NSWI101: System Behaviour Models and Verification

Lab 01

Jan Kofroň

FACULTY
OF MATHEMATICS
AND PHYSICS
Charles University

Department of
Distributed and
Dependable
Systems

LTS AND EQUIVALENCES

1. Draw two LTSs being trace equivalent but different

LTS AND EQUIVALENCES

1. Draw two LTSs being trace equivalent but different
2. Draw two LTSs being simulation equivalent but different

LTS AND EQUIVALENCES

1. Draw two LTSs being trace equivalent but different
2. Draw two LTSs being simulation equivalent but different
3. Draw two LTSs being bisimilar but different

Algebra of Communicating Processes (ACP)

- Simple process algebra by Jan Bergstra and Jan Willem Klop (1982)
- Just few syntactical constructs:
  - Choice (+)
  - Sequencing (.)
  - Concurrency (||)
  - Process communication (γ)
  - Abstraction (τ)
- Example of processes:
  - $p : (gen_1 + gen_2).send$
  - $q : recv.proc$
  - Defining communication: $\gamma(send, recv) = trans$
  - Composition of processes: $p||q = (gen_1 + gen_2).trans.proc$
  - Hiding internal computation (abstraction): $\tau_{\{gen_1, gen_2, proc\}}(p||q) = \tau.trans.\tau$

ACP SEMANTICS

For process variables x, y

- $x + y = y + x$
- $(x + y) + z = x + (y + z)$
- $x + x = x$
- $(x + y).z = x.z + y.z$
- $(x.y).z = x.(y.z)$
- $x + \delta = x$
- $\delta.x = \delta$

Note that $z.(x + y) = z.x + z.y$ is not included (non-deterministic choice)!

Processes in ACP

Producer and consumer – p, c generating and transferring data:

$$PROD : (gen_1 + gen_2).send.PROD$$

$$CONS : recv.proc.CONS$$

Specify communication:

$$\gamma(send, recv) = trans$$

Compose processes:

$$COMP = \delta_{\{send,recv\}}(PROD \parallel CONS)$$

$$COMP = (gen_1 + gen_2).trans.proc.COMP$$

ALTERNATING BIT PROTOCOL (ABP) IN ACP

Sender generates data and adds one bit to the message to receiver whose value changes each time another message is sent

Tasks:

1. Model ABP in ACP
2. Think of required properties – can they be verified in your model?