# *Real-Time Systems*

# *Lecture 4: Duration Calculus II*

*2017-11-02*

Dr. Bernd Westphal

Albert-Ludwigs-Universität Freiburg, Germany

# Content

- **Formulae**
  - syntax, priority groups
  - syntactic substitution
  - semantics
  - well-definedness
  - remarks, substitution lemma
- **DC Abbreviations**
  - point interval, almost everywhere
  - for some subinterval / for all subintervals
- **Validity, Satisfiability, Realisability**
  - realisability / validity from $0$
- Proving design ideas correct: **Method**
  - Example: gas burner

# *Duration Calculus: Formulae*

# *Duration Calculus: Overview*

We will introduce **four syntactical categories** (and **abbreviations**):

(i) **Symbols:**

$$\overbrace{true, false, =, <, >, \leq, \geq}^{p,q}, \quad f, g, \quad X, Y, Z, \quad d, \quad x, y, z,$$

(ii) **State Assertions:**

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

(iii) **Terms:**

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

(iv) **Formulae:**

$$F ::= p(\theta_1, \dots, \theta_n) \mid \neg F_1 \mid F_1 \wedge F_2 \mid \forall\, x \bullet F_1 \mid F_1 \overset{\text{chop operator}}{;} F_2$$

(v) **Abbreviations:**

$$\lceil\,\rceil, \quad \lceil P \rceil, \quad \lceil P \rceil^t, \quad \lceil P \rceil^{\leq t}, \quad \Diamond F, \quad \Box F$$

# *Formulae: Syntax*

- The set of **DC formulae** is defined by the following grammar:

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

where $p$ is a predicate symbol, $\theta_i$ are terms, and $x$ is a global variable.

- **chop operator**: ‘**;**’
- **atomic formula**: $p(\theta_1, \dots, \theta_n)$
- **rigid formula**: all terms are rigid (no $\ell$, no $\int P$)
- **chop free**: ‘**;**’ doesn’t occur
- usual notion of **free** and **bound** (global) variables

$$\forall \underset{\text{bound}}{x} \bullet \int P > \underset{\text{bound}}{x} \land \underset{\text{free}}{y} < 10$$

- Note: quantification only over (**first-order**) global variables, not over (**second-order**) state variables.

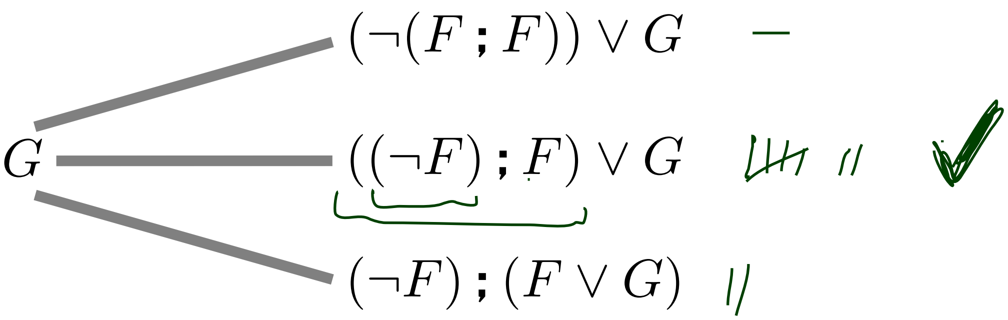
$\hookrightarrow \exists X \bullet \int X > 3$ <u>NOT</u>

# *Formulae: Priority Groups*

- To avoid parentheses, we define the following five **priority groups** from highest to lowest priority (or precedence):
  - $\neg$ (negation)
  - $;$ (chop)
  - $\wedge$, $\vee$ (and/or)
  - $\implies$, $\iff$ (implication/equivalence)
  - $\exists$, $\forall$ (quantifiers)

**Examples**:

- $\neg F \,;\, F \vee G$



$(\neg(F \,;\, F)) \vee G$ —

$\underbrace{(\underbrace{(\neg F)} \,;\, F)} \vee G$ 卌 || ✓

$(\neg F) \,;\, (F \vee G)$ ||

- $\forall\, x \bullet \quad F \quad \wedge \quad G$

# *Syntactic Substitution...*

...of a term $\theta$ for a variable $x$ in a formula $F$.

- We use

$$F[x := \theta]$$

  to denote the formula that results from performing the following steps:

  (i) transform $F$ into $\tilde{F}$ by (consistently) **renaming bound variables**
  such that **no free occurrence** of $x$ in $\tilde{F}$
  appears within a **quantified subformula** $\exists\, z \bullet G$ or $\forall\, z \bullet G$
  for some $z$ **occurring in term** $\theta$,

  (ii) textually replace all free occurrences of $x$ in $\tilde{F}$ by $\theta$.

