Goals

Communication and Concurrency: Introduction

Colin Stirling (cps)
School of Informatics
16th September 2013

 Goals

- **Modelling**: a notation for describing concurrent systems (CCS)
- **Equivalence**: when two descriptions are the same system
- **Properties**: modal and temporal properties of systems.
Goals

- **Modelling:** a notation for describing concurrent systems (CCS)
- **Equivalence:** when two descriptions are the same system
- **Properties:** modal and temporal properties of systems.
- **Model checking:** algorithmic techniques for checking equivalence and properties.

**Software tools:** automatically checks properties and equivalence
**Specification: Temporal Properties**

- Mutual exclusion
- Absence of deadlock
- Absence of starvation

**CCS model of Peterson's solution**

\[
\begin{align*}
B_{1f} &= \delta_{t1f}.B_{1f} + b_{1wf}.B_{1f} + b_{1tf}.B_{1t} \\
B_{1t} &= \delta_{t1t}.B_{1t} + b_{1wt}.B_{1t} + b_{1wf}.B_{1f} \\
B_{2f} &= \delta_{t2f}.B_{2f} + b_{2wf}.B_{2f} + b_{2tf}.B_{2t} \\
B_{2t} &= \delta_{t2t}.B_{2t} + b_{2wt}.B_{2t} + b_{2wf}.B_{2f} \\
K_1 &= \delta_{t1}.K_1 + k_{1w}.K_1 + k_{1w}.K_2 \\
K_2 &= \delta_{t2}.K_2 + k_{2w}.K_2 + k_{1w}.K_1 \\
P_1 &= \delta_{t1wt}\text{ req1.}[\mathbb{W}]P_{11} \\
P_{11} &= b_{2rt}.P_{11} + b_{2rf}.P_{12} + k_{2r}.P_{11} + k_{1r}.P_{12} \\
P_2 &= \delta_{t2wt}\text{ req2.}[\mathbb{W}]P_{21} \\
P_{21} &= b_{1rf}.P_{22} + b_{1rt}.P_{21} + k_{1r}.P_{21} + k_{2r}.P_{22} \\
P_22 &= \text{enter2.exit2.}[\mathbb{W}]P_2 \\
Peterson &= (P_1 | P_2 | K_1 | B_{1f} | B_{2f}) \setminus L
\end{align*}
\]

**Formalising Temporal Properties**

\[
\begin{align*}
\text{Mutex} &= AG ([\text{exit1}]ff \lor [\text{exit2}] ff) \\
\text{NoDeadlock} &= AG \langle - \rangle tt \\
\text{NoStarvation} &= AG([\text{req1}] AF \langle \text{exit1} \rangle tt) \land \\
&\quad AG([\text{req2}] AF \langle \text{exit2} \rangle tt)
\end{align*}
\]
Model checking

- The Edinburgh Concurrency Workbench
  - A tool for simulating and verifying CCS agents
  - http://homepages.inf.ed.ac.uk/perdita/cwb/
- Proving Peterson’s solution correct
  - Command: checkprop(Peterson,Mutex);
    - true
  - Command: checkprop(Peterson,NoDeadlock);
    - true
  - Command: checkprop(Peterson,NoStarvation);
    - true
Modelling and model checking large (and infinite state) systems
  ▶ **Circuits:** since Pentium-bug Intel uses model checking
  ▶ **Software:** Microsoft prototype software model checking
  ▶ **Life:** cells and pathways (Systems biology: huge new area)
Modelling and model checking large (and infinite state) systems

- **Circuits**: since Pentium-bug Intel uses model checking
- **Software**: Microsoft prototype software model checking
- **Life**: cells and pathways (Systems biology: huge new area)

Paper on hardware verification and one on BLAST tool for software verification on course web page

Look up “model checking” in Wikipedia, Google, ...