Advanced System Software

Advanced Topics: Duration Calculus

Takuo Watanabe

Department of Computer Science, Tokyo Institute of Technology

Nov. 19, 2018

Agenda

- Introduction
  - Modeling Real-Time Systems using Time Intervals
  - Introduction to Duration Calculus
- Duration Calculus
  - Syntax
  - Semantics
  - Proof Rules

Textbook for This Topic

- Real-Time Systems: Formal Specification and Automatic Verification, Chapter 1–3
  - Ernst-Rüdiger Olderog and Henning Dierks
  - Cambridge University Press, 2008.
  - ISBN: 9780521883337
  - DOI: http://doi.org/10.1017/CBO9780511619953
- Duration Calculus: A Formal Approach to Real-Time Systems
  - Zhou Chaochen and Michael R. Hansen
  - Springer, 2004.
  - ISBN: 9783642074042
  - DOI: http://doi.org/10.1007/978-3-662-06784-0

Introduction

Modeling Real-Time Systems using Time Intervals

Observables

To describe real-time system formally, we represent them by a collection of time-dependent state variables (*time-varying variables* or *observables*) such as

$$\mathrm{obs} : \mathrm{Time} \to \mathcal{D}.$$

- Time: Time Domain
  - Continuous: $\mathrm{Time} \cong \mathbb{R}_{\geq 0}$
  - Discrete: $\mathrm{Time} \cong \mathbb{N}$
- $\mathcal{D}$: data type of obs

Example

$$\begin{aligned} G &: \mathrm{Time} \to \{0, 1\} \\ \mathrm{track} &: \mathrm{Time} \to \{\mathrm{empty},\ \mathrm{appr},\ \mathrm{cross}\} \\ \mathrm{gate} &: \mathrm{Time} \to [0, 90] \end{aligned}$$

Example: Safety and Liveness Properties

Let B, G be Boolean observables (*i.e.*, $B, G : \text{Time} \to \{0, 1\}$).

- Safety Properties: "Something bad never happens"

$$\forall t \in \text{Time}. \neg B(t)$$

- Liveness Properties: "Something good eventually happens"

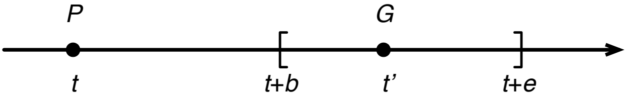
$$\exists t \in \text{Time}. G(t)$$

Note: We use 0 and 1 to denote Boolean values false and true respectively.

Bounded Response Properties

A desired reaction to an input at t should occur within a specified time interval $[t + b,\ t + e]$ $(0 < b \leq e)$.

$$\forall t \in \mathrm{Time}.[P(t) \Rightarrow \exists t' \in [t + b,\ t + e].G(t')].$$



Duration Properties

For any time intervals $[b, e]$ satisfying a condition $A(b, e)$, the accumulated time in which a condition C holds has an upper bound $u(b, e)$.

$$\forall [b, e] \in \mathrm{Intv}. \left[A(b, e) \Rightarrow \int_b^e C(t)dt \leq u(b, e) \right]$$

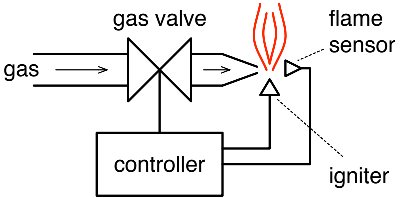
where Intv is the set of all closed time intervals, i.e.,

$$\mathrm{Intv} \stackrel{\mathrm{def}}{=} \{[b, e] \mid b, e \in \mathrm{Time} \wedge b \leq e\}.$$

Note: The range of C and u is a subdomain of $\mathcal{R}$ and C should be Riemann-integrable over $[b, e]$.

Example: Gas Burner

System Description



The state of the system is represented by two Boolean observables $F, G : \text{Time} \rightarrow \{0, 1\}$. G describes whether the gas valve is open and F describes whether the flame is detected by the flame sensor.

Example: Gas Burner

Gas Leakage

The leakage of the gas can be described by a Boolean observable L defined as $L(t) = G(t) \times (1 - F(t))$.

Requirement (Req)

For each time interval of at least 60 seconds duration the leakage periods do not exceed 5% of the duration.

