Verification using Model Checking

http://cl-informatik.uibk.ac.at/teaching/ss07/vmc/

Stefan Blom

Computational Logic
Institute of Computer Science
University of Innsbruck

SS 2007

Evaluation

- Exercises counting for $\frac{1}{3}$ of the final mark.
  - Practice exercises do not count.
  - Everybody works alone.
- Written exam counting for $\frac{2}{3}$ of the final mark.
  - First attempt in the last week: July 5, 9:00-11:00 SR12.

Topics

- Model Specification.
- Property Specification.
- Algorithms

Model Specification

- automata based
  - timed automata
  - annotated finite automata
  - basics of state charts (briefly)
- process algebra based
  - basics: actions, sequential compostion, choice
  - intermediate: parallel composition and recursive equations
  - advanced: synchronisation
  - adding in (abstract) data types

Property Specification

- Syntax and semantics of LTL, CTL, CTL*, modal μ-calculus
- Hierarchy of LTL, CTL, CTL*, modal μ-calculus
- Translation between informal descriptions and formula's
- safety, liveness, fairness

  For example, if two agents try to get exclusive access to a resource
  - Access might be granted to both agents. (safety violation)
  - Access might be granted to neither agent. (liveness violation)
  - Starvation of one of the agents. (fairness violation)

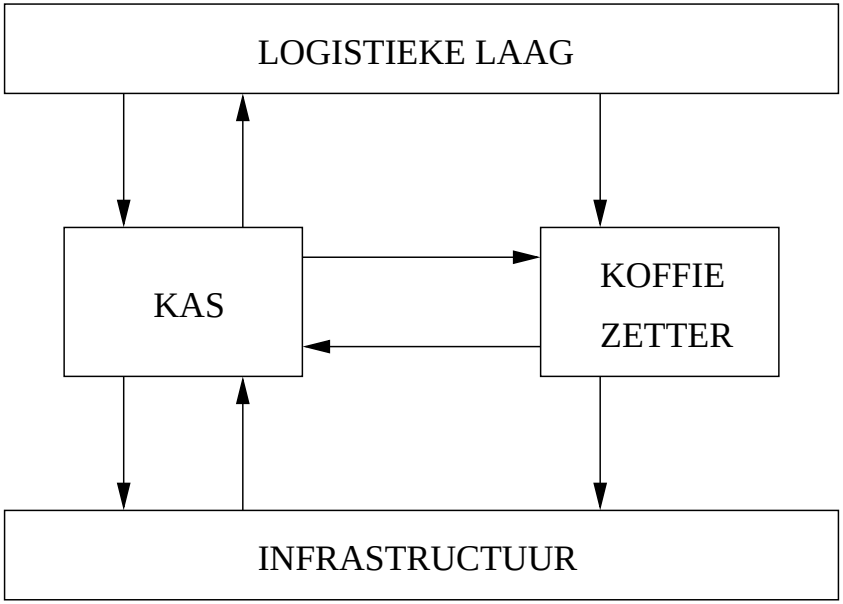
Algorithms

- automata theoretic approach to LTL
  - LTL to Buechi transformation
  - Nested depth first search
  - Transforming a Buechi Automaton for fairness
- encoding into boolean vector, BDD, SAT.

