Software Quality FS 2011

Exercise 1 - Introduction

Cédric Jeanneret

Requirements Engineering Research Group
Department of Informatics
University of Zurich
http://www.ifi.uzh.ch/rerg/people/jeanneret

University of Zurich^UZH

Exercises

Objectives and Formalities

In theory there is no difference between theory and practice. In practice there is.

-- Albert Einstein or Jan L. A. van de Snepscheut or Yogi Berra.

- Necessary condition to pass the module
- 3 assignments (1 week)
- Can be solved in groups of 3

Exercises

Schedule

#	Theme	Release	Due	Discussion
1	Model Checking	Feb 28th	Mar 7th	Mar 14th
2	Test Debugging	Mar 14th	Mar 21st	Mar 28th
3	Planning Quality	Mar 28th	Apr 4th	Apr 18th

Exercises
Daiquiri

Wiki: http://daiquiri.ifi.uzh.ch/trac/swq11/

- Documentation
- Tutorials (written by former students)
- Sample Solutions (written by former students)

Register on this wiki with ***sxxyyyzz*** as username

- SVN, Hudson, Trac (required for exercise 2)

Model Checking

Presentation of SPIN

- Appeared in 1991
- ACM Software System Award in 2001
- A command line tool
  - Eclipse Plugin
  - xspin (TCL)
  - jSpin (Java)
- Requires C pre-processor / compiler
- Available on the macs in the lab rooms

Model Checking

Presentation of SPIN

- Promela Language (non-deterministic)
  - loosely based on Hoare’s CSP and Dijkstra’s guarded commands
- Simulator (Random, Interactive, Replay)
- Exhaustive Verifier
  - Unreachable code
  - Presence of deadlocks
  - Violation of assertions
  - Satisfaction of never claims
  - Presence of non-progress loops
  - Satisfaction of LTL Properties

Colony of Chameleons
Introduction

A colony of chameleons includes 54 individuals

- 20 red
- 18 blue
- 16 green

Whenever two chameleons of different colors meet, each changes to the third color.

Could red chameleons (temporarily) disappear?

Colony of Chameleons

Promela Model

```
#define NRED                (20)
#define NBLUE               (18)
#define NGREEN              (16)

short nRed = NRED;
short nBlue = NBLUE;
short nGreen = NGREEN;

active proctype mutations() { … }
active proctype observer() { … }
```

"C" Macros:
- Constants
- Predicates

Data Types

Global Variables

Communications Channels

Process Declarations

Colony of Chameleons

Mutations Process

```
active proctype mutations()
{
    do
    :: d_step {nRed && nBlue;
            nRed--; nBlue--; nGreen = nGreen + 2;}
    :: d_step {nRed && nGreen;
            nRed--; nGreen--; nBlue = nBlue + 2;}
    :: d_step {nBlue && nGreen;
            nBlue--; nGreen--; nRed = nRed + 2;}
    :: else
    od
}
```

Model Checking

Random / Interactive Simulation

model.pml
spin

spin model.pml

spin -i model.pml

Colony of Chameleons

LTL Formula

Could red chameleons (temporarily) disappear?

LTL Formula:

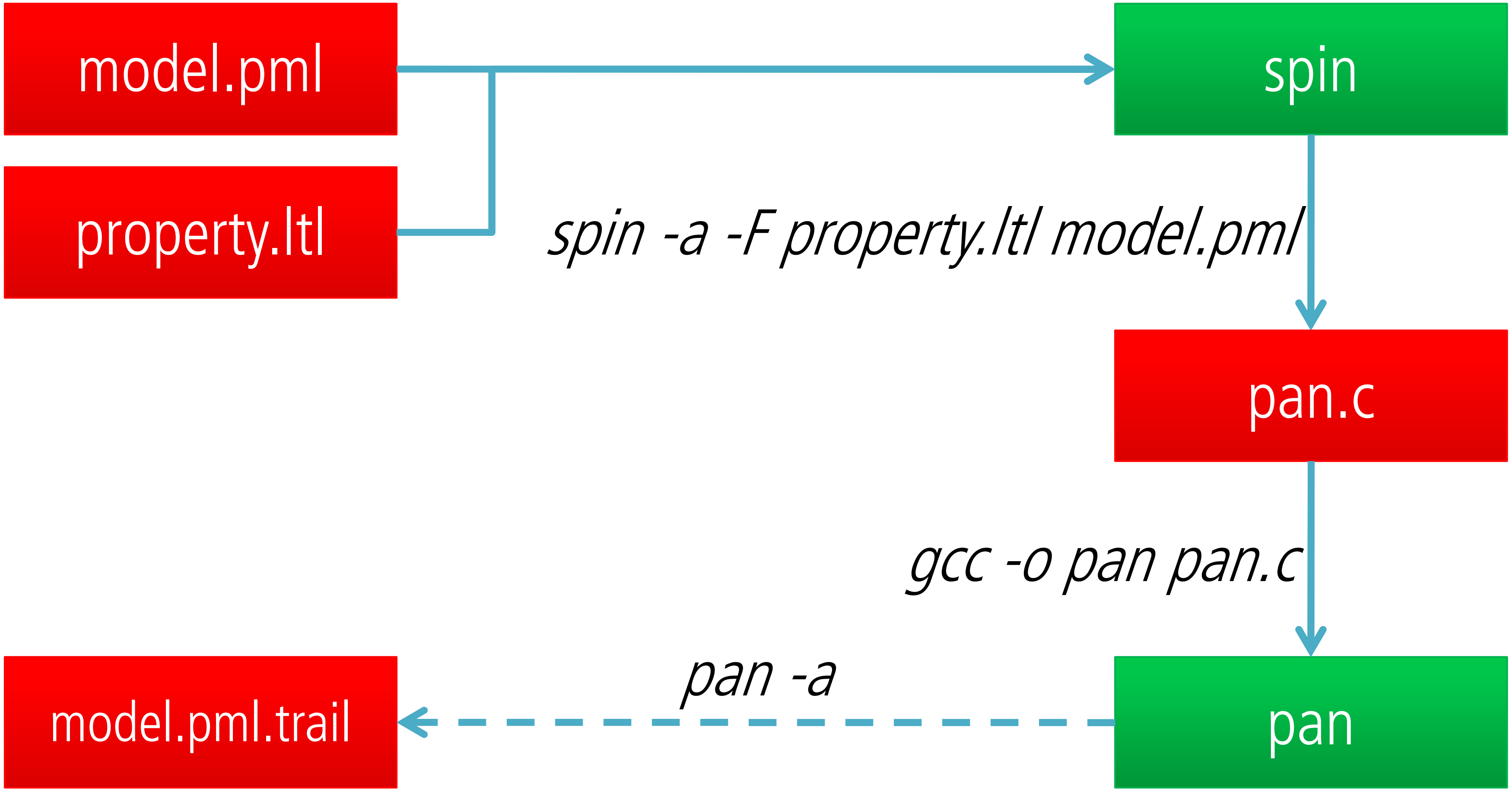
```
<> noRedChameleonLeft
```

Addition to the Promela Model:

```
#define noRedChameleonLeft   (!nRed)
```

Model Checking

Verification



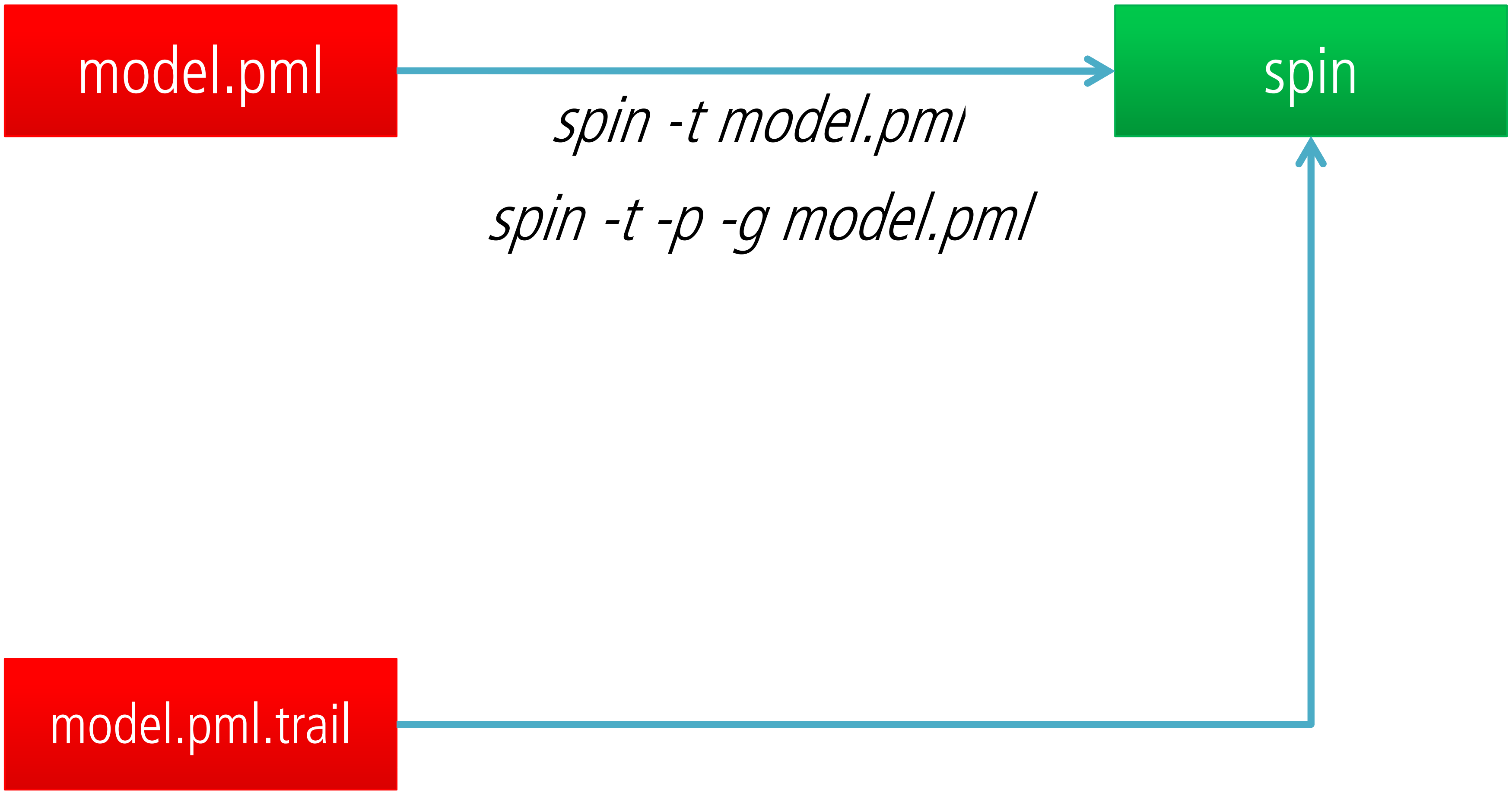
spin -a -F property.ltl model.pml

gcc -o pan pan.c

pan -a

Model Checking

Guided Simulation



spin -t model.pml

spin -t -p -g model.pml