Formalization Req

Using L, we can formalize the requirement as follows:

$$\text{Req} \overset{\text{def}}{\Longleftrightarrow}$$

$$\forall [b,\, e] \in \text{Intv.} \left(e - b \geq 60 \Rightarrow \int_b^e L(t)dt \leq \frac{e-b}{20} \right).$$

Example: Gas Burner

Design Constraints

- Des-1: The controller can stop each leak within a second:

$$\text{Des-1} \overset{\text{def}}{\Longleftrightarrow} \forall [b,\ e] \in \text{Intv}.(\forall t \in [b, e].L(t) \Rightarrow e - b \leq 1).$$

- Des-2: After each leak the controller waits for 30 seconds before opening the valve and igniting the gas again:

$$\text{Des-2} \overset{\text{def}}{\Longleftrightarrow} \forall [b,\ e] \in \text{Intv}.(\ L(b) \land L(e) \land (\exists t \in [b, e].\neg L(t)) \Rightarrow e - b \geq 30\ ).$$

Example: Gas Burner

Correctness

If the controller is constructed to satisfy the two design constraints, the system satisfies the requirement:

$$\text{Des-1} \land \text{Des-2} \Rightarrow \text{Req}.$$

In other words, for all interpretations of F and G satisfying Des-1 and Des-2, the safety requirement Req holds.

Introduction

Introduction to Duration Calculus

Duration Calculus (DC)

- An interval temporal logic for continuous time introduced by Z. Chaochen, C. A. R. Hoare and A. P. Ravn.
- Example: Gas Burner

$$\begin{aligned}
\text{Req} &\overset{\text{def}}{\Longleftrightarrow} \Box(\ell \geq 60 \Rightarrow \textstyle\int L \leq \ell/20) \\
\text{Des-1} &\overset{\text{def}}{\Longleftrightarrow} \Box(\lceil L \rceil \Rightarrow \ell \leq 1) \\
\text{Des-2} &\overset{\text{def}}{\Longleftrightarrow} \Box(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil \Rightarrow \ell > 30).
\end{aligned}$$

DC Formulas

- State Assertions:

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

- Terms:

$$\theta ::= x \mid \ell \mid \int P \mid f(\theta, \ldots, \theta).$$

- Formulas:

$$F ::= p(\theta, \ldots, \theta) \mid \neg F \mid F \wedge F \mid \forall x.F \mid F; F.$$

Symbols

Metavariables

- Function symbols (constants): f, g, h, $\ldots$
- Predicate symbols (constants): p, q, r, $\ldots$
- Global variables: x, y, z, $\ldots \in \mathsf{GVar}$
- State variables (Observables): X, Y, Z, $\ldots \in \mathsf{Obs}$

Note: Function/predicate symbols include constants such as 0, 1, 2, true, false. We use infix notations for some functions and predicates such as $+$, $-$, $\times$, $=$, $\neq$, $<$, $>$, $\leq$, $\geq$.

Symbols

Interpretations of Function and Predicate Symbols

- An n-ary function symbol f is interpreted as a real funtion $\hat{f} : \mathbb{R}^n \to \mathbb{R}$.
- An n-ary predicate symbol p is interpreted as a predicate $\hat{p} : \mathbb{R}^n \to \{\text{tt}, \text{ff}\}$.

Example

- $\widehat{\text{true}} = \text{tt}$, $\widehat{\text{false}} = \text{ff}$.
- $\hat{0}, \hat{1}, \hat{2}, \ldots \in \mathbb{R}$.
- $\hat{+}, \hat{-}, \hat{\times}, \ldots : \mathbb{R}^2 \to \mathbb{R}$.
- $\hat{=}, \hat{\neq}, \hat{<}, \hat{\leq}, \ldots : \mathbb{R}^2 \to \{\text{tt}, \text{ff}\}$.

For simplicity, we use use symbols $0, 1, +, -, =, <, \ldots$ to mean $\hat{0}, \hat{1}, \hat{+}, \hat{-}, \hat{=}, \hat{<}, \ldots$ unless otherwise specified.

Symbols

Interpretation of Global Variables

The semantics of a global variable is given by a *valuation* $\mathcal{V}$.

$$\mathcal{V}(x) \in \mathbb{R}.$$

$\mathsf{Val} = \mathsf{GVar} \to \mathbb{R}$. The adjective *global* means that the value of a global variable is independent of the time.

