Modular Relative Time
Petri Nets

(or: how to avoid being
ticked by tocks)

Contents

• State Space Explosion
• Modular Nets
• Relative Time Nets
• Modular Relative Time Nets
• The Infinite Procrastinator
• Conclusion
State Space Explosion

- non-trivial problems x quality requirements
  = exhaustive searches of complex solutions
- Potential Approaches:
  - smaller models
  - more hardware
  - data handling schema
  - *more efficient models*

Modular Nets

- Modular re-use is *highly* desirable
- However each different combination runs the risk of unexpected side-effects
- But the more independent the modules are, the worse the explosion becomes!
- Solution: model the independent parts separately as much as possible
Relative Time Nets

- Time is a critical part of many applications
  - eg. protocol time-out is an important property
- But monotonically increasing time can make a finite ‘untimed’ model infinite!
- Solution:
  - search the states for congruencies; or
  - build the states so they are always exposed
Modular Relative Time Nets

- What happens when we combine both?
- Can “modularity” support “global time”?
- Probably not completely, but we can try…
  - local states for each module explored separately
  - each module also has a global clock offset
  - modules synchronise only when the fused events occur at the same global time
MRTPNs – cont’d

<insert MRTPN graphs here>

The Infinite Procrastinator

• Relative time good for *lots* of activity, not so good with *minimal* activity
• MRTPNs have the same weakness – don’t help as much if certain modules get lonely
• Solution:
  – focus on state exploration, some arcs ignored
  – extend the formalism to allow fairness
  – look for omega limits on entire modules
Infinite Procrastinator – cont’d

<insert pathological graphs here>

Conclusion

- We want to find out if finite ‘untimed’ models can be finite timed models as well
- We want to use modular building blocks
- But there are problems using both at once
- So the question becomes how much we can do with both, and/or when does it become less efficient then the other techniques?