We analyze reactive software systems consisting of a large number of components that communicate with each other by exchanging messages. Real life examples are tele-communication systems, internet-based systems, embedded systems, etc. Such systems usually possess a huge or even infinite state space, which makes it difficult or impossible to explore systematically the whole state space for checking properties of interest.

**Abstraction**

We abstract from
- program code;
- message orders;
- cycle activation conditions;
- cycle dependencies;

**ILP problem(s)**

\[ x_1 \cdot 0 + x_2 \cdot 1 + x_3 \cdot 1 > 0 \]

Its solutions represent the undesired behavior. So, if no solution exists, then the system satisfies the considered property.

**Counterexamples**

It corresponds to the behavior in which only the cycle c2 is repeated infinitely often.

**Yes, property satisfied**

When necessary, we extract information which cycles must rely on which other cycles for repeated executions. Using such information, we are able to exclude a huge number of unrealistic combinations of cycle executions.

**Refinement**

Based on the discovery of control flow cycle dependencies

**Case study**

PBX – An IBM Telephony Switch System
- 29 classes of components, multiple instances of each class
- More than 75 unbounded message buffers
- State space: much larger than having 10^{46} states

Our analysis to determine communication buffer boundedness
- Runtime: 72 seconds
- #Cycles: 2345
- Size of ILP: 981*294

**Publications**

- TACAS 2006 automated termination proofs – for abstracting Java code on UML RT state machines
- CONCUR 2006 livelock freedom check
- TOOLS-EUROPE 2008 an executable UML-RT semantics in AsmL
- SPIN 2008 dependency analysis for control flow cycles
- TACAS 2004 boundedness test for UML RT models
- SPIN 2004 boundedness test for Promela models
- SPIN 2005 refinement

**Prototype tools**

- IBOC boundedness checker
- aLive livelock freedom checker
- PONES termination prover

A screenshot of the IBOC tool

**DFG Colloquium**

Konstanz

Wei Wei – Postdoc - Associated membership August 1, 2005 – April 3, 2008

Work group: Software Engineering