

Real-Time Systems

Lecture 05: Duration Calculus III

2013-05-07

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

Contents & Goals

Last Lecture:

- DC Syntax and Semantics: Terms, Formulae

This Lecture:

- **Educational Objectives:** Capabilities for following tasks/questions.
 - Read (and at best also write) Duration Calculus formulae – including abbreviations.
 - What is Validity/Satisfiability/Realisability for DC formulae?
 - How can we prove a design correct?
- **Content:**
 - Duration Calculus Abbreviations
 - Basic Properties
 - Validity, Satisfiability, Realisability
 - *A correctness proof for a gas burner design*

Duration Calculus Cont'd

Duration Calculus: Overview

We will introduce three (or five) syntactical “levels”:

(i) **Symbols:**

$f, g, \text{ true, false, } =, <, >, \leq, \geq, x, y, z, X, Y, Z, d$

(ii) **State Assertions:**

$P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \wedge P_2$

(iii) **Terms:**

$\theta ::= x \mid \ell \mid f P \mid f(\theta_1, \dots, \theta_n)$

(iv) **Formulae:**

$F ::= p(\theta_1, \dots, \theta_n) \mid \neg F_1 \mid F_1 \wedge F_2 \mid \forall x \bullet F_1 \mid F_1 ; F_2$

(v) **Abbreviations:**

$[\] , [P] , [P]^t , [P]^{\leq t} , \diamond F , \square F$

Formulae: Remarks

Remark 2.10. [*Rigid and chop-free*] Let F be a duration formula, \mathcal{I} an interpretation, \mathcal{V} a valuation, and $[b, e] \in \text{Intv}$.

- If F is **rigid**, then

$$\forall [b', e'] \in \text{Intv} : \mathcal{I}[[F]](\mathcal{V}, [b, e]) = \mathcal{I}[[F]](\mathcal{V}, [b', e']).$$

- If F is **chop-free** or θ is **rigid**, then in the calculation of the semantics of F , every occurrence of θ denotes the same value.

“;”
does not
occur
in F

in F

e.g. $\underbrace{f(x) > 3; f(x) > 5}_{\theta}$

e.g. $\underbrace{l > 0}_{\theta} \wedge \underbrace{l > 1}_{\theta}$

$l > 0; l > 1$ is chop-free

Substitution Lemma

Lemma 2.11. [Substitution]

Consider a formula F , a global variable x , and a term θ such that F is **chop-free** or θ is **rigid**.

Then for all interpretations \mathcal{I} , valuations \mathcal{V} , and intervals $[b, e]$,

$$\mathcal{I}[\![F[x := \theta]]\!] (\mathcal{V}, [b, e]) = \mathcal{I}[\![F]] (\mathcal{V}[x := d], [b, e])$$

where $d = \mathcal{I}[\![\theta]] (\mathcal{V}, [b, e])$.

*syntactic
modification
of F*

*semantical
modification
of assignment*

Term $\rightarrow (l, x)$

- $F := \left((l = x); (l = x) \right) \implies l = 2 \cdot x, \theta := l \quad \mathcal{V}, [e, b] = [5, 11]$

- $\mathcal{I}[\![F[x := \theta]]\!] (\mathcal{V}, [e, b]) = \mathcal{I}[\![l = l; l = l \Rightarrow l = 2 \cdot l]\!] (\mathcal{V}, [e, 5]) = \text{ff}$ if $e < b$

- $\mathcal{I}[\![F]\!] (\mathcal{V}[x := 6], [e, b]) = \text{tt}$, F is even valid
 $d = \mathcal{I}[\![\theta]\!] (\mathcal{V}, [5, 11]) = 6$

Duration Calculus: Overview

We will introduce three (or five) syntactical “levels”:

(i) **Symbols:**

$f, g, \text{ true, false, } =, <, >, \leq, \geq, x, y, z, X, Y, Z, d$

(ii) **State Assertions:**

$P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \wedge P_2$

(iii) **Terms:**

$\theta ::= x \mid \ell \mid f P \mid f(\theta_1, \dots, \theta_n)$

(iv) **Formulae:**

$F ::= p(\theta_1, \dots, \theta_n) \mid \neg F_1 \mid F_1 \wedge F_2 \mid \forall x \bullet F_1 \mid F_1 ; F_2$