Exercises

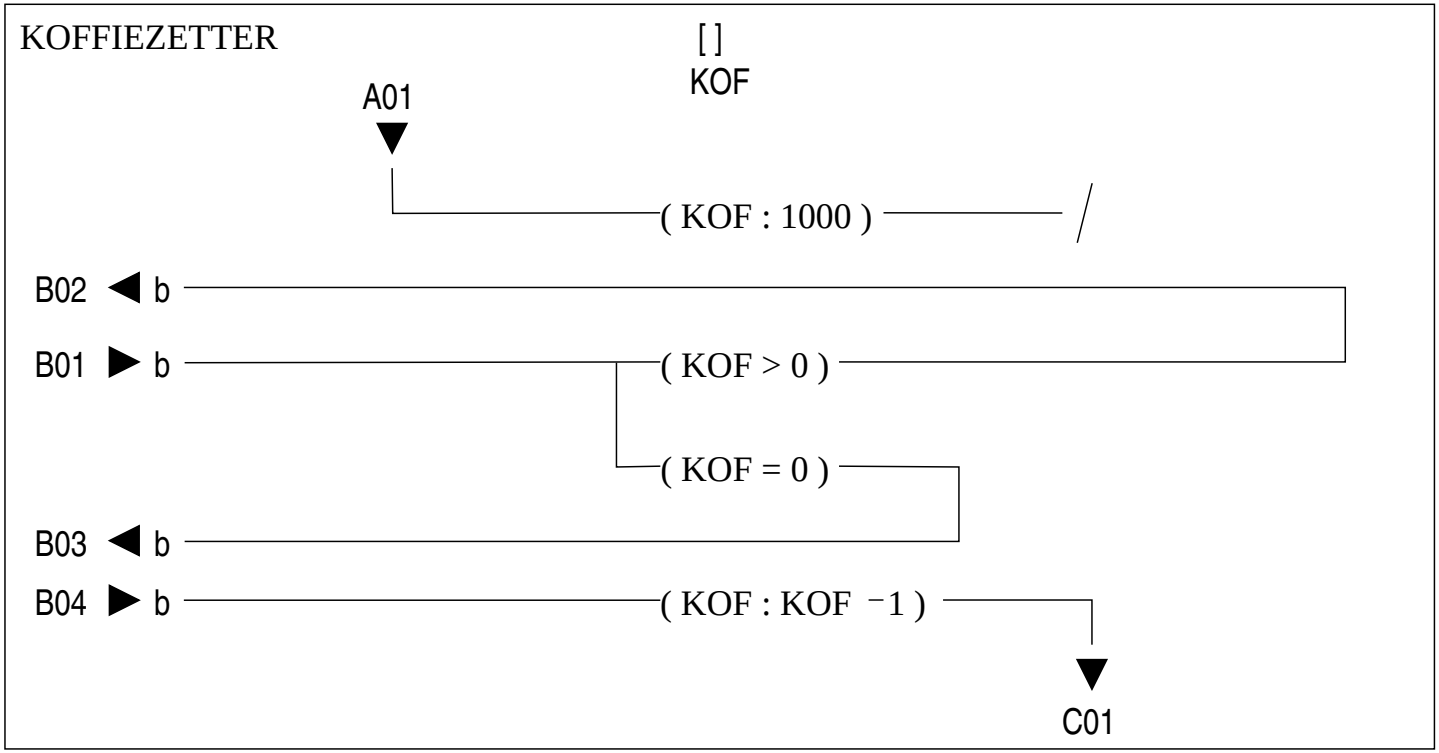
- small examples only
- problem domains
  - distributed algorithms
    (e.g. termination detection)
  - communication protocols
    (e.g. bounded retransmission protocol)
  - safety critical systems
    (e.g. traffic lights)
- timed automata in Uppaal
  - write simple automata and test against given formula's
  - write formulas to find the mistakes in given models.
- implement definitions and constructions to understand them.