Interpretation of State Variables (Observables)

The semantics of a state variable is given by an *interpretation* $\mathcal{I}$.

$$\mathcal{I}(X) : \mathsf{Time} \to \mathcal{D}.$$

We write $X_{\mathcal{I}}$ instead of $\mathcal{I}(X)$ for simplicity.

State Assertions

Basic Syntax

The set of *state assertions* (P, Q, R, $\ldots$) are defined by the following abstract syntax:

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

where d denotes a constant of the data type of X.

Extensions

Other logical connectives are defined as follows:

$$\begin{aligned}
P \vee Q &\equiv \neg(\neg P \wedge \neg Q), \\
P \Rightarrow Q &\equiv \neg P \vee Q, \\
P \Leftrightarrow Q &\equiv (P \Rightarrow Q) \wedge (Q \Rightarrow P).
\end{aligned}$$

State Assertions

Boolean State Variables

For a Boolean state variable (observable) X, we write X to represent $X = 1$.

Operator Precedence

We adopt the following (usual) operator precedence for logical connectives:

$$\neg > \wedge, \vee \; > \Rightarrow, \Leftrightarrow .$$

For example, $\neg P \wedge Q \Rightarrow R$ stands for $((\neg P) \wedge Q) \Rightarrow R$.
In addition, $\Rightarrow$ is right associative. So $P \Rightarrow Q \Rightarrow R$ stands for $P \Rightarrow (Q \Rightarrow R)$.

State Assertions

Semantics

The semantics of a state assertion is defined as a function

$$\mathcal{I}[\![P]\!] : \mathsf{Time} \to \{0, 1\}$$

where $\mathcal{I}$ is an interpretation. The function is defined inductively on the structure of P:

$$\begin{aligned}
\mathcal{I}[\![0]\!](t) &= 0, \\
\mathcal{I}[\![1]\!](t) &= 1, \\
\mathcal{I}[\![X = d]\!](t) &= \begin{cases} 1 & (\text{if } X_{\mathcal{I}}(t) = d) \\ 0 & (\text{otherwise}), \end{cases} \\
\mathcal{I}[\![\neg P]\!](t) &= 1 - \mathcal{I}[\![P]\!](t), \\
\mathcal{I}[\![P \wedge Q]\!](t) &= \mathcal{I}[\![P]\!](t) \cdot \mathcal{I}[\![Q]\!](t).
\end{aligned}$$

We write $P_{\mathcal{I}}$ instead of $\mathcal{I}[\![P]\!]$ for simplicity.

Terms

Syntax

The set of *duration terms* (*DC terms* or just *terms*) is defined inductively by the following abstract syntax:

$$\theta ::= x \mid \ell \mid \int P \mid f(\theta, \ldots, \theta).$$

The symbol ℓ called the *length operator* and the symbol $\int$ is called the *integral operator*.

Rigid Terms

Terms without the symbols ℓ and $\int$ are called *rigid*.

Terms

Semantics

The semantics of a term is defined as a function

$$\mathcal{I}[\![\theta]\!] : \mathsf{Val} \times \mathsf{Intv} \to \mathbb{R}$$

where $\mathcal{I}$ is an interpretation. The function is defined inductively on the structure of θ:

$$\begin{array}{rcl}
\mathcal{I}[\![x]\!](\mathcal{V}, [b, e]) & = & \mathcal{V}(x), \\
\mathcal{I}[\![\ell]\!](\mathcal{V}, [b, e]) & = & e - b, \\
\mathcal{I}[\![\int P]\!](\mathcal{V}, [b, e]) & = & \int_b^e P_{\mathcal{I}}(t)dt, \\
\mathcal{I}[\![f(\theta_1, \ldots, \theta_n)]\!](\mathcal{V}, [b, e]) & = & \\
& & \hat{f}(\mathcal{I}[\![\theta_1]\!](\mathcal{V}, [b, e]), \ldots, \mathcal{I}[\![\theta_n]\!](\mathcal{V}, [b, e])).
\end{array}$$

Note that the semantics of a rigid term does not depend on time intervals.

Terms

Finite Variability

We assume the following condition:

> *For each state variable X and each interval $[b, e]$ there is a finite partition of $[b, e]$ such that the interpretation $X_{\mathcal{I}}$ is constant on each part. Thus on each interval $[b, e]$ the function $X_{\mathcal{I}}$ has only finitely many points of discontinuity.*

This condition guarantees the integrability of $P_{\mathcal{I}}$.

