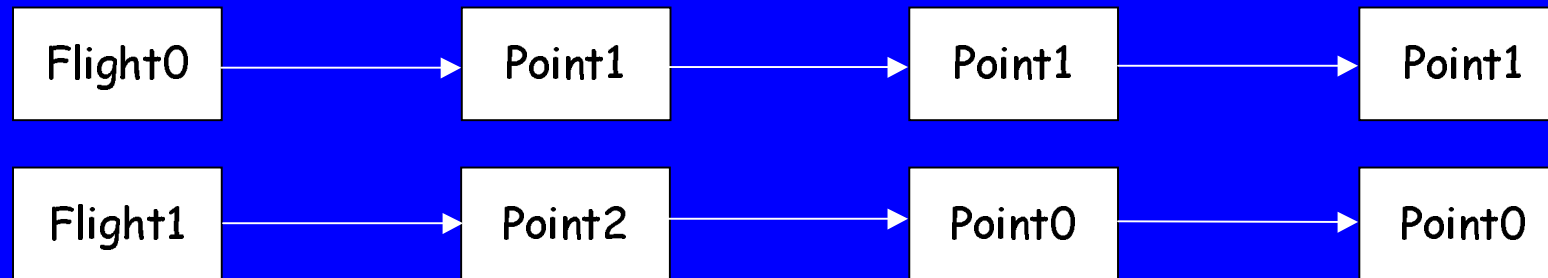
```
assert conflictsSame {  
  al s, s' : State | nextState(s, s') =>  
    s.conflicts = s'.conflicts  
}
```

- If we transition from one state to the next, do the conflicts change?

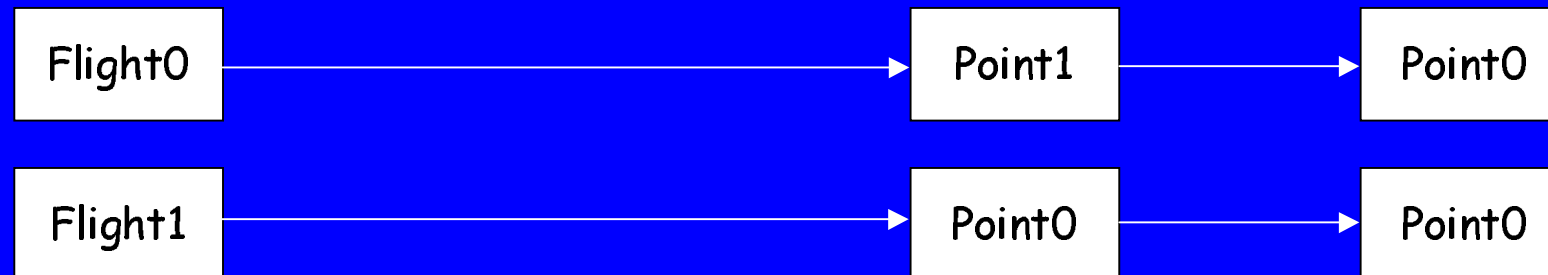
Results (not really)

Counterexample Found!

Probing Trajectories Before - No Conflicts



Probing Trajectories After - Conflict!



What's Next

- probing trajectories static => conflictsSame
- Introduce lag
 - flights move, but TSAFE retains old trajectory data
- Allow for a dynamic set of flights
 - flights go in and out of view
- Get some real results