# *Syntactic Substitution...*

...of a term $\theta$ for a variable $x$ in a formula $F$.

- We use

$$F[x := \theta]$$

to denote the formula that results from performing the following steps:

(i) transform $F$ into $\tilde{F}$ by (consistently) **renaming bound variables** such that **no free occurrence** of $x$ in $\tilde{F}$ appears within a **quantified subformula** $\exists\, z \bullet G$ or $\forall\, z \bullet G$ for some $z$ **occurring in term** $\theta$,

(ii) textually replace all free occurrences of $x$ in $\tilde{F}$ by $\theta$.

**Example**:

- $\theta_1 := \ell$, $\quad F[x := \theta_1] = (\ell \geq y \implies \exists\, z \bullet z \geq 0 \wedge \ell = y + z)$

# *Syntactic Substitution...*

...of a term $\theta$ for a variable $x$ in a formula $F$.

- We use

$$F[x := \theta]$$

  to denote the formula that results from performing the following steps:

  (i) transform $F$ into $\tilde{F}$ by (consistently) **renaming bound variables** such that **no free occurrence** of $x$ in $\tilde{F}$ appears within a **quantified subformula** $\exists\, z \bullet G$ or $\forall\, z \bullet G$ for some $z$ **occurring in term** $\theta$,

  (ii) textually replace all free occurrences of $x$ in $\tilde{F}$ by $\theta$.

**Example**:

- $\theta_1 := \ell, \quad F[x := \theta_1] = (\ell \geq y \implies \exists\, z \bullet z \geq 0 \wedge \ell = y + z)$

- $\theta_2 := \ell + z, \quad F \qquad\qquad = (x \qquad \geq y \implies \exists\, z \bullet z \geq 0 \wedge x \qquad = y + z)$

# *Syntactic Substitution...*

...of a term $\theta$ for a variable $x$ in a formula $F$.

- We use
  $$F[x := \theta]$$
  to denote the formula that results from performing the following steps:
  (i) transform $F$ into $\tilde{F}$ by (consistently) **renaming bound variables** such that **no free occurrence** of $x$ in $\tilde{F}$ appears within a **quantified subformula** $\exists\, z \bullet G$ or $\forall\, z \bullet G$ for some $z$ **occurring in term** $\theta$,
  (ii) textually replace all free occurrences of $x$ in $\tilde{F}$ by $\theta$.

**Example**:

- $\theta_1 := \ell, \quad F[x := \theta_1] = (\ell \geq y \implies \exists\, z \bullet z \geq 0 \wedge \ell = y + z)$
- $\theta_2 := \ell + z, \quad F[x := \theta_2] = (\ell + z \geq y \implies \underline{\exists\, z} \bullet z \geq 0 \wedge \ell + \overset{\text{suddenly bound}}{\boxed{z}} = y + z)$

# *Syntactic Substitution...*

...of a term $\theta$ for a variable $x$ in a formula $F$.

- We use

$$F[x := \theta]$$

to denote the formula that results from performing the following steps:

(i) transform $F$ into $\tilde{F}$ by (consistently) **renaming bound variables** such that **no free occurrence** of $x$ in $\tilde{F}$ appears within a **quantified subformula** $\exists\, z \bullet G$ or $\forall\, z \bullet G$ for some $z$ **occurring in term** $\theta$,

(ii) textually replace all free occurrences of $x$ in $\tilde{F}$ by $\theta$.

**Example**:

- $\theta_1 := \ell, \quad F[x := \theta_1] = (\ell \geq y \implies \exists\, z \bullet z \geq 0 \wedge \ell = y + z)$ ✓

- $\theta_2 := \ell + z, \quad F[x := \theta_2] = (\ell + z \geq y \implies \exists\, z \bullet z \geq 0 \wedge \ell + z = y + z)$ ✗

- $F[x := \theta_2] = \ell + z \geq y \implies \exists\, \tilde{z} \bullet \tilde{z} \geq 0 \wedge \ell + z = y + \tilde{z})$ ✓

# *Formulae: Semantics*

- The **semantics** of a **formula** is a function

$$\mathcal{I}[\![F]\!] : \mathsf{Val} \times \mathsf{Intv} \to \{\mathsf{tt}, \mathsf{ff}\}$$

$\mathcal{I}[\![F]\!](\mathcal{V}, [b,e])$: truth value of $F$ under interpretation $\mathcal{I}$ and valuation $\mathcal{V}$ in the interval $[b,e]$.