Formulas

Syntax

The set of *duration formulas* (*DC formulas* or just *formulas*) is defined inductively by the following abstract syntax:

$$F ::= p(\theta, \ldots, \theta) \mid \neg F \mid F \wedge F \mid \forall x.F \mid F; F.$$

The symbol ; called the *chop operator*.

Extensions and Operator Precedence

We introduce formulas using $\vee$, $\Rightarrow$ and $\Leftrightarrow$ as usual extensions to the above basic syntax. In addition, we use the following extension and the operator precedence:

- $\exists x.F \equiv \neg\forall x.\neg F$,
- $\neg > ; > \wedge, \vee > \Rightarrow, \Leftrightarrow > \forall, \exists.$

Formulas

Derived Formulas

$$\Diamond F \overset{\text{def}}{\iff} \text{true}; F; \text{true}$$

$$\Box F \overset{\text{def}}{\iff} \neg \Diamond \neg F$$

$$\lceil\rceil \overset{\text{def}}{\iff} \ell = 0$$

$$\lceil P \rceil \overset{\text{def}}{\iff} \int P = \ell \land \ell > 0$$

$$\lceil P \rceil^{t} \overset{\text{def}}{\iff} \int P = \ell \land \ell = t$$

$$\lceil P \rceil^{\leq t} \overset{\text{def}}{\iff} \int P = \ell \land \ell \leq t$$

Formulas

Semantics

The semantics of a term is defined as a function

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

where $\mathcal{I}$ is an interpretation. The function is defined inductively on the structure of F:

$$\begin{array}{l}
\mathcal{I}[\![p(\theta_1, \ldots, \theta_n)]\!](\mathcal{V}, [b, e]) = \\
\qquad\qquad \hat{p}(\mathcal{I}[\![\theta_1]\!](\mathcal{V}, [b, e]), \ldots, \mathcal{I}[\![\theta_n]\!](\mathcal{V}, [b, e])),
\end{array}$$

$$\begin{array}{rcl}
\mathcal{I}[\![\neg F]\!](\mathcal{V}, [b, e]) & = & \neg \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]), \\
\mathcal{I}[\![F \wedge G]\!](\mathcal{V}, [b, e]) & = & \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]) \wedge \mathcal{I}[\![G]\!](\mathcal{V}, [b, e]), \\
\mathcal{I}[\![\forall x . F]\!](\mathcal{V}, [b, e]) & = & \forall d \in \mathbb{R} . \mathcal{I}[\![F]\!](\mathcal{V}[x := d], [b, e]), \\
\mathcal{I}[\![F ; G]\!](\mathcal{V}, [b, e]) & = & \\
\multicolumn{3}{r}{\exists m \in [b, e] . \mathcal{I}[\![F]\!](\mathcal{V}, [b, m]) \wedge \mathcal{I}[\![G]\!](\mathcal{V}, [m, e]).}
\end{array}$$

Summary

- Observables
- Example: Gas Burner System
- Syntax of Duration Calculus (DC)
  - Symbols
  - State Assertions
  - Terms
  - Formulas

Duration Calculus

Syntax

DC Syntax: Symbols

- $X, Y, Z, \ldots \in \text{Obs}$: state variables (observables)
- $x, y, z, \ldots \in \text{GVar}$: global variables
- $f, g, h, \ldots$: function symbols
- $p, q, r, \ldots$: predicate symbols
- $P, Q, R, \ldots$: state assertions
- $\theta, \theta', \theta'', \ldots$: DC terms
- $F, G, H, \ldots$: DC formulas

Note: Function/predicate symbols include constants such as 0, 1, 2, true, false. We use infix notations for some functions and predicates such as $+$, $-$, $\times$, $=$, $\neq$, $<$, $>$, $\leq$, $\geq$.

DC Syntax: Core Syntax

$$
\begin{aligned}
P &::= 0 \mid 1 \mid X = d \mid \neg P \mid P \wedge P && \text{(state assertions)}\\
\theta &::= x \mid \ell \mid \textstyle\int P \mid f(\theta, \ldots, \theta) && \text{(DC terms)}\\
F &::= p(\theta, \ldots, \theta) \mid \neg F \mid F \wedge F \mid \forall x.F \mid F ; F && \text{(DC formulas)}
\end{aligned}
$$

Note: d is a constant of the type of X. If the domain of X is $\{0, 1\}$, we write X for $X = 1$.

