

Model Checking I

alias

Reactive Systems Verification

Luca Tesei

MSc in Computer Science, University of Camerino

Topics

- Parallelism
- Interleaving operator for Transition Systems
- Examples

Material

Reading:

Chapter 2 of the book, pages 35–39.

More:

The slides in the following pages are taken from the material of the course “Introduction to Model Checking” held by Prof. Dr. Ir. Joost-Pieter Katoen at Aachen University.

Introduction

Modelling parallel systems

Transition systems

Modeling hard- and software systems

Parallelism and communication



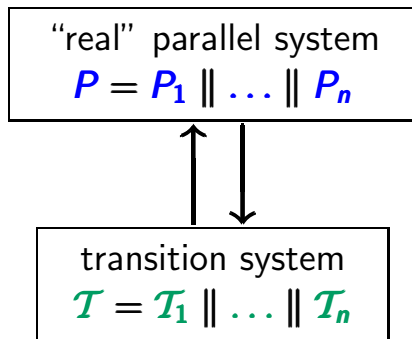
Linear Time Properties

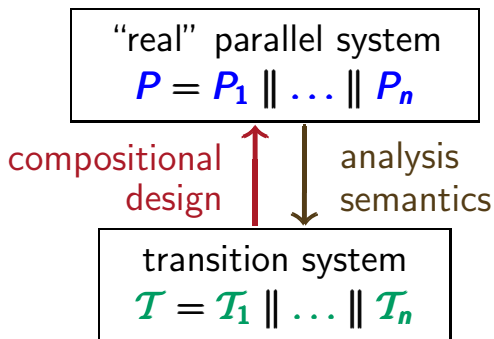
Regular Properties

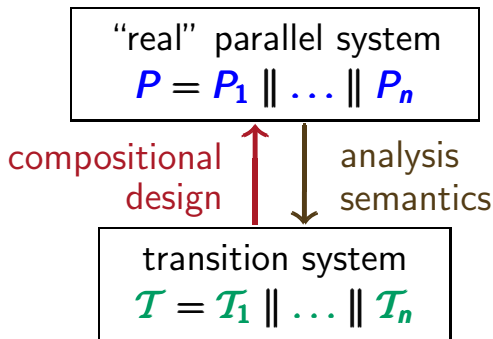
Linear Temporal Logic

Computation-Tree Logic

Equivalences and Abstraction





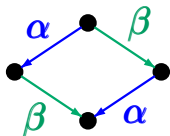
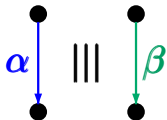


goal: define semantic parallel operators on transition systems or program graphs that model “real” parallel operators

- interleaving of **concurrent, independent** actions of parallel processes (modelled by TS)
- representation by **nondeterministic choice**:
“which subprocess performs the next step?”

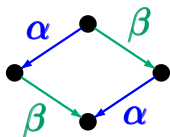
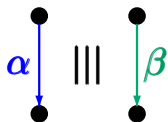
- interleaving of **concurrent, independent** actions of parallel processes (modelled by TS)
- representation by **nondeterministic choice**:
“which subprocess performs the next step?”

$$\mathit{effect}(\alpha ||| \beta) = \mathit{effect}(\alpha; \beta + \beta; \alpha)$$



- interleaving of **concurrent, independent** actions of parallel processes (modelled by TS)
- representation by **nondeterministic choice**:
“which subprocess performs the next step?”

$$\mathit{effect}(\alpha ||| \beta) = \mathit{effect}(\alpha; \beta + \beta; \alpha)$$

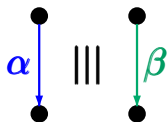


parallel execution
of α and β on
two processors

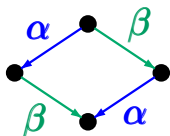
Interleaving operator $|||$ for TS

- interleaving of **concurrent, independent** actions of parallel processes (modelled by TS)
- representation by **nondeterministic choice**:
“which subprocess performs the next step?”