- $\mathcal{I}[\![F]\!](\mathcal{V}, [b,e])$ is defined **inductively** over the structure of $F$:

$$\mathcal{I}[\![p(\theta_1, \dots, \theta_n)]\!](\mathcal{V}, [b,e]) = \hat{p}(\,\mathcal{I}[\![\theta_1]\!](\mathcal{V}, [b,e]),\ \dots,\ \mathcal{I}[\![\theta_n]\!](\mathcal{V}, [b,e])\,),$$

base

step

$$\mathcal{I}[\![\neg F_1]\!](\mathcal{V}, [b,e]) = \mathsf{tt} \ \text{ iff } \ \mathcal{I}[\![F_1]\!](\mathcal{V}, [b,e]) = \mathsf{ff},$$

$$\mathcal{I}[\![F_1 \wedge F_2]\!](\mathcal{V}, [b,e]) = \mathsf{tt} \ \text{ iff } \ \mathcal{I}[\![F_i]\!](\mathcal{V}, [b,e]) = \mathsf{tt}, i \in \{1,2\},$$

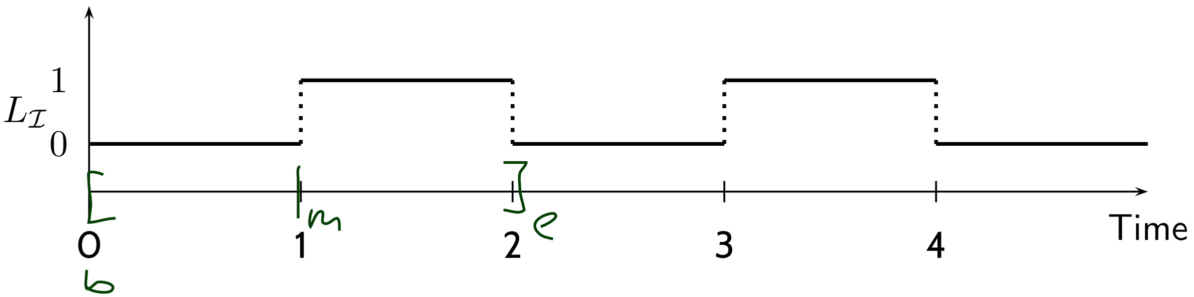
function symbol

$$\mathcal{I}[\![\forall\, x \bullet F_1]\!](\mathcal{V}, [b,e]) = \mathsf{tt} \ \text{ iff } \ \text{for all } a \in \mathbb{R},$$
$$\mathcal{I}[\![F_1[x := a]]\!](\mathcal{V}, [b,e]) = \mathsf{tt}$$

$$\mathcal{I}[\![F_1 \,;\, F_2]\!](\mathcal{V}, [b,e]) = \ \text{ iff } \ \text{there is an } m \in [b,e] \text{ such that}$$
$$\mathcal{I}[\![F_1]\!](\mathcal{V}, [b,m]) = \mathsf{tt} \text{ and } \mathcal{I}[\![F_2]\!](\mathcal{V}, [m,e]) = \mathsf{tt}.$$

# *Formulae: Example*

$$F := \int L = 0 \, ; \int L = 1$$



- $\mathcal{I}[\![F]\!](\mathcal{V}, [0,2]) = \mathsf{tt}$

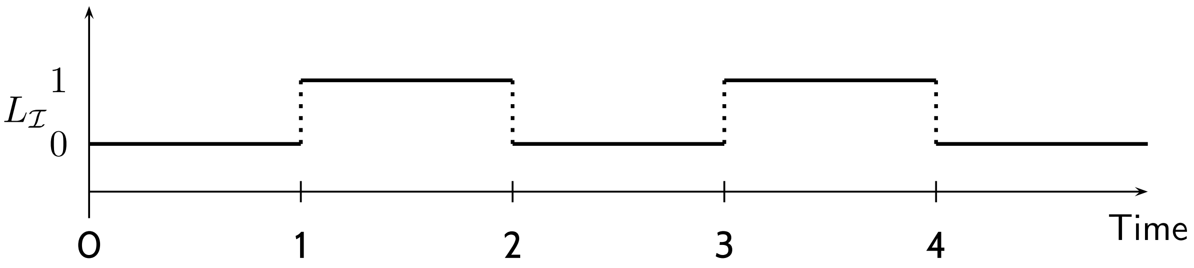
  **Proof**:

  - Choose $m = 1$ as **chop point**.