DC Syntax: Derived Syntax (1)

$$
\begin{aligned}
P \lor Q &\overset{\text{def}}{\iff} \neg(\neg P \land \neg Q) \\
P \Rightarrow Q &\overset{\text{def}}{\iff} \neg P \lor Q \\
P \Leftrightarrow Q &\overset{\text{def}}{\iff} (P \Rightarrow Q) \land (Q \Rightarrow P)
\end{aligned}
$$

$$
\begin{aligned}
F \lor G &\overset{\text{def}}{\iff} \neg(\neg F \land \neg G) \\
F \Rightarrow G &\overset{\text{def}}{\iff} \neg F \lor G \\
F \Leftrightarrow G &\overset{\text{def}}{\iff} (F \Rightarrow G) \land (G \Rightarrow F) \\
\exists x.F &\overset{\text{def}}{\iff} \neg \forall x. \neg F
\end{aligned}
$$

DC Syntax: Derived Syntax (2)

$$\Diamond F \overset{\text{def}}{\iff} \text{true}; F; \text{true}$$

$$\Box F \overset{\text{def}}{\iff} \neg\Diamond\neg F$$

$$\lceil\rceil \overset{\text{def}}{\iff} \ell = 0$$

$$\lceil P \rceil \overset{\text{def}}{\iff} \int P = \ell \wedge \ell > 0$$

$$\lceil P \rceil^{t} \overset{\text{def}}{\iff} \int P = \ell \wedge \ell = t$$

$$\lceil P \rceil^{\leq t} \overset{\text{def}}{\iff} \int P = \ell \wedge \ell \leq t$$

Duration Calculus

Semantics

Semantics: Interpretation of State Variables

Interpretation

An *interpretation* $\mathcal{I}$ gives the value of a given state variable at a given time point. It can be written as

$$\mathcal{I}(X) : \mathsf{Time} \to \mathcal{D}$$

where $\mathcal{D}$ is the type of X.

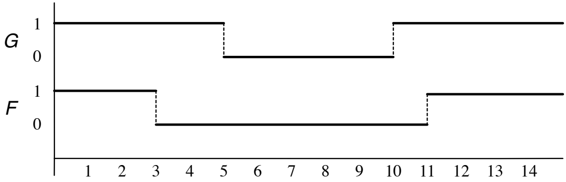
Notation:

We write $X_{\mathcal{I}}$ instead of $\mathcal{I}(X)$ for simplicity.

$$X_{\mathcal{I}}(t) = \mathcal{I}(X)(t)$$

Example 1

Let G and F be state variables of type $\{0, 1\}$. Let $\mathcal{I}$ be an interpretation defined as the following diagram.

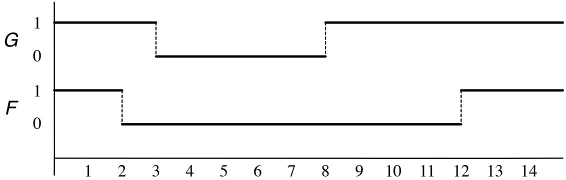


Under $\mathcal{I}$, for example, we have

$$G_{\mathcal{I}}(4) = 1,\ F_{\mathcal{I}}(4) = 0,\ G_{\mathcal{I}}(9) = 0,\ F_{\mathcal{I}}(11.5) = 1.$$

Example 2

Let $\mathcal{I}'$ be an interpretation defined as the following diagram.

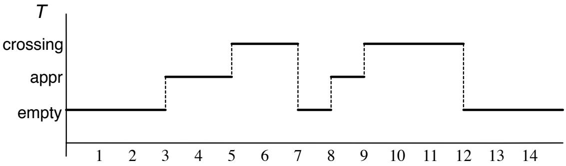


Under $\mathcal{I}'$, for example, we have

$$G_{\mathcal{I}'}(4) = 0,\ F_{\mathcal{I}'}(4) = 0,\ G_{\mathcal{I}'}(9) = 1,\ F_{\mathcal{I}'}(11.5) = 0.$$

Example 3

Let T be a state variable of type $\{\text{empty}, \text{appr}, \text{crossing}\}$. Let $\mathcal{I}''$ be an interpretation defined as the following diagram.