A simple coffee machine

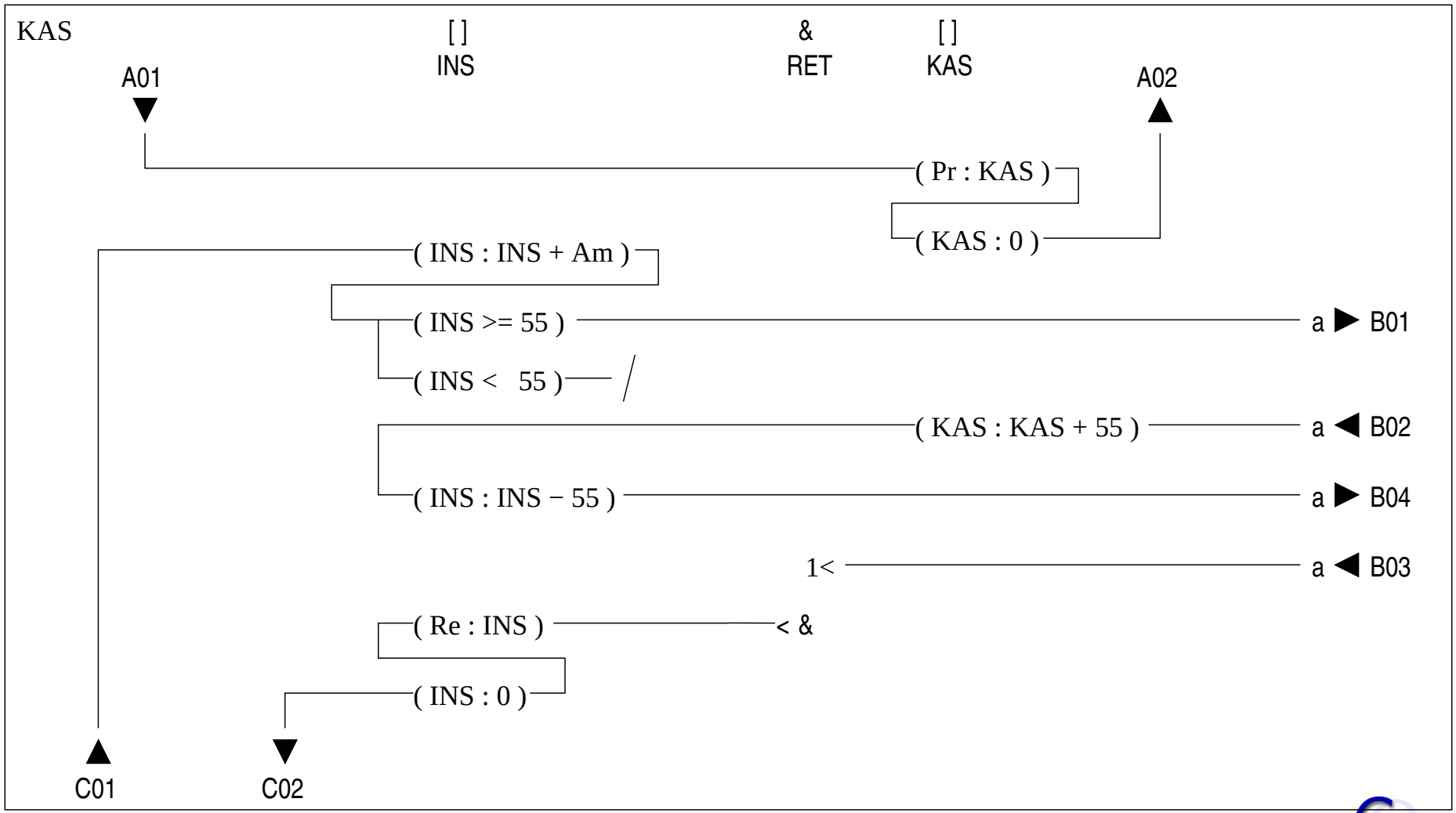


- The top box stands for the users
- The boxes in the middle are the software control components
- The bottom box stand for the physical components
- Communication is by message passing

The brewing component



The payment component



Message types

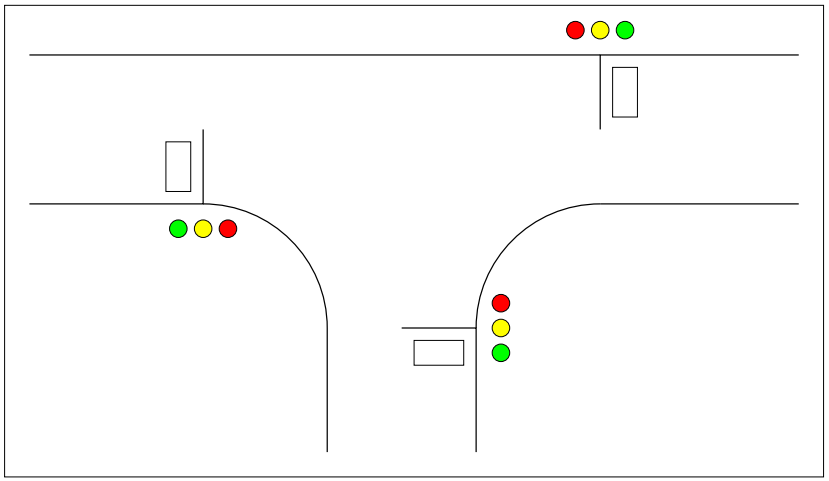
- messages up/down for brewing component
  - A01 Refill the coffee supply
  - C01 Brew and serve one cup of coffee
- messages up/down for payment component
  - A01 Ask for the proceeds.
  - A02 Send the Proceeds.
  - C01 Amount of money inserted.
  - C02 Refund money.
- messages between components
  - B01 Check if coffee can be served
  - B02 Yes, coffee can be served
  - B03 No, coffee cannot be served
  - B04 Brew and serve one cup of coffee

Variables

[] integer variable

& delayed execution flag:
These flags can be set like any boolean variable. If a flag is set at the beginning of a new time slice then the corresponding flow (starting with &) is executed.

Requirements for traffic lights



- Variables ($i \in \{1, 2, 3\}$ represent the direction):

boolean	R_i, Y_i, G_i	states of the colored lights
boolean	S_i	states of the car-waiting sensors
timer	t_i	set to 0 when a car arrives

- What are the requirements?