# Formulae: Example

$$F := \int L = 0 \,;\, \int L = 1$$

$$\equiv ((\int L) = 0) \;\;;\;\; ((\int L) = 1) \quad \equiv \quad = (\, (\int L), 0 \,) \;\;;\;\; = (\, (\int L), 1 \,)$$



- $\mathcal{I}[\![F]\!](\mathcal{V}, [0, 2]) = \mathrm{tt}$

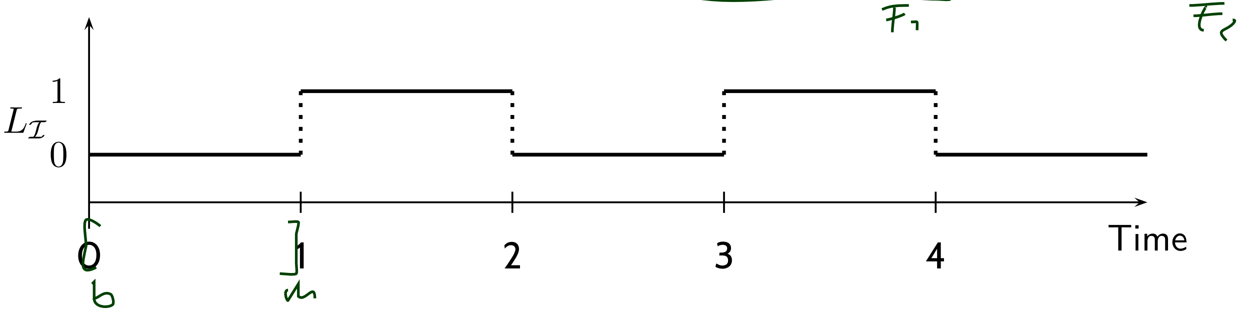
  **Proof**:

  - Choose $m = 1$ as **chop point**.

# *Formulae: Example*

$$F := \int L = 0 \,;\, \int L = 1$$

$$\equiv ((\int L) = 0) \;\;;\;\; ((\int L) = 1) \quad \equiv \underbrace{=(\,(\int L), 0\,)}_{F_1} \;;\; \underbrace{=(\,(\int L), 1\,)}_{F_2}$$



- $\mathcal{I}[\![F]\!](\mathcal{V}, [0,2]) = \text{tt}$

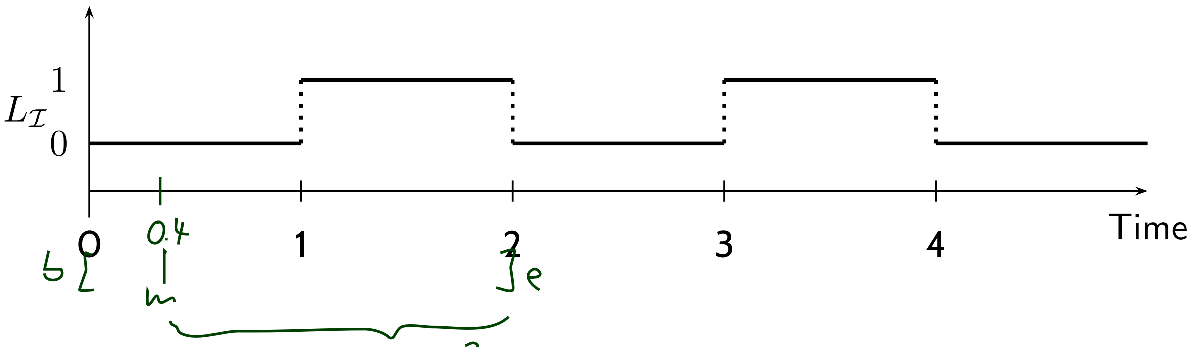
  **Proof**:

  - Choose $m = 1$ as **chop point**. Then
  - $\mathcal{I}[\![= ((\int L), 0)]\!](\mathcal{V}, [0,1]) \;= \hat{=}(\, \mathcal{I}[\![\int L]\!](\mathcal{V}, [0,1]),\, \mathcal{I}[\![0]\!](\mathcal{V}, [0,1])\,)$
    $$= \hat{=}\left(\int_0^1 L_\mathcal{I}(t)\; dt, \quad \hat{0}\right) \;= \hat{=}(0, 0) = \text{tt},$$

# Formulae: Example

$$F := \int L = 0 \,;\, \int L = 1$$

$$\equiv ((\textstyle\int L) = 0) \;\;;\;\; ((\textstyle\int L) = 1) \quad \equiv \quad =(\,(\textstyle\int L), 0\,) \;;\; =(\,(\textstyle\int L), 1\,)$$



$$\int_{0.4}^{2} L_\mathcal{I}(t)\,dt = 1$$

- $\mathcal{I}[\![F]\!](\mathcal{V}, [0,2]) =$ tt

  **Proof**:

  - Choose $m = 1$ as **chop point**. Then
  - $\mathcal{I}[\![= ((\int L), 0)]\!](\mathcal{V}, [0,1]) \;= \hat{=}(\,\mathcal{I}[\![\int L]\!](\mathcal{V}, [0,1]),\; \mathcal{I}[\![0]\!](\mathcal{V}, [0,1])\,)$