Under $\mathcal{I}''$, for example, we have

$$T_{\mathcal{I}''}(2) = \text{empty},\ T_{\mathcal{I}''}(4.5) = \text{appr},\ T_{\mathcal{I}''}(6) = \text{crossing},$$
$$T_{\mathcal{I}''}(7.5) = \text{empty},\ T_{\mathcal{I}''}(10) = \text{crossing}.$$

Semantics of State Assertions

Let $\mathcal{I}$ be a given interpretation. The semantics of a state assertion P (under $\mathcal{I}$) is a function

$$\mathcal{I}[\![P]\!] : \mathsf{Time} \to \{0, 1\}$$

that is defined inductively on the structure of P:

$$\begin{aligned}
\mathcal{I}[\![0]\!](t) &= 0, \\
\mathcal{I}[\![1]\!](t) &= 1, \\
\mathcal{I}[\![X = d]\!](t) &= \begin{cases} 1 & (\text{if } X_{\mathcal{I}}(t) = d) \\ 0 & (\text{otherwise}), \end{cases} \\
\mathcal{I}[\![\neg P]\!](t) &= 1 - \mathcal{I}[\![P]\!](t), \\
\mathcal{I}[\![P \wedge Q]\!](t) &= \mathcal{I}[\![P]\!](t) \cdot \mathcal{I}[\![Q]\!](t).
\end{aligned}$$

We write $P_{\mathcal{I}}$ instead of $\mathcal{I}[\![P]\!]$ for simplicity.

Example 4

Let L be a state assertion defined as

$$L \overset{\text{def}}{\iff} G \wedge \neg F.$$

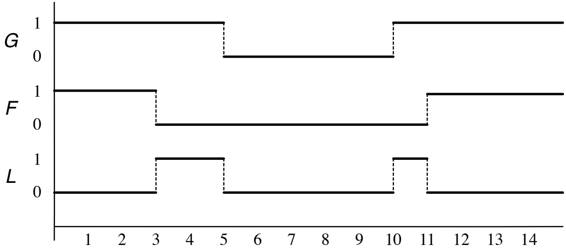
$\mathcal{I}[\![L]\!]$ satisfies

$$\begin{aligned}
\mathcal{I}[\![L]\!](t) &= \mathcal{I}[\![G \wedge \neg F]\!](t) \\
&= \mathcal{I}[\![G]\!](t) \cdot \mathcal{I}[\![\neg F]\!](t) \\
&= \mathcal{I}[\![G]\!](t) \cdot (1 - \mathcal{I}[\![F]\!](t)) \\
&= G_{\mathcal{I}}(t) \cdot (1 - F_{\mathcal{I}}(t)).
\end{aligned}$$

The last equation holds because the type of G and F is $\{0, 1\}$.

Example 4

Under the interpretation $\mathcal{I}$ in Example 1, the semantics (or interpretation) of L can be depicted as:



Interpretation of Global Variables

Valuation

A *valuation* is a function

$$\mathcal{V} : \mathsf{GVar} \to \mathbb{R}$$

that gives the value of a given global variable.
We use Val to denote the set of all valuations (i.e., $\mathsf{Val} = \mathsf{GVar} \to \mathbb{R}$).

Note:

The adjective *global* means that the value of a global variable is independent of the time.

Interpretations of Function/Predicate Symbols

- An n-ary function symbol f is interpreted as a real funtion $\hat{f} : \mathbb{R}^n \to \mathbb{R}$.
- An n-ary predicate symbol p is interpreted as a predicate $\hat{p} : \mathbb{R}^n \to \{\mathrm{tt}, \mathrm{ff}\}$.

Example

- $\widehat{\mathrm{true}} = \mathrm{tt}$, $\widehat{\mathrm{false}} = \mathrm{ff}$.
- $\hat{0}, \hat{1}, \hat{2}, \ldots \in \mathbb{R}$.
- $\hat{+}, \hat{-}, \hat{\times}, \ldots : \mathbb{R}^2 \to \mathbb{R}$.
- $\hat{=}, \hat{\neq}, \hat{<}, \hat{\leq}, \ldots : \mathbb{R}^2 \to \{\mathrm{tt}, \mathrm{ff}\}$.