(v) **Abbreviations:**

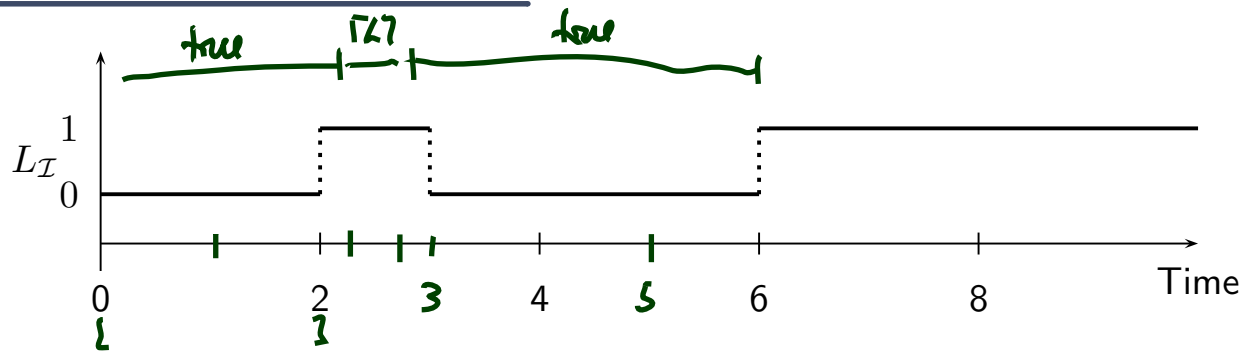
$[\] , [P] , [P]^t , [P]^{\leq t} , \diamond F , \square F$

Duration Calculus Abbreviations

Abbreviations

- $\lceil \rceil := \ell = 0$ (point interval)
- $\lceil P \rceil := \left(\int P = \ell \right) \wedge \ell > 0$ (almost everywhere)
- $\lceil P \rceil^t := \lceil P \rceil \wedge \ell = t$ (for time t)
- $\lceil P \rceil^{\leq t} := \lceil P \rceil \wedge \ell \leq t$ (up to time t)
- $\diamond F := \text{true} ; F ; \text{true}$ (for some subinterval)
- $\square F := \neg \diamond \neg F$ (for all subintervals)

Abbreviations: Examples



$$\mathcal{I}[\int L = 0] \quad \mathbb{I}(\mathcal{V}, [0, 2]) = \#$$

$$\mathcal{I}[\int L = 1] \quad \mathbb{I}(\mathcal{V}, [2, 6]) = \#$$

$$\mathcal{I}[\int L = 0 ; \int L = 1] \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\neg L] \quad \mathbb{I}(\mathcal{V}, [0, 2]) = \#$$

$$\mathcal{I}[L] \quad \mathbb{I}(\mathcal{V}, [2, 3]) = \#$$

$$\mathcal{I}[\neg L] ; [L] \quad \mathbb{I}(\mathcal{V}, [0, 3]) = \#$$

$$\mathcal{I}[\neg L] ; [L] ; [\neg L] \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\diamond [L]] \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\diamond [\neg L]] \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\diamond [\neg L]^2] \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\neg L]^2 ; [\neg L]^1 ; [\neg L]^3 \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$$\mathcal{I}[\neg L]^2 ; [L]^1 ; [\neg L]^3 \quad \mathbb{I}(\mathcal{V}, [0, 6]) = \#$$

$\int L = \text{len} > 0$

true; $\neg L$; true

unique chop point

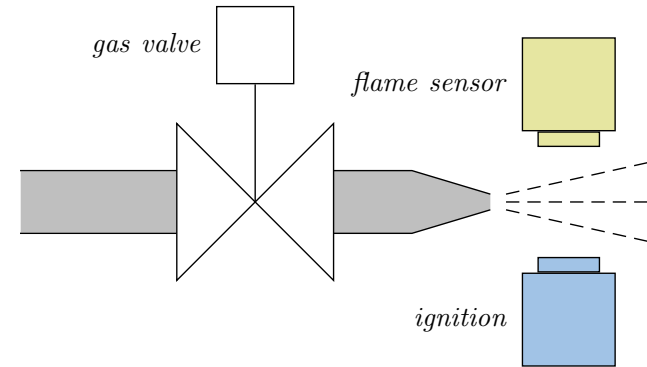
$2 \leq m_1 < m_2 \leq 3$
are witness chop points

Duration Calculus: Looking back

- Duration Calculus is an **interval logic**.
- Formulae are evaluated in an (**implicitly given**) interval.

Back to our gas burner:

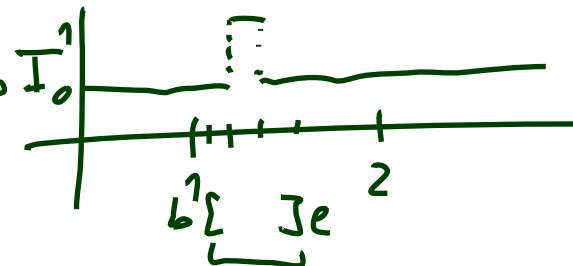
- $G, F, I, H, \quad \mathcal{D}(G) = \dots = \mathcal{D}(H) = \{0, 1\}$
- Define L as $G \wedge \neg F$.