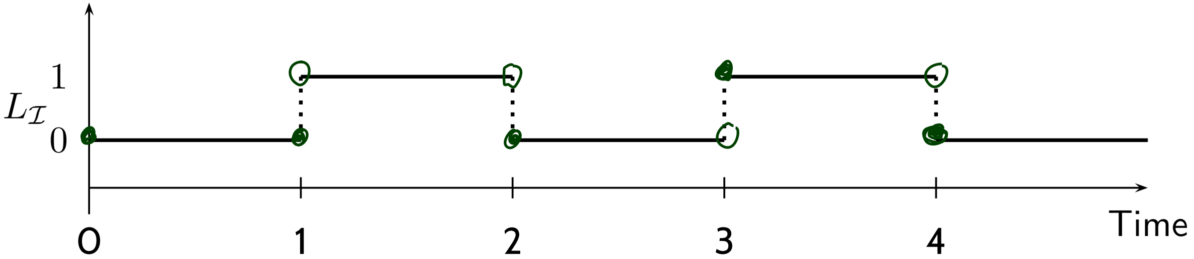
    $$= \hat{=}\left(\int_0^1 L_\mathcal{I}(t)\;dt, \quad \hat{0}\right) \;= \hat{=}(0,0) = \text{tt},$$
  - and $\mathcal{I}[\![= (\quad (\int L), \quad 1 \quad )]\!](\mathcal{V}, [1,2])$

    $= \hat{=}(\quad \mathcal{I}[\![\int L]\!](\mathcal{V}, [1,2]), \quad \mathcal{I}[\![1]\!](\mathcal{V}, [1,2]) \quad ) = \hat{=}(1,1) =$ tt, □

# *Formulae: Example*

$$F := \textstyle\int L = 0 \,;\, \int L = 1$$

$$\equiv ((\textstyle\int L) = 0) \;\; ; \;\; ((\int L) = 1) \quad \equiv \quad =(\,(\int L), 0\,) \;\; ; \;\; =(\,(\int L), 1\,)$$



- $\mathcal{I}[\![F]\!](\mathcal{V}, [0,2]) = \mathrm{tt}$

  **Proof**:

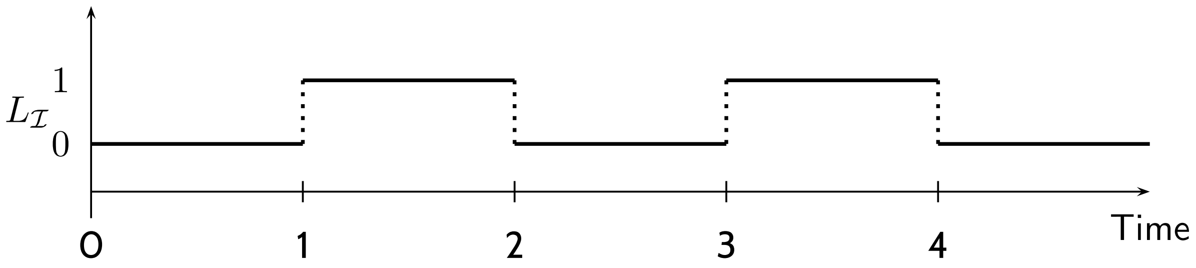
  - Choose $m = 1$ as **chop point**. Then
  - $\mathcal{I}[\![=((\int L), 0)]\!](\mathcal{V}, [0,1]) \;=\hat{=}(\,\mathcal{I}[\![\int L]\!](\mathcal{V}, [0,1]), \,\mathcal{I}[\![0]\!](\mathcal{V}, [0,1])\,)$
    $$= \hat{=}\left(\int_0^1 L_\mathcal{I}(t)\,dt, \quad \hat{0}\right) \;=\hat{=}(0,0) = \mathrm{tt},$$
  - and $\mathcal{I}[\![=(\quad (\int L), \quad 1 \quad)]\!](\mathcal{V}, [1,2])$
    $= \hat{=}(\quad \mathcal{I}[\![\int L]\!](\mathcal{V}, [1,2]), \quad \mathcal{I}[\![1]\!](\mathcal{V}, [1,2]) \quad) = \hat{=}(1,1) = \mathrm{tt},$ □

- Is the **chop point** $m$ **unique**?

# *Formulae: Example*

$$F := \int L = 0 \,\mathbf{;}\, \int L = 1$$

$$\equiv ((\int L) = 0) \;\;\mathbf{;}\;\; ((\int L) = 1) \quad \equiv \quad = (\,(\int L), 0\,) \;\mathbf{;}\; = (\,(\int L), 1\,)$$



- $\mathcal{I}[\![F]\!](\mathcal{V}, [0,2]) = $ tt

  **Proof**:

  - Choose $m = 1$ as **chop point**. Then
  - $\mathcal{I}[\![= ((\int L), 0)]\!](\mathcal{V}, [0,1]) \;= \hat{=}(\,\mathcal{I}[\![\int L]\!](\mathcal{V}, [0,1]), \,\mathcal{I}[\![0]\!](\mathcal{V}, [0,1])\,)$
    $$= \hat{=}\left(\int_0^1 L_\mathcal{I}(t)\,dt, \quad \hat{0}\right) = \hat{=}(0,0) = \text{tt},$$
  - and $\mathcal{I}[\![= (\quad (\int L), \quad 1 \quad)]\!](\mathcal{V}, [1,2])$
    $$= \hat{=}(\quad \mathcal{I}[\![\int L]\!](\mathcal{V}, [1,2]), \quad \mathcal{I}[\![1]\!](\mathcal{V}, [1,2]) \quad) = \hat{=}(1,1) = \text{tt},$$ □

NO, all $m \in [0,1]$ are proper chop points (and only those)

- Is the **chop point** $m$ **unique**?

• $\mathcal{I}[\![\int \neg L < 1 ; \int L < 1]\!](\mathcal{V}, [0,2]) = $ ff

- Would the **chop point** for formula $\int \neg L = 1 \,\mathbf{;}\, \int L = 1$ be **unique**?

# *Formulae: Remarks*

$x+y>3\ ;\ x+y>5$

- **rigid formula**: all terms are rigid
- **rigid term**: no length or integral operators
- **chop free**: ';' doesn't occur

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

- If $F$ is **rigid**, then

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

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

# *Substitution Lemma*

**Lemma 2.11.** [*Substitution*]
Consider a formula $F$, a global variable $x$, and a term $\theta$ such that $F$ is **chop-free** or $\theta$ is **rigid**.

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

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

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

- **Negative Example**: $F := (\ell = x) ; (\ell = x) \implies (\ell = 2 \cdot x), \quad \theta := \ell$
  - $\mathcal{I}[\![F[x := \ell]]\!](\mathcal{V}, [b, e]) = \mathcal{I}[\![(\ell = \ell) ; (\ell = \ell) \Rightarrow (\ell = 2 \cdot \ell)]\!](\mathcal{V}, [b, e])$

    ↳ yields ff for $b < e$
  - $\mathcal{I}[\![F]\!](\mathcal{V}[x := a], [b, e]) = \mathrm{tt}$ (even valid)

# *Duration Calculus: Overview*

We will introduce **four syntactical categories** (and **abbreviations**):

(i) **Symbols:**

$$\overbrace{true, false, =, <, >, \leq, \geq}^{p,q}, \quad f, g, \quad X, Y, Z, \quad d, \quad x, y, z,$$

(ii) **State Assertions:**

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

(iii) **Terms:**

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

(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:**

$$\lceil\, \rceil, \quad \lceil P \rceil, \quad \lceil P \rceil^t, \quad \lceil P \rceil^{\leq t}, \quad \Diamond F, \quad \Box F$$

# *Duration Calculus Abbreviations*

# Abbreviations

state assertion

- $\lceil\rceil := \ell = 0$ **(point interval)**
- $\lceil P \rceil := (\int P = \ell) \wedge (\ell > 0)$ **(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

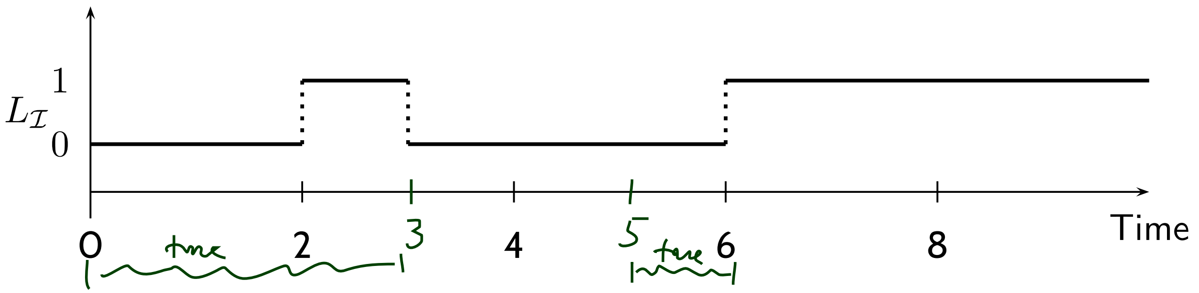
- $\Diamond F := \mathit{true} \,;\, F \,;\, \mathit{true}$ **(for some subinterval)**
- $\Box F := \neg \Diamond \neg F$ **(for all subintervals)**

box

⊚ $\Diamond \lceil P \rceil$ not satisfied
on any point interval

# Abbreviations: Examples

$(\int(\neg L)) = \ell \wedge \ell > 0$



true ; ⌈L⌉ ; true

$\mathcal{I}[\![ \, (\int L) = 0 \, ]\!](\mathcal{V}, \; [0,2] \;) = tt$

$\mathcal{I}[\![ \, \int L = 1 \, ]\!](\mathcal{V}, \; [2,6] \;) = tt$

$\mathcal{I}[\![ \, \int L = 0 \, ; \int L = 1 \, ]\!](\mathcal{V}, \; [0,6] \;) = tt \quad , \quad m = 2$

$\mathcal{I}[\![ \, \lceil \neg L \rceil \, ]\!](\mathcal{V}, \; [0,2] \;) = tt$

$\mathcal{I}[\![ \, \lceil L \rceil \, ]\!](\mathcal{V}, \; [2,3] \;) = tt$

$\mathcal{I}[\![ \, \lceil \neg L \rceil \, ; \lceil L \rceil \, ]\!](\mathcal{V}, \; [0,3] \;) = tt \quad , \quad m = 2$

$\mathcal{I}[\![ \, \lceil \neg L \rceil \, ; \lceil L \rceil \, ; \lceil \neg L \rceil \, ]\!](\mathcal{V}, \; [0,6] \;) = tt \quad , \quad m_1 = 2 , \quad m_2 = 3$

$\mathcal{I}[\![ \, \Diamond \lceil L \rceil \, ]\!](\mathcal{V}, \; [0,6] \;) = tt \quad m_1 = 2 \quad m_2 = 3$

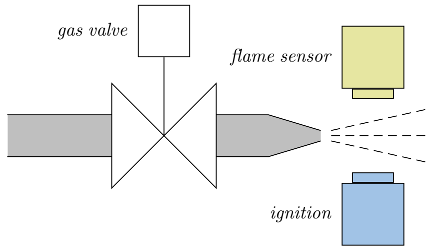
$\mathcal{I}[\![ \, \Diamond \lceil \neg L \rceil \, ]\!](\mathcal{V}, \; [0,6] \;) = .$

$\mathcal{I}[\![ \, \Diamond \lceil \neg L \rceil^2 \, ]\!](\mathcal{V}, \; [0,6] \;) = tt \quad m_1 = 3 \text{ or } 0 \quad m_2 = 5 \text{ or } 2$

$\mathcal{I}[\![ \, \Diamond \lceil \neg L \rceil^2 \, ; \lceil \cancel{\neg} L \rceil^1 \, ; \lceil \neg L \rceil^3 \, ]\!](\mathcal{V}, \; [0,6] \;) = tt \quad m_1 = 2 \quad m_2 = 3$

# *Duration Calculus: Preview*

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



- $G, F, I, H : \{0, 1\}$
- Define $L : \{0, 1\}$ as $G \wedge \neg F$.

$\lceil$Form$\rceil$

**Strangest operators**:

- **almost everywhere** – Example: $\lceil G \rceil$

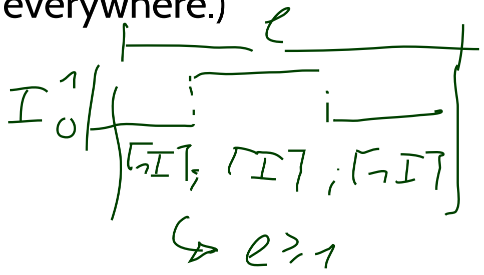
  (Holds in a given interval $[b, e]$ iff the gas valve is open almost everywhere.)

- **chop** ; – Example: $(\lceil \neg I \rceil \,;\, \lceil I \rceil \,;\, \lceil \neg I \rceil) \implies \ell \geq 1$

  (Ignition phases last at least one time unit.)

- **integral** $\int$ – Example: $\ell \geq 60 \implies \int L \leq \frac{\ell}{20}$

  (At most 5% leakage time within intervals of at least 60 time units.)



# *Content*

- **Formulae**
  - syntax, priority groups
  - syntactic substitution
  - semantics
  - well-definedness
  - remarks, substitution lemma
- **DC Abbreviations**
  - point interval, almost everywhere
  - for some subinterval / for all subintervals
- **Validity, Satisfiability, Realisability**
  - realisability / validity from $0$
- Proving design ideas correct: **Method**
  - Example: gas burner

# *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.

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

# Validity vs. Satisfiability vs. Realisability

**Remark 2.13.** For all DC formulae $F$,

- $F$ is satisfiable if and only if $\neg F$ is not valid,
  $F$ is valid if and only if $\neg F$ 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?*

- $\ell \geq 0$
- $\ell = \int 1$
- $\ell = 30 \iff \ell = 10 \,;\, \ell = 20$
- $((F \,;\, G) \,;\, H) \iff (F \,;\, (G \,;\, H))$
- $\int L \leq x$
- $\ell = 2$

# *Initial Values*

- $\mathcal{I}, \mathcal{V} \models_0 F$ (read: $\mathcal{I}$ and $\mathcal{V}$ **realise $F$ from $0$**) iff

$$\forall t \in \mathsf{Time} : \mathcal{I}, \mathcal{V}, [0,t] \models F.$$

- $F$ 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**.