For simplicity, we use use (abuse) symbols $0, 1, +, -, =, <, \ldots$ to mean $\hat{0}, \hat{1}, \hat{+}, \hat{-}, \hat{=}, \hat{<}, \ldots$ unless otherwise specified.

Semantics of DC Terms

Let $\mathcal{I}$ be a given interpretation. The semantics of a DC term θ (under $\mathcal{I}$) is a function

$$\mathcal{I}[\![\theta]\!] : \mathsf{Val} \times \mathsf{Intv} \to \mathbb{R}$$

where $\mathsf{Intv} = \{[b, e] \mid b, e \in \mathsf{Time} \wedge b \leq e\}$ is the set of closed time intervals. $\mathcal{I}[\![\theta]\!]$ is defined inductively on the structure of θ.

$$\begin{array}{rcl}
\mathcal{I}[\![x]\!](\mathcal{V}, [b, e]) & = & \mathcal{V}(x), \\
\mathcal{I}[\![\ell]\!](\mathcal{V}, [b, e]) & = & e - b, \\
\mathcal{I}[\![\int P]\!](\mathcal{V}, [b, e]) & = & \int_b^e P_{\mathcal{I}}(t)dt, \\
\mathcal{I}[\![f(\theta_1, \ldots, \theta_n)]\!](\mathcal{V}, [b, e]) & = & \\
\multicolumn{3}{r}{\hat{f}(\mathcal{I}[\![\theta_1]\!](\mathcal{V}, [b, e]), \ldots, \mathcal{I}[\![\theta_n]\!](\mathcal{V}, [b, e])).}
\end{array}$$

Note that $P_{\mathcal{I}}$ should be Riemann integrable.

Example 6

$$\begin{aligned}
\mathcal{I}[\![\textstyle\int L]\!](\mathcal{V},[2,6]) &= \int_2^6 L_{\mathcal{I}}(t)dt \\
&= \int_2^3 L_{\mathcal{I}}(t)dt + \int_3^5 L_{\mathcal{I}}(t)dt + \int_5^6 L_{\mathcal{I}}(t)dt \\
&= \int_2^3 0dt + \int_3^5 1dt + \int_5^6 0dt \\
&= 0 + (5-3) + 0 \\
&= 2.
\end{aligned}$$

Finite Variability

We assume the following condition:

> *For each state variable X and each interval $[b, e]$ there is a finite partition of $[b, e]$ such that the interpretation $X_{\mathcal{I}}$ is constant on each part. Thus on each interval $[b, e]$ the function $X_{\mathcal{I}}$ has only finitely many points of discontinuity.*

In other words, $X_{\mathcal{I}}$ is a step (staircase) function. If this condition holds, any $P_{\mathcal{I}}$ is Riemann integrable.

Semantics of DC Formulas

Let $\mathcal{I}$ be a given interpretation. The semantics of a DC formula F (under $\mathcal{I}$) is a function

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

where tt and ff are truth values. $\mathcal{I}[\![F]\!]$ is defined inductively on the structure of F.

$$\begin{array}{rcl}
\multicolumn{3}{l}{\mathcal{I}[\![p(\theta_1, \ldots, \theta_n)]\!](\mathcal{V}, [b, e]) =} \\
\multicolumn{3}{r}{\hat{p}(\mathcal{I}[\![\theta_1]\!](\mathcal{V}, [b, e]), \ldots, \mathcal{I}[\![\theta_n]\!](\mathcal{V}, [b, e])),} \\
\mathcal{I}[\![\neg F]\!](\mathcal{V}, [b, e]) & = & \neg \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]), \\
\mathcal{I}[\![F \wedge G]\!](\mathcal{V}, [b, e]) & = & \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]) \wedge \mathcal{I}[\![G]\!](\mathcal{V}, [b, e]), \\
\mathcal{I}[\![\forall x.F]\!](\mathcal{V}, [b, e]) & = & \forall d \in \mathbb{R}.\mathcal{I}[\![F]\!](\mathcal{V}[x := d], [b, e]), \\
\mathcal{I}[\![F ; G]\!](\mathcal{V}, [b, e]) & = & \\
\multicolumn{3}{r}{\exists m \in [b, e].\mathcal{I}[\![F]\!](\mathcal{V}, [b, m]) \wedge \mathcal{I}[\![G]\!](\mathcal{V}, [m, e]).}
\end{array}$$