$$\mathit{effect}(\alpha ||| \beta) = \mathit{effect}(\alpha; \beta + \beta; \alpha)$$



parallel execution
of α and β on
two processors



serial execution on
a *single processor*
in arbitrary order

\cong

Interleaving operator \parallel for TS

$$\mathcal{T}_1 = (S_1, Act_1, \longrightarrow_1, S_{0,1}, AP_1, L_1)$$

$$\mathcal{T}_2 = (S_2, Act_2, \longrightarrow_2, S_{0,2}, AP_2, L_2)$$

$$\mathcal{T}_1 = (S_1, Act_1, \longrightarrow_1, S_{0,1}, AP_1, L_1)$$

$$\mathcal{T}_2 = (S_2, Act_2, \longrightarrow_2, S_{0,2}, AP_2, L_2)$$

The transition system $\mathcal{T}_1 \parallel \mathcal{T}_2$ is defined by:

$$\mathcal{T}_1 \parallel \mathcal{T}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \longrightarrow, S_{0,1} \times S_{0,2}, AP, L)$$

where the transition relation \longrightarrow is given by:

$$\mathcal{T}_1 = (S_1, Act_1, \longrightarrow_1, S_{0,1}, AP_1, L_1)$$

$$\mathcal{T}_2 = (S_2, Act_2, \longrightarrow_2, S_{0,2}, AP_2, L_2)$$

The transition system $\mathcal{T}_1 \parallel\parallel \mathcal{T}_2$ is defined by:

$$\mathcal{T}_1 \parallel\parallel \mathcal{T}_2 = (S_1 \times S_2, Act_1 \cup Act_2, \longrightarrow, S_{0,1} \times S_{0,2}, AP, L)$$

where the transition relation \longrightarrow is given by:

$$\frac{s_1 \xrightarrow{\alpha}_1 s'_1}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s'_1, s_2 \rangle} \quad \frac{s_2 \xrightarrow{\alpha}_2 s'_2}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s'_2 \rangle}$$

Interleaving operator \parallel for TS

$$\mathcal{T}_1 = (\mathcal{S}_1, \text{Act}_1, \longrightarrow_1, \mathcal{S}_{0,1}, \text{AP}_1, L_1)$$

$$\mathcal{T}_2 = (\mathcal{S}_2, \text{Act}_2, \longrightarrow_2, \mathcal{S}_{0,2}, \text{AP}_2, L_2)$$

The transition system $\mathcal{T}_1 \parallel \mathcal{T}_2$ is defined by:

$$\mathcal{T}_1 \parallel \mathcal{T}_2 = (\mathcal{S}_1 \times \mathcal{S}_2, \text{Act}_1 \cup \text{Act}_2, \longrightarrow, \mathcal{S}_{0,1} \times \mathcal{S}_{0,2}, \text{AP}, L)$$

where the transition relation \longrightarrow is given by:

$$\frac{s_1 \xrightarrow{\alpha}_1 s'_1}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s'_1, s_2 \rangle} \quad \frac{s_2 \xrightarrow{\alpha}_2 s'_2}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s'_2 \rangle}$$

atomic propositions: $\text{AP} = \text{AP}_1 \uplus \text{AP}_2$

labeling function: $L(\langle s_1, s_2 \rangle) = L_1(s_1) \cup L_2(s_2)$

just a simple notation for operational semantics

$$\frac{\text{premise}}{\text{conclusion}}$$

just a simple notation for **operational semantics**

$$\frac{\text{premise}}{\text{conclusion}}$$

E.g., “the relation \longrightarrow is given by ...”

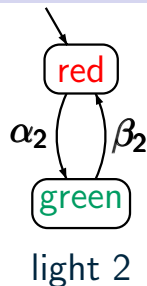
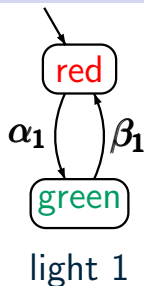
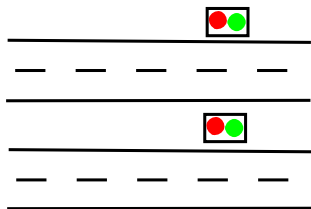
$$\frac{s_1 \xrightarrow{\alpha}_1 s'_1}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s'_1, s_2 \rangle} \qquad \frac{s_2 \xrightarrow{\alpha}_2 s'_2}{\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s'_2 \rangle}$$

means that \longrightarrow is the **smallest relation** such that:

- (1) If $s_1 \xrightarrow{\alpha}_1 s'_1$, then $\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s'_1, s_2 \rangle$
- (2) If $s_2 \xrightarrow{\alpha}_2 s'_2$, then $\langle s_1, s_2 \rangle \xrightarrow{\alpha} \langle s_1, s'_2 \rangle$

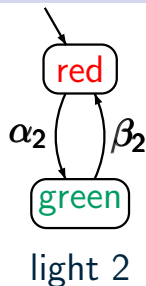
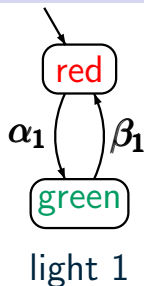
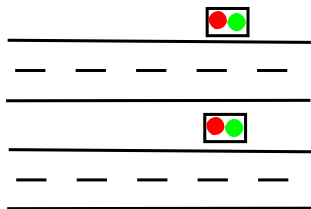
Useless lights for non-crossing streets

PC2.2-4

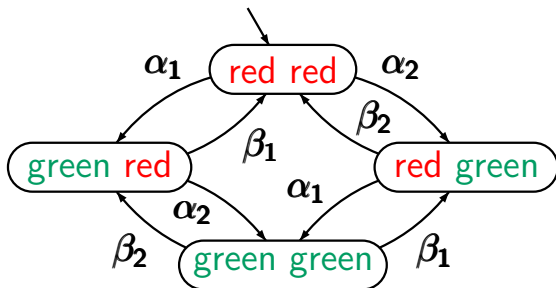


Useless lights for non-crossing streets

PC2.2-4



light 1 ||| light 2

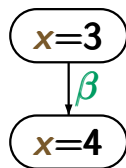
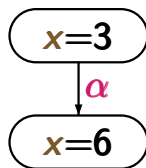


Dependent actions

PC2.2-5

dependent actions $\alpha \hat{=} x := 2x$ and $\beta \hat{=} x := x + 1$

representations in
transition systems

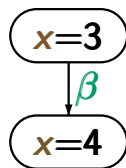
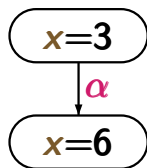


Dependent actions

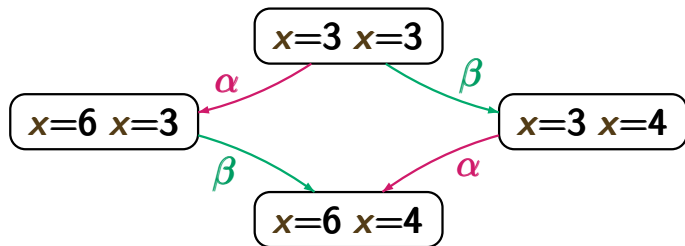
PC2.2-5

dependent actions $\alpha \hat{=} x := 2x$ and $\beta \hat{=} x := x + 1$

representations in
transition systems



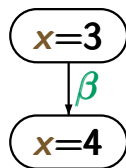
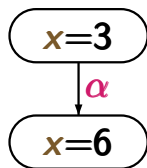
interleaving operator \parallel



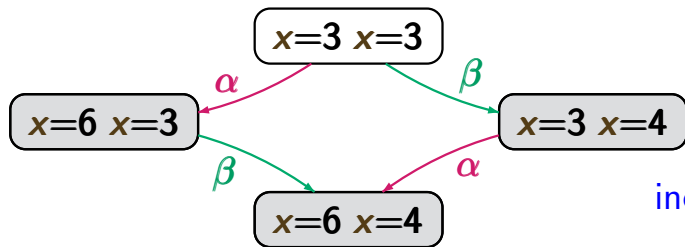
Interleaving fails for dependent actions

dependent actions $\alpha \hat{=} x := 2x$ and $\beta \hat{=} x := x + 1$

representations in transition systems



interleaving operator \parallel for transition systems “fails”



inconsistent
states