- $\mathcal{I} \models_0 F$ (read: $\mathcal{I}$ **realises $F$ from $0$**) iff $\forall \mathcal{V} \in \mathsf{Val} : \mathcal{I}, \mathcal{V} \models_0 F.$

- $\models_0 F$ (read: $F$ is **valid from $0$**) iff $\forall \mathcal{I} : \mathcal{I} \models_0 F.$

# *Initial or not Initial...*

**Remark.** For all interpretations $\mathcal{I}$, valuations $\mathcal{V}$, and DC formulae $F$,

(i) $\mathcal{I}, \mathcal{V} \models F$ implies $\mathcal{I}, \mathcal{V} \models_0 F$,

(ii) if $F$ is realisable then $F$ is realisable from $0$, but not vice versa,

(iii) $F$ is valid iff $F$ is valid from $0$.

# Content

- Formulae
  - syntax, priority groups
  - syntactic substitution
  - semantics
  - well-definedness
  - remarks, substitution lemma
- DC Abbreviations
  - point interval, almost everywhere
  - for some subinterval / for all subintervals
- Validity, Satisfiability, Realisability
  - realisability / validity from $0$
- Proving design ideas correct: Method
  - Example: gas burner

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

# *Methodology (in an ideal world)*

In order to **prove** a controller design **correct** wrt. a **specification**:

(i) Choose **observables** 'Obs'.

(ii) Formalise the **requirements** 'Spec'
as a conjunction of DC formulae (over 'Obs').

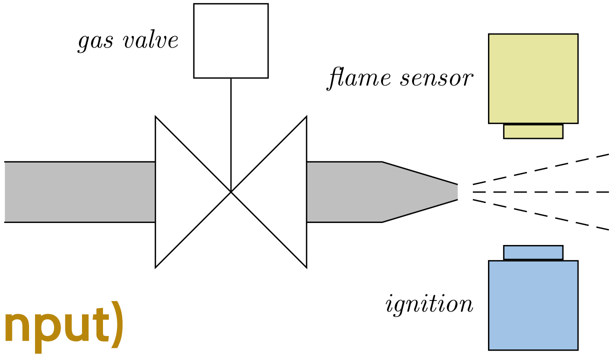
(iii) Formalise a **controller design** 'Ctrl'
as a conjunction of DC formulae (over 'Obs').

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

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

so "just" prove $\models_0 \mathsf{Ctrl} \implies \mathsf{Spec}$.

# *Gas Burner Revisited*



(i) Choose **observables**:

- $F : \{0,1\}$: value $1$ models "flame sensed now" **(input)**
- $G : \{0,1\}$: value $1$ models "gas valve is open now" **(output)**
- define $L := G \wedge \neg F$ to model **leakage**

(ii) Formalise the **requirement**:

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

"in each interval of length at least 60 time units, at most 5% of the time leakage"

(iii) Formalise **controller design ideas**:

- $\text{Des-1} := \Box(\lceil L \rceil \implies \ell \leq 1)$

  "leakage phases last for at most one time unit"
- $\text{Des-2} := \Box(\lceil L \rceil \,;\, \lceil \neg L \rceil \,;\, \lceil L \rceil \implies \ell > 30)$

  "non-leakage phases between two leakage-phases last at least 30 time units"

(iv) Prove **correctness**, i.e. prove $\models (\text{Des-1} \wedge \text{Des-2} \implies \text{Req})$.

(Or do we want "$\models_0$"...?)

# Content

- Formulae
  - syntax, priority groups
  - syntactic substitution
  - semantics
  - well-definedness
  - remarks, substitution lemma
- DC Abbreviations
  - point interval, almost everywhere
  - for some subinterval / for all subintervals
- Validity, Satisfiability, Realisability
  - realisability / validity from $0$
- Proving design ideas correct: Method
  - Example: gas burner

# *Tell Them What You've Told Them...*

- **Duration Calculus Formulae**
  - using, e.g., the **chop operator**

  are **evaluated** for **intervals** and **valuations**.

  The **semantics** of a **DC formula** is a **truth value**.

- The following **abbreviations** are sometimes useful
  - **point interval** ($\lceil\rceil$), **almost everywhere** ($\lceil P \rceil$),
  - **for some subinterval** ($\Diamond F$), **for all subintervals** ($\Box F$)

- **DC Formulae** have notions of
  - **satisfiability** and **validity** (as usual),
  - **realisability** ("for all subintervals")
  - also: from $0$

- Outlook on next lecture:
  proving design ideas correct wrt. requirements.

# *References*

# References

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

EXAM

- oral / written

- DATE
  (mid / late March )

↳ Tue fix on