Software architecture for reactive systems: A short look at model checking (revisions)

José Proença

HASLab - INESC TEC
Universidade do Minho
Braga, Portugal

February 2017

Model checking

Recall "Especificação e Modelação":

- Modelling reactive systems – Kripke structures and ~~Petri Nets~~ SMV
- Specification – Temporal logics (LTL ~~and CTL/CTL*~~)
- Verification – Check if a formula holds in a system

SMV model checker

What we will see

- Labelled transition systems (LTS) as Kripke structures
  - Process algebra (not ~~Petri Nets~~ SMV) to define LTS
  - mCRL2 toolset to model (not SMV)
  - Equivalence of LTS
- Modal logics – generalising temporal logics (CTL*,CTL,LTL)
- Using mCRL2 toolset to verify properties

- Later: Timed-automata and UPPAAL model checker (CTL)

Model

$\mathfrak{M}, w \models \phi$ – **what does it mean?**

Model definition

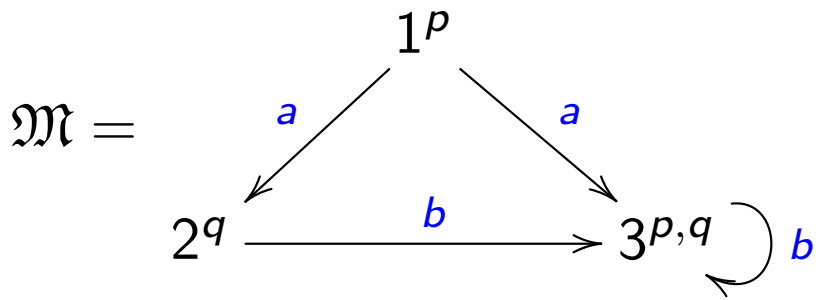
A model for the language is a pair $\mathfrak{M} = \langle \mathfrak{F}, V \rangle$, where

- $\mathfrak{F} = \langle W, \{R_m\}_{m \in \mathrm{MOD}} \rangle$ is a Kripke frame, ie, a non empty set W and a family R_m of binary relations (called *accessibility relations*) over W, one for each modality symbol $m \in \mathrm{MOD}$. Elements of W are called points, states, worlds or simply vertices in directed graphs.
- $V : \mathrm{PROP} \longrightarrow \mathcal{P}(W)$ is a valuation.

Kripke structures from last semester

- $\mathrm{MOD} = \{\mathbf{1}\}$
- (S, I, R, L) where $S = W$, $I = \{w\}$, $R = R_{\mathbf{1}}$, $L = V$
- $\mathfrak{F} = \langle W, R \rangle$ instead of $\mathfrak{F} = \langle W, \{R_m\}_{m \in \mathrm{MOD}} \rangle$

Example

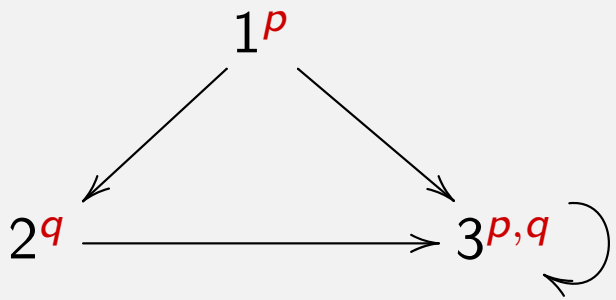


$$
\begin{aligned}
W &= \{1, 2, 3\} \\
MOD &= \{a, b\} \\
R_a &= \{(1, 2), (1, 3)\} \\
R_b &= \{(2, 3), (3, 3)\} \\
V &= \{1 \mapsto \{p\}, \\
&\quad 2 \mapsto \{q\}, \\
&\quad 3 \mapsto \{p, q\}\}
\end{aligned}
$$

- $\mathfrak{M}, 1 \models p$
  means p holds in state 1
- $\mathfrak{M}, 2 \models [b]\, p$
  means p holds in every state reachable with b from 2.

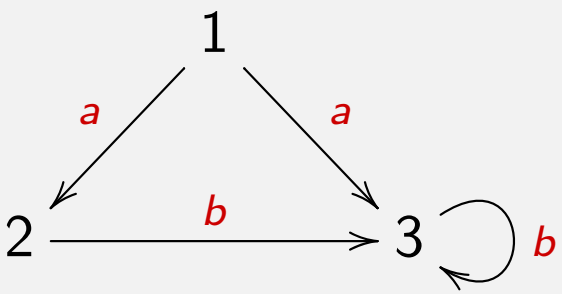
Key differences

Before



- emphasize on states - desired/forbidden states
- SMV language to generate models
- $\mathfrak{M}, 1 \models p$, $\mathfrak{M}, 1 \models F\,G\,p$

Now



- emphasize on actions - desired/forbidden sequences
- Process algebra to generate models
- $\mathfrak{M}, 2 \models [a]\,\text{false}$