

The Timer Cascade: Functional Modelling and Real Time Calculi

Raymond Boute Andreas Schäfer

INTEC, Universiteit Gent, Belgium

Department für Informatik, Universität Oldenburg, Germany

ICTAC 05



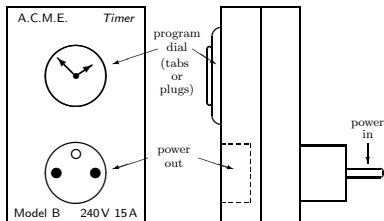
- 1 Motivation
- 2 The Timer Cascade
- 3 Functional Modelling
- 4 Real-Time Calculi



- Various formalisms for modelling and reasoning about real-time and hybrid systems.
- Originating from ICTAC04 in Guiyang: Timer cascade as a common case study
 - for comparing and understanding the formalisms
 - interesting and challenging but scaling down.

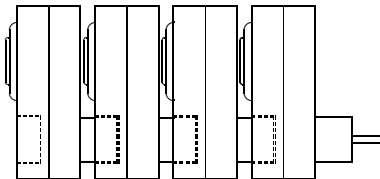


Common household device with no internal battery:





Common household device with no internal battery:
Cascaded clock is only running if timer is powered.





Functional Mathematics (FunMath)

- functional predicate calculus
- concrete generic functionals

Here: Application of paper and pencil modelling and reasoning



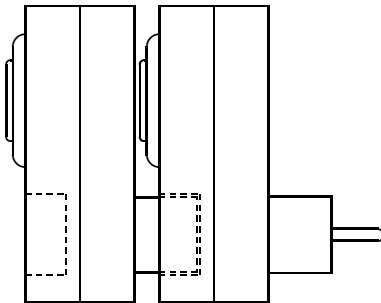
Assumptions: Infinite Programs

Signal space: $\text{Sig} := \{P : \mathbb{R}_{\geq 0} \rightarrow \mathbb{B} \mid P \text{ is p.c.}\}$ Programs are of type Sig .

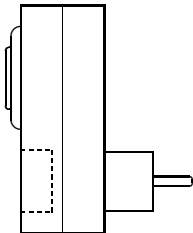
Behavior: $\text{Bvr} := \text{Sig} \rightarrow \text{Sig}$

def $\text{Tmr} : \text{Sig} \rightarrow \text{Bvr}$ **with** $\text{Tmr } P \mid t \equiv \mid t \wedge P(\int \mid t)$ Blub

def $\text{Csc} : \text{Sig}^* \rightarrow \text{Bvr}$ **with** $\text{Csc } p = \bigcirc j : \mathcal{D} p. \text{Tmr}(p(\#p - 1 - j))$



Given two timers with programs P and P' .



Is there a program $P \triangleright P'$ for one timer exhibiting the same behavior?



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \text{Tmr}(\text{Tmr} P P')(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge P'(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle P(\int (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \text{Tmr} P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle (\text{Tmr} P \circ \text{Tmr} P') I t \end{aligned}$$



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \text{Tmr}(\text{Tmr} P P')(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge P'(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle P(\int (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \text{Tmr} P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle (\text{Tmr} P \circ \text{Tmr} P') I t \end{aligned}$$



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \text{Tmr}(\text{Tmr} P P')(f I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(f P'(f I t)) \wedge P'(f I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(f P'(f I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle P(f (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \text{Tmr} P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle (\text{Tmr} P \circ \text{Tmr} P') I t \end{aligned}$$



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \text{Tmr}(\text{Tmr} P P')(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge P'(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(\int P'(\int I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle P(\int (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \text{Tmr}P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle (\text{Tmr}P \circ \text{Tmr}P') I t \end{aligned}$$



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \text{Tmr}(\text{Tmr} P P')(f I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(f P'(f I t)) \wedge P'(f I t) \wedge I t \\ &= \langle \text{Def.} \rangle P(f P'(f I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle P(f (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \text{Tmr} P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle (\text{Tmr} P \circ \text{Tmr} P') I t \end{aligned}$$



