Real-Time Systems

Lecture 06: DC Properties I

2014-05-22

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

– 06 – 2014-05-22 – main –

Contents & Goals

Last Lecture:

- DC Syntax and Semantics: Abbreviations ("almost everywhere")
- Satisfiable/Realisable/Valid (from 0)

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
  - What are obstacles on proving a design correct in the real-world, and how to overcome them?
  - Facts: decidability properties.
  - What's the idea of the considered (un)decidability proofs?
- **Content:**
  - Semantical Correctness Proof
  - (Un-)Decidable problems of DC variants in discrete and continuous time

Specification and Semantics-based Correctness Proofs of Real-Time Systems with DC

Methodology: Ideal World...

(i) Choose a collection of **observables** 'Obs'.

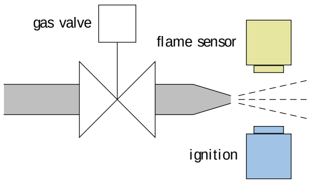
(ii) Provide the **requirement/specification** 'Spec' as a conjunction of DC formulae (over 'Obs').

(iii) Provide a description 'Ctrl' of the **controller** in form of a DC formula (over 'Obs').

(iv) We say 'Ctrl' is **correct** (wrt. 'Spec') iff

$$\models_0 \mathsf{Ctrl} \implies \mathsf{Spec}.$$

Gas Burner Revisited



(i) Choose **observables**:

- two boolean observables G and F
  (i.e. Obs $= \{G, F\}$, $\mathcal{D}(G) = \mathcal{D}(F) = \{0, 1\}$)
- $G = 1$: gas valve open (output)
- $F = 1$: have flame (input)
- define $L := G \wedge \neg F$ (leakage)

(ii) Provide the **requirement**:

$$\text{Req} :\iff \Box(\ell \geq 60 \implies 20 \cdot \textstyle\int L \leq \ell)$$

Gas Burner Revisited

(iii) Provide a description ‘Ctrl’ of the **controller** in form of a DC formula (over ‘Obs’). Here, firstly consider a **design**:

- Des-1 : $\iff \Box(\lceil L \rceil \implies \ell \leq 1)$
- Des-2 : $\iff \Box(\lceil L \rceil \,;\, \lceil \neg L \rceil \,;\, \lceil L \rceil \implies \ell > 30)$

(iv) Prove **correctness**:

- We want (or do we want $\models_0$ ...?):

$$\models (\text{Des-1} \wedge \text{Des-2} \implies \text{Req}) \qquad \text{(Thm. 2.16)}$$

- We do show

$$\models \text{Req-1} \implies \text{Req} \qquad \text{(Lem. 2.17)}$$

  with the simplified requirement

$$\text{Req-1} := \Box(\ell \leq 30 \implies \textstyle\int L \leq 1),$$

- and we show

$$\models (\text{Des-1} \wedge \text{Des-2}) \implies \text{Req-1}. \qquad \text{(Lem. 2.19)}$$

Gas Burner Revisited: Lemma 2.17

Claim:

$$\models \underbrace{\Box(\ell \le 30 \implies \textstyle\int L \le 1)}_{\text{Req-1}} \implies \underbrace{\Box(\ell \ge 60 \implies 20 \cdot \textstyle\int L \le \ell)}_{\text{Req}}$$

Proof:

- Assume 'Req-1'.
- Let $L_\mathcal{I}$ be any interpretation of L, s.t. $\mathcal{I} \models$ Req-1 and $[b, e]$ an interval with $e - b \ge 60$.
- Show "$20 \cdot \int L \le \ell$", i.e.

$$\mathcal{I}[\![20 \cdot \textstyle\int L \le \ell]\!](\mathcal{V}, [b,e]) = tt$$

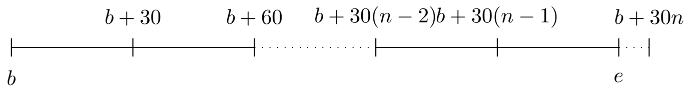
i.e.

$$20 \cdot \int_b^e L_\mathcal{I}(t)\, dt \le (e-b)$$

Gas Burner Revisited: Lemma 2.17

$$\models \underbrace{\Box(\ell \le 30 \implies \textstyle\int L \le 1)}_{\text{Req-1}}$$
$$\implies \Box(\ell \ge 60 \implies 20 \cdot \textstyle\int L \le \ell)$$

- Set $n := \lceil \frac{e-b}{30} \rceil$, i.e. $n \in \mathbb{N}$ with $n - 1 < \overset{(*)}{\frac{e-b}{30}} \le n$, and split the interval



$$20 \cdot \int_b^e L_\mathcal{I}(t)\, dt$$

$$= 20 \cdot \left(\sum_{i=0}^{n-2} \underbrace{\int_{b+30 \cdot i}^{b+(i+1) \cdot 30} L_\mathcal{I}(t)\, dt}_{\text{"}\int L\text{"}} + \int_{b+30 \cdot (n-1)}^{e} L_\mathcal{I}(t)\, dt \right)$$

$\{\text{Req-1}\}$ $\le 20 \cdot \sum_{i=0}^{n-2} 1 + 20 \cdot 1$

$= 20 \cdot n$

$\{(*)\}$ $< 20 \cdot \left(\frac{e-b}{30} + 1\right)$

$= \frac{2}{3}(e-b) + 20$

$\{e-b \ge 60 \iff 20 \le \frac{1}{3}(e-b)\}$ $\le e-b$

Some Laws of the DC Integral Operator

Theorem 2.18.
For all state assertions P and all real numbers $r_1, r_2 \in \mathbb{R}$,

(i) $\models \int P \leq \ell$,

(ii) $\models (\int P = r_1) \,;\, (\int P = r_2) \implies \int P = r_1 + r_2$,

(iii) $\models \lceil \neg P \rceil \implies \int P = 0$,

(iv) $\models \lceil\rceil \implies \int P = 0$.

Gas Burner Revisited: Lemma 2.18

(i) $\int P \leq \ell$ (iii) $\lceil \neg P \rceil \Rightarrow \int P = 0$

(ii) $(\int P = r_1);(\int P = r_2) \Rightarrow \int P = r_1 + r_2$

(iv) $\lceil\rceil \Rightarrow \int P = 0$

Claim: for all $\mathcal{I}, \mathcal{V}, [b,e]$

$$\models (\underbrace{\Box(\lceil L \rceil \implies \ell \leq 1)}_{\text{Des-1}} \wedge \underbrace{\Box(\lceil L \rceil \,;\, \lceil \neg L \rceil \,;\, \lceil L \rceil \implies \ell > 30))}_{\text{Des-2}} \implies \underbrace{\Box(\ell \leq 30 \implies \int L \leq 1)}_{\text{Req-1}}$$

Proof:

$\ell \leq 30$

$\Rightarrow \lceil\rceil$
$\vee \lceil L \rceil ; (\lceil\rceil \vee \lceil \neg L \rceil)$
$\vee \lceil \neg L \rceil ; (\lceil\rceil \vee \lceil L \rceil)$
$\vee \lceil \neg L \rceil ; \lceil L \rceil ; \lceil \neg L \rceil$ (*)

$\ell \leq 30 \wedge \Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil)$ ↯ Des-2

{Des-2} $\Rightarrow$ (*)

{Des-1} $\Rightarrow \lceil\rceil$
$\vee (\ell \leq 1) ; (\lceil\rceil \vee \lceil \neg L \rceil)$
$\vee \lceil \neg L \rceil ; (\lceil\rceil \vee (\ell \leq 1))$
$\vee \lceil \neg L \rceil ; (\ell \leq 1) ; \lceil \neg L \rceil$

{(i)} $\Rightarrow \lceil\rceil$
$\vee (\int L \leq 1) ; (\lceil\rceil \vee \lceil \neg L \rceil)$
$\vee \lceil \neg L \rceil ; (\lceil\rceil \vee (\int L \leq 1))$
$\vee \lceil \neg L \rceil ; (\int L \leq 1) ; \lceil \neg L \rceil$

{(iv), (iii)} $\Rightarrow \int L = 0$
$\vee (\int L \leq 1) ; (\int L = 0 \vee \int L = 0)$
$\vee (\int L = 0) ; (\int L = 0 \vee \int L \leq 1)$
$\vee (\int L = 0) ; (\int L \leq 1) ; (\int L = 0)$

{(ii)} $\Rightarrow \int L = 0$
$\vee \int L \leq 1 + 0$
$\vee \int L \leq 0 + 1$
$\vee \int L \leq 0 + 1 + 0$

$\Rightarrow \int L \leq 1$ $\square$

Obstacles in Non-Ideal World

Methodology: The World is Not Ideal...

(i) Choose a collection of **observables** 'Obs'.

(ii) Provide **specification** 'Spec' (conjunction of DC formulae (over 'Obs')).

(iii) Provide a description 'Ctrl' of the **controller** (DC formula (over 'Obs')).

(iv) Prove 'Ctrl' is **correct** (wrt. 'Spec').

That looks **too simple to be practical**. Typical **obstacles**:

(i) It may be impossible to realise 'Spec'
if it doesn't consider properties of **the plant**.

(ii) There are typically intermediate **design levels** between 'Spec' and 'Ctrl'.

(iii) 'Spec' and 'Ctrl' may use **different observables**.

(iv) **Proving** validity of the implication is not trivial.

(i) Assumptions As A Form of Plant Model

- Often the controller will (or can) operate correctly only under some **assumptions**.
- For instance, with a level crossing
  - we may assume an upper bound on the speed of approaching trains, (otherwise we'd need to close the gates arbitrarily fast)
  - we may assume that trains are not arbitrarily slow in the crossing, (otherwise we can't make promises to the road traffic)
- We shall specify such assumptions as a DC formula 'Asm' on the **input observables** and verify correctness of 'Ctrl' wrt. 'Spec' by proving validity (from 0) of

$$\text{Ctrl} \wedge \text{Asm} \implies \text{Spec}$$

- Shall we **care** whether 'Asm' is satisfiable? YES!

(ii) Intermediate Design Levels

- A top-down development approach may involve
  - Spec — specification/requirements
  - Des — design
  - Ctrl — implementation
- Then correctness is established by proving validity of

$$\text{Ctrl} \implies \text{Des} \tag{1}$$

and

$$\text{Des} \implies \text{Spec} \tag{2}$$

(then concluding Ctrl $\implies$ Spec by transitivity)

- Any preference on the order?

(iii): Different Observables

- Assume, 'Spec' uses more abstract observables Obs_A and 'Ctrl' more concrete ones Obs_C.
- For instance:
  - in Obs_A: only consider gas valve open or closed ($\mathcal{D}(G) = \{0, 1\}$)
  - in Obs_C: may control two valves and care for intermediate positions, for instance, to react to different heating requests ($\mathcal{D}(G_1) = \{0, 1, 2, 3\}, \mathcal{D}(G_2) = \{0, 1, 2, 3\}$)
- To prove correctness, we need information how the observables are related — an **invariant** which **links** the data values of Obs_A and Obs_C.
- **If** we're given the linking invariant as a DC formula, say '$\mathsf{Link}_{C,A}$', **then** proving correctness of 'Ctrl' wrt. 'Spec' amounts to proving validity (from 0) of

$$\mathsf{Ctrl} \wedge \mathsf{Link}_{C,A} \implies \mathsf{Spec}.$$

- For instance,

$$\mathsf{Link}_{C,A} = \lceil G \iff (G_1 + G_2 > 0) \rceil$$

Obstacle (iv): How to Prove Correctness?

- by hand on the basis of DC semantics,
- maybe supported by proof rules,
- sometimes a general theorem may fit (e.g. cycle times of PLC automata),
- algorithms as in Uppaal.

DC Properties

Decidability Results: Motivation

- Recall:

  Given **assumptions** as a DC formula 'Asm' on the input observables, verifying **correctness** of 'Ctrl' wrt. 'Spec' amounts to proving

$$\models_0 \mathsf{Ctrl} \wedge \mathsf{Asm} \implies \mathsf{Spec} \tag{1}$$

- If 'Asm' is **not satisfiable** then (1) is trivially valid, and thus each 'Ctrl' correct wrt. 'Spec'.
- So: strong interest in assessing the **satisfiability** of DC formulae.

- Question: is there an automatic procedure to help us out? (a.k.a.: is it **decidable** whether a given DC formula is satisfiable?)

- More interesting for 'Spec': is it **realisable** (from 0)?
- Question: is it **decidable** whether a given DC formula is realisable?

Decidability Results for Realisability: Overview

Fragment	Discrete Time	Continous Time
RDC	decidable	decidable
RDC $+ \ \ell = r$	decidable for $r \in \mathbb{N}$	undecidable for $r \in \mathbb{R}^+$
RDC $+ \int P_1 = \int P_2$	undecidable	undecidable
RDC $+ \ \ell = x, \forall x$	undecidable	undecidable
DC	—"—	—"—

References

[Olderog and Dierks, 2008] Olderog, E.-R. and Dierks, H. (2008). Real-Time Systems - Formal Specification and Automatic Verification. Cambridge University Press.