Strangest operators:

- **almost everywhere** — Example: $\lceil G \rceil$
(Holds in a given interval $[b, e]$ iff the gas valve is open almost everywhere.)

- **chop** — Example: $\lceil \lceil \neg I \rceil ; \lceil I \rceil ; \lceil \neg I \rceil \rceil \implies \ell \geq 1$
(Ignition phases last at least one time unit.)



- **integral** — Example: $\ell \geq 60 \implies \int L \leq \frac{\ell}{20}$
(At most 5% leakage time within intervals of at least 60 time units.)

DC Validity, Satisfiability, Realisability

Validity, Satisfiability, Realisability

Let \mathcal{I} be an interpretation, \mathcal{V} a valuation, $[b, e]$ an interval, and F a DC formula.

- $\mathcal{I}, \mathcal{V}, [b, e] \models F$ (" F **holds** in $\mathcal{I}, \mathcal{V}, [b, e]$ ") iff $\mathcal{I}[[F]](\mathcal{V}, [b, e]) = \text{tt}$.
- F is called **satisfiable** iff it holds in some $\mathcal{I}, \mathcal{V}, [b, e]$.
- $\mathcal{I}, \mathcal{V} \models F$ (" \mathcal{I} and \mathcal{V} **realise** F ") iff $\forall [b, e] \in \text{Intv} : \mathcal{I}, \mathcal{V}, [b, e] \models F$.
- F is called **realisable** iff some \mathcal{I} and \mathcal{V} realise F .
- $\mathcal{I} \models F$ (" \mathcal{I} **realises** F ") iff $\forall \mathcal{V} \in \text{Val} : \mathcal{I}, \mathcal{V} \models F$.
- $\models F$ (" F is **valid**") iff \forall interpretation $\mathcal{I} : \mathcal{I} \models F$.

Validity vs. Satisfiability vs. Realisability

Remark 2.13. For all DC formulae F ,

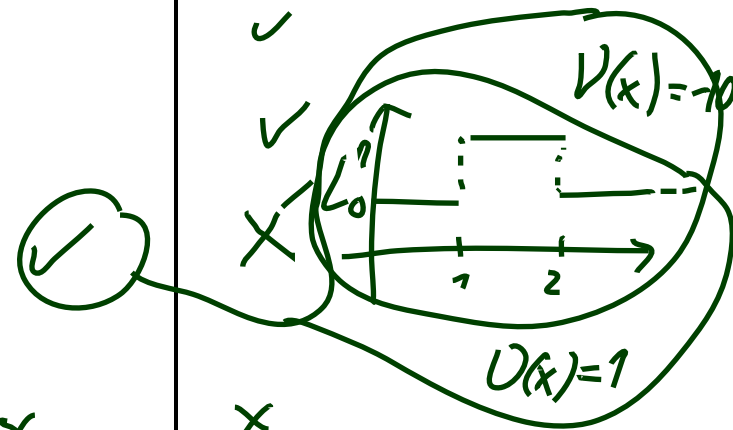
- F is satisfiable iff $\neg F$ is not valid,
 F is valid iff $\neg F$ is not satisfiable.
- If F is valid then F is realisable, but not vice versa.
- If F is realisable then F is satisfiable, but not vice versa.

Examples: Valid? Realisable? Satisfiable?

- $\mathcal{I}, \mathcal{V}, [b, e] \models F$ (" F **holds** in $\mathcal{I}, \mathcal{V}, [b, e]$ ") iff $\mathcal{I}[F](\mathcal{V}, [b, e]) = \text{tt}$.
- F is called **satisfiable** iff it holds in some $\mathcal{I}, \mathcal{V}, [b, e]$.
- $\mathcal{I}, \mathcal{V} \models F$ (" \mathcal{I} and \mathcal{V} **realise** F ") iff $\forall [b, e] \in \text{Intv} : \mathcal{I}, \mathcal{V}, [b, e] \models F$.
- F is called **realisable** iff some \mathcal{I} and \mathcal{V} realise F .
- $\mathcal{I} \models F$ (" \mathcal{I} **realises** F ") iff $\forall \mathcal{V} \in \text{Val} : \mathcal{I}, \mathcal{V} \models F$.
- $\models F$ (" F is **valid**") iff \forall interpretation $\mathcal{I} : \mathcal{I} \models F$.

	Satisfiable	Realisable	Valid
$l \geq 0$	✓	✓	✓
$l = f 1$			✓
$(l = 30) \iff ((l = 10); (l = 20))$			✓
$((F ; G) ; H) \iff (F ; (G ; H))$			✓
$f L \leq x$	✓	✓	✓
$l = 2$	✓	✗	✗
$l < 0$	✗	✗	✗

state as.