Requirement: One program $P \triangleright P'$ with behavior of the composition of two timer with programs P and P' , i.e. $\text{Tmr}(P \triangleright P') = \text{Tmr}P \circ \text{Tmr}P'$

$$\begin{aligned} \text{Tmr}(\text{Tmr} P P') I t &= \langle \text{Def.} \rangle \quad \text{Tmr}(\text{Tmr} P P')(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle \quad P(\int P'(\int I t)) \wedge P'(\int I t) \wedge I t \\ &= \langle \text{Def.} \rangle \quad P(\int P'(\int I t)) \wedge \text{Tmr} P' I t \\ &= \langle \text{Lemma} \rangle \quad P(\int (\text{Tmr} P' I) t) \wedge \text{Tmr} P' I t \\ &= \langle \text{Def.} \rangle \quad \text{Tmr}P(\text{Tmr} P' I) t \\ &= \langle \text{Def.} \rangle \quad (\text{Tmr}P \circ \text{Tmr}P') I t \end{aligned}$$



Sig with

- \triangleright defined by $P \triangleright P' = \text{Tmr } P P'$
- function 1

is a monoid and Tmr an (injective) monoid homomorphism.



Apply model-checking for Duration Calculus

- is an interval temporal logic for real-time systems
- equipped with powerful proof system (yet not axiomatisable for continuous time domain)
- decidable for discrete time domain.
- incorporates the \int operator to measure the duration of states
- and the chop-operator \frown to split intervals.

Here: Application of automatic model-checking



More concrete modelling than in FunMath, assuming cyclic behavior.
Observables (Variables of type sf)

- power_in_i : Timer i is powered.
- power_out_i : Outlet of timer i is powered.
- pass_i : Current can pass through timer i .



Modelling behavior by formulas:

- $(\int \text{power_in}_i \text{ mod } \text{cycle}_i < \text{start}_i) \Rightarrow \text{true} \wedge [\neg \text{pass}_i]$
If the duration of power_in_i is below the start value, pass has to be false.
- $((\text{start}_i \leq \int \text{power_in}_i \text{ mod } \text{cycle}_i \leq \text{stop}_i) \Rightarrow \text{true} \wedge [\text{pass}_i])$
- $((\text{start}_i \leq \int \text{power_in}_i \text{ mod } \text{cycle}_i \leq \text{stop}_i) \Rightarrow \text{true} \wedge [\text{pass}_i])$
- $((\int \text{power_in}_i \text{ mod } \text{cycle}_i > \text{stop}_i) \Rightarrow \text{true} \wedge [\neg \text{pass}_i])$
- $\square \vee [(\text{power_in}_i \wedge \text{pass}) \Leftrightarrow \text{power_out}_i]$
- $\square \vee [\text{power_in}_{i+1} \Leftrightarrow \text{power_out}_i]$
- $\square \vee [\text{power_in}_0]$



Given two conjunctions of formulas

- TC^C describing a concrete cascade and
- TC^A describing an abstract cascade.

Refinement requirement

$$TC^C \wedge TC^A \Rightarrow [\text{power_out}_1^C \Leftrightarrow \text{power_out}^A].$$



Using model-checking technique (DCValid) requires the following restrictions

- Discrete Time (not severe)
- Concrete values for programs (full arithmetic not available)
- An auxiliary observable indicating zero positions to handle cyclic behavior

Benefits:

-



- Embedding of DC in FunMath is possible:
 - Predicates have type $IP := \mathcal{I} \rightarrow \mathbb{B}$
 - Chop \wedge has type $IP^2 \rightarrow IP$ by $(P \wedge Q)I \equiv \exists t : T.P I_{\leq t} \wedge Q I_{\geq t}$



- Completing the elaboration of functional and DC model.
- Broad comparative survey of formalisms for real-time and hybrid systems.
 - elaborating an example highlighting its strong points presenting and
 - using the timer cascade a running example
 - investigating linking of formalisms and tool interaction.