Software Quality FS 2012

Introduction - Exercise 1

Eya Ben Charrada

Exercises
Formalities

- Necessary conditions to pass the module:

  1. Pass 2 of the 3 assignments (Can be solved in groups of 3)

  2. Pass the exam

Exercises
Schedule

#	Theme	Release	Due	Discussion
1	Model Checking	Feb 27th	Mar 7th	Mar 12th
2	Testing Debugging	Mar 12th	Mar 19st	Mar 26th
3	Software Metrics	Mar 26th	Apr 2th	Apr 23th

Model Checking
Presentation of SPIN

- 1980 (Bell labs) – 1991 (freely available)
- Widely used in industries building critical systems
- Simulator and Exhaustive verifier (Unreachable code, deadlocks, violation of assertions, etc)
- Model to be verified written in Promela
- Properties expressed in LTL

Model Checking
Presentation of SPIN

- Command line tool

- Requires C pre-processor / compiler

- Available on the macs in the lab

(room 0.B.04)

Colony of Chameleons
Introduction

A colony of chameleons includes 99 individuals

- 35 red
- 34 blue
- 30 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   (34)
#define NBLUE   (35)
#define NGREEN   (30)

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
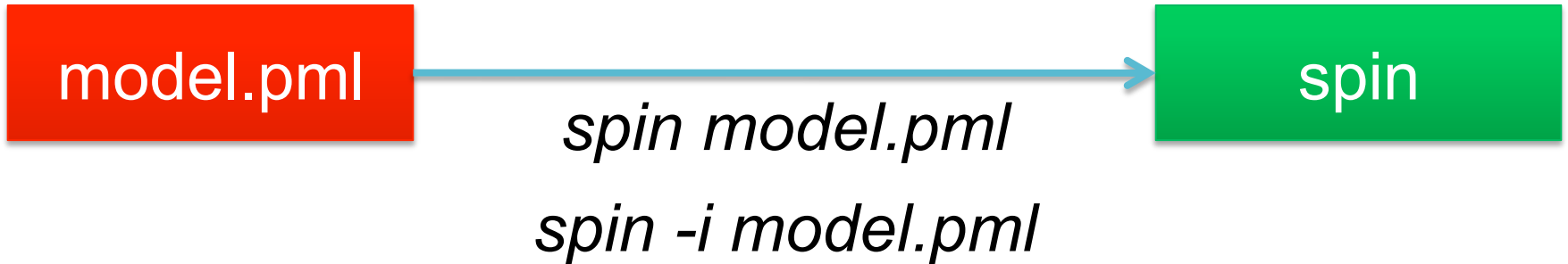
Process Declarations

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
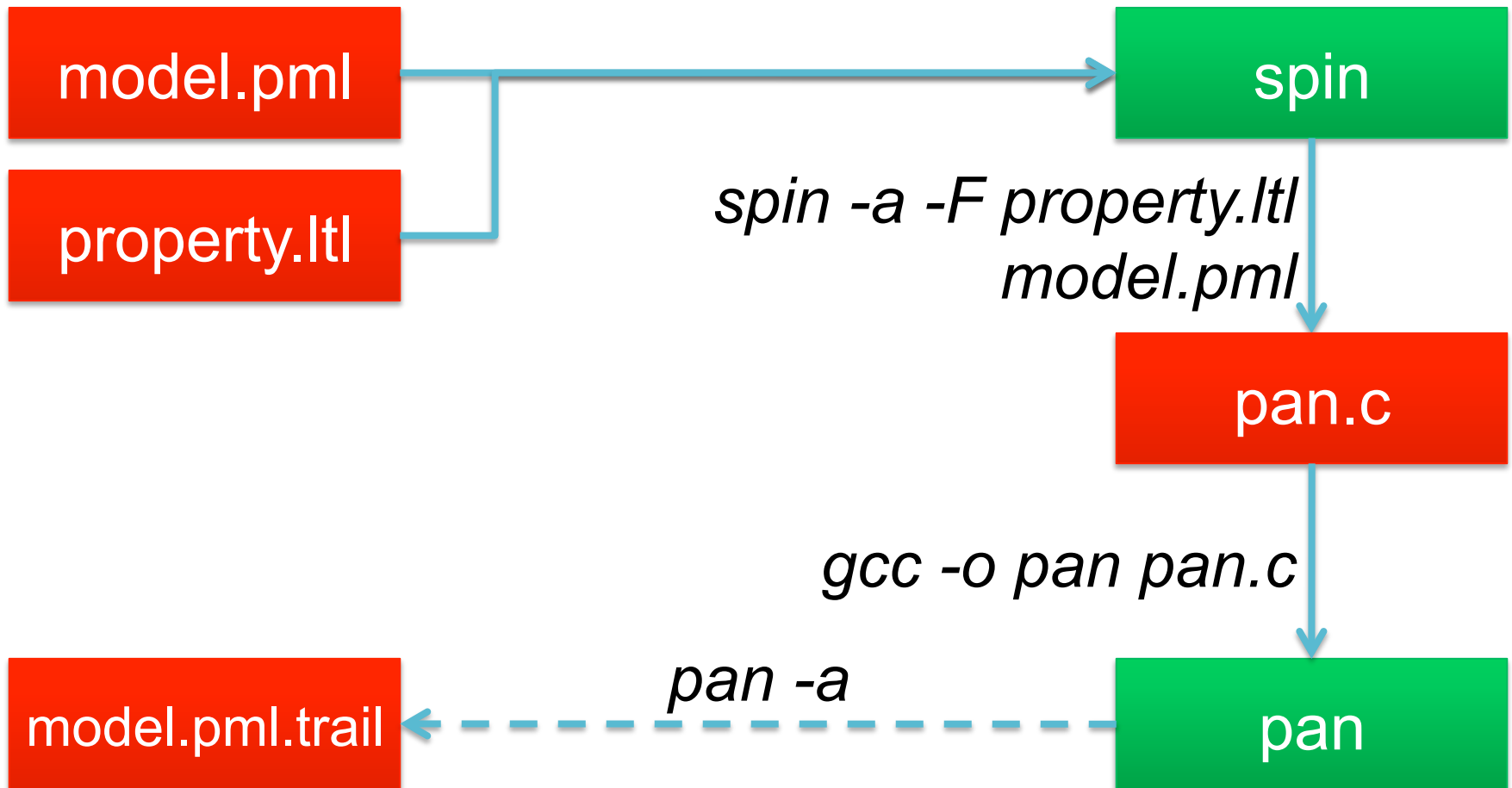
Could red chameleons (temporarily) disappear?

LTL Formula:

    <> noRedChameleonLeft

Addition to the Promela Model:

    #define noRedChameleonLeft   (!nRed)
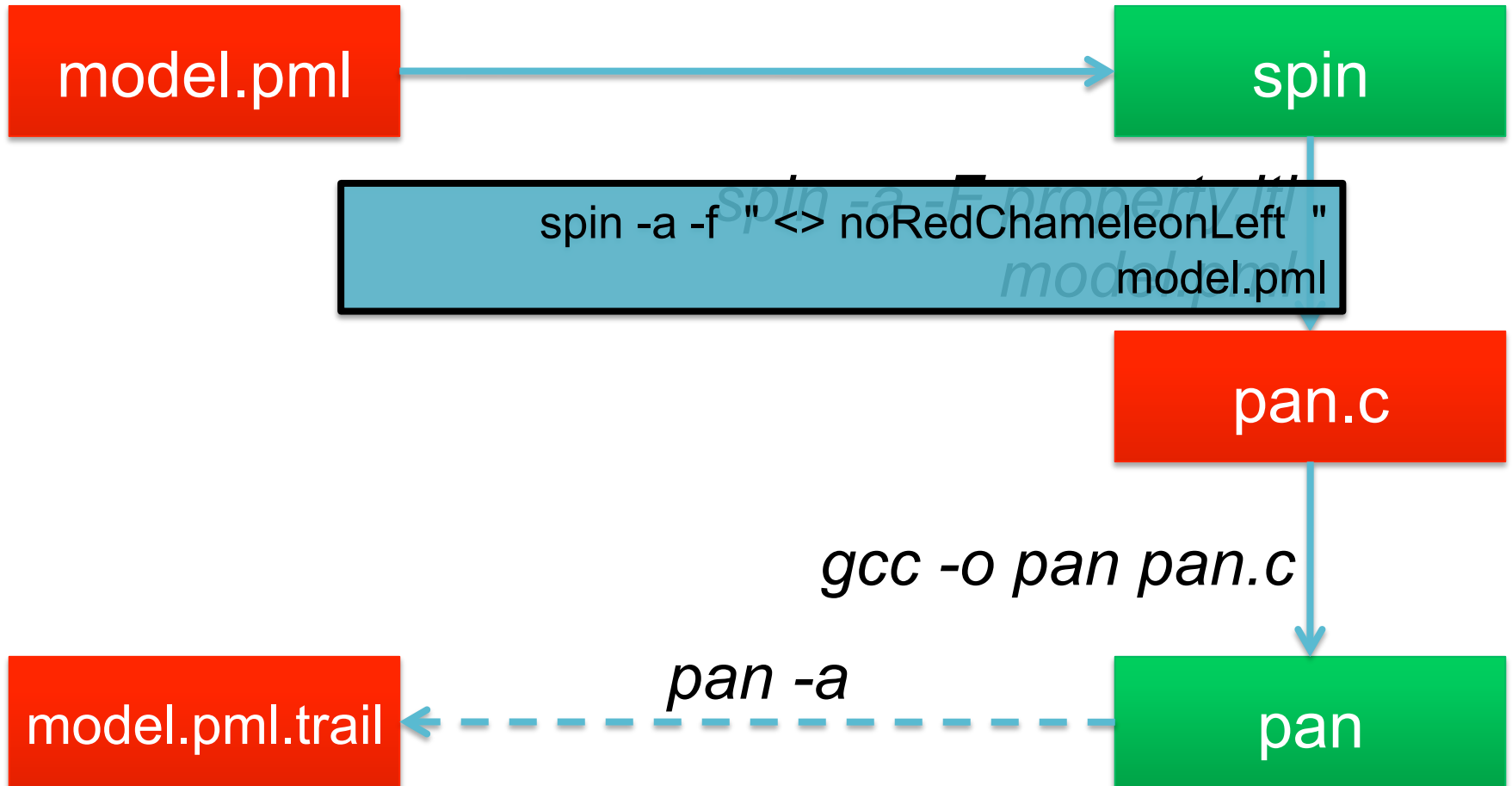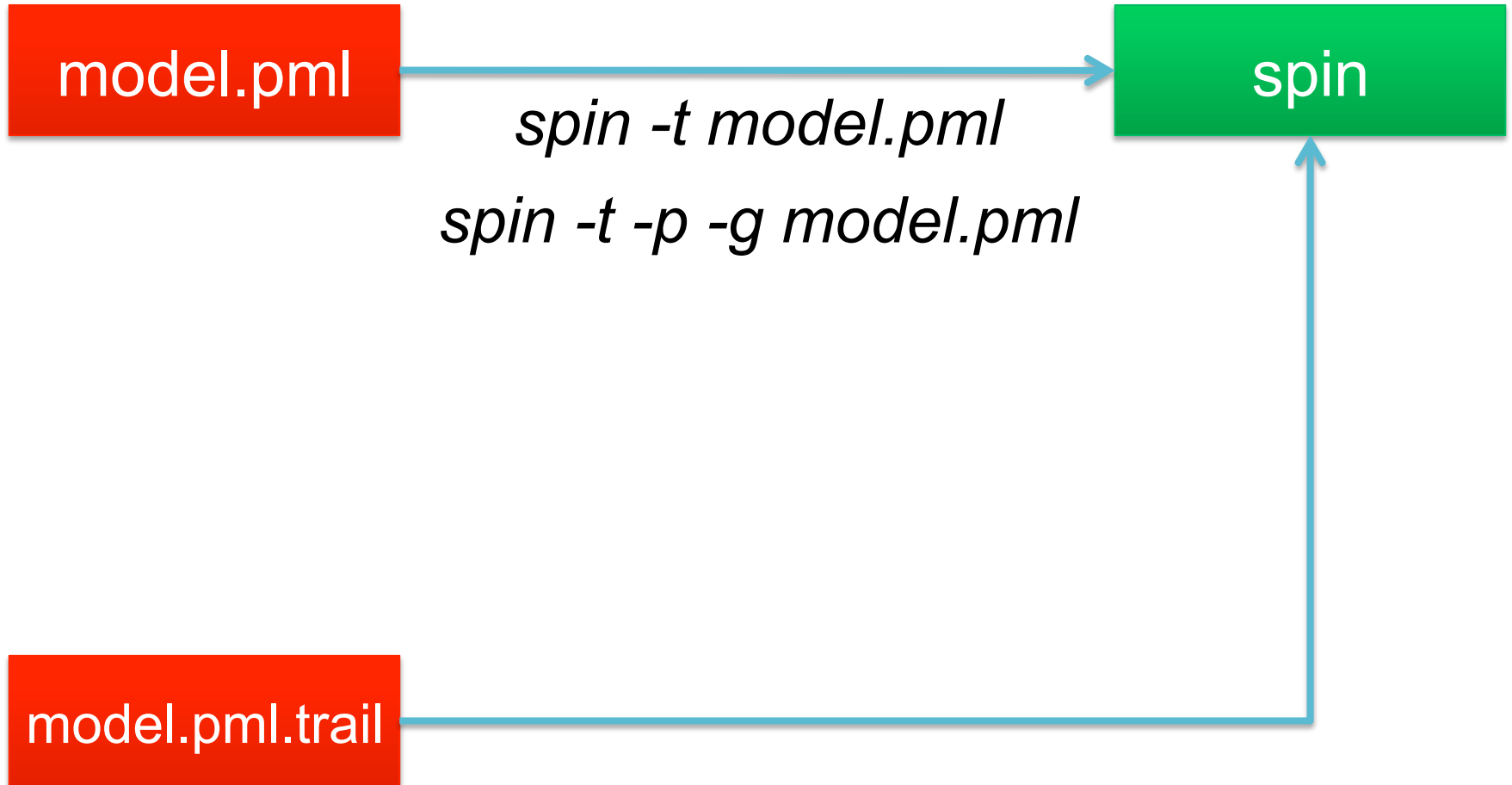
Model Checking
Verification

model.pml

property.ltl

spin

spin -a -F property.ltl model.pml

pan.c

gcc -o pan pan.c

pan -a

model.pml.trail

pan

Model Checking
Verification

model.pml → spin

spin -a -f " <> noRedChameleonLeft " model.pml

pan.c

gcc -o pan pan.c

pan -a

model.pml.trail ← pan

Model Checking
Guided Simulation

model.pml

spin -t model.pml

spin -t -p -g model.pml

spin

model.pml.trail

GOOD LUCK!