Initial Values

- $\mathcal{I}, \mathcal{V} \models_0 F$ (“ \mathcal{I} and \mathcal{V} **realise** F **from** 0”) iff
$$\forall t \in \text{Time} : \mathcal{I}, \mathcal{V}, \underbrace{[0, t]} \models F.$$
- F is called **realisable from 0** iff some \mathcal{I} and \mathcal{V} realise F from 0.
- Intervals of the form $[0, t]$ are called **initial intervals**.
- $\mathcal{I} \models_0 F$ (“ \mathcal{I} **realises** F **from** 0”) iff $\forall \mathcal{V} \in \text{Val} : \mathcal{I}, \mathcal{V} \models_0 F.$
- $\models_0 F$ (“ F is **valid from** 0”) iff \forall interpretation $\mathcal{I} : \mathcal{I} \models_0 F.$

Initial or not Initial...

For all interpretations \mathcal{I} , valuations \mathcal{V} , and DC formulae F ,

- (i) $\mathcal{I}, \mathcal{V} \models F$ implies $\mathcal{I}, \mathcal{V} \models_0 F$, but not vice versa,
- (ii) if F is realisable then F is realisable from 0, but not vice versa,
- (iii) F is valid iff F is valid from 0.

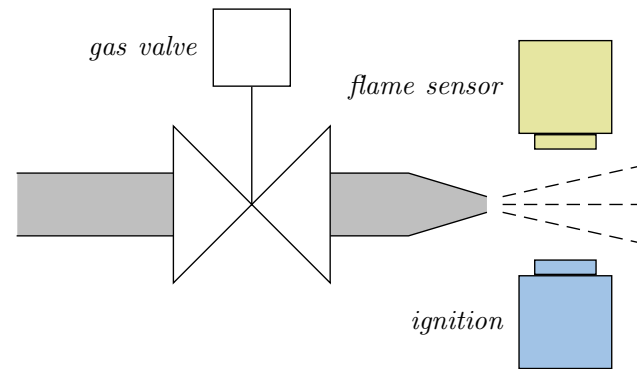
Specification and Semantics-based Correctness Proofs of Real-Time Systems with DC

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

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

Gas Burner Revisited



(i) Choose **observables**:

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

(output)

(input)

(ii) Provide the **requirement**:

$$\text{Req} : \iff \square(\ell \geq 60 \implies \int L \leq \ell)$$

Gas Burner Revisited

(iii) Provide a description 'Ctrl'

of the **controller** in form of a DC formula (over 'Obs').

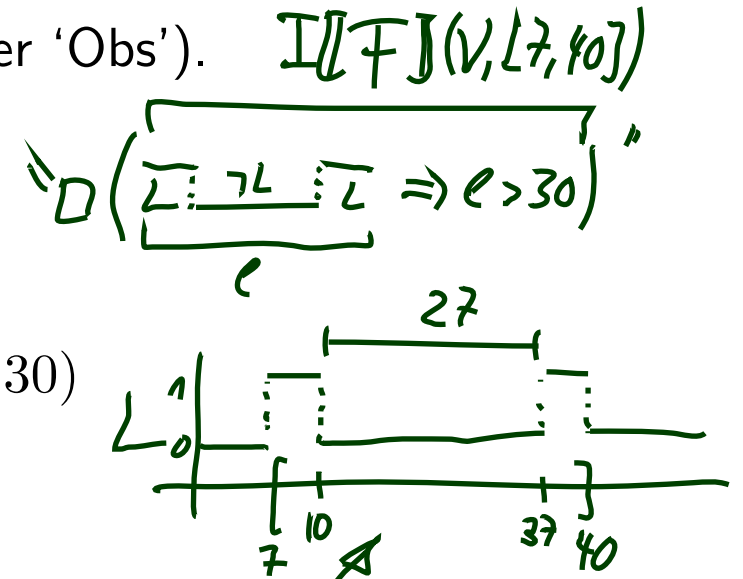
Here, firstly consider a **design**:

- Des-1 : $\iff \Box([L] \implies \ell \leq 1)$
- Des-2 : $\iff \Box([L] ; [\neg L] ; [L]) \implies \ell > 30)$

(iv) Prove **correctness**:

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

$$\models (\underline{\text{Des-1} \wedge \text{Des-2}} \implies \text{Req})$$



(Thm. 2.16)

not following Des-2 intuition

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 : $\iff \Box([\!L\!] \implies \ell \leq 1)$
- Des-2 : $\iff \Box([\!L\!] ; [\!\neg L\!] ; [\!L\!] \implies \ell > 30)$

(iv) Prove **correctness**:

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

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

- We do show

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

with the simplified requirement

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

- and we show

$$\models (\text{Des-1} \wedge \text{Des-2}) \implies \text{Req-1}. \quad (\text{Lem. 2.19}) \quad 21/36$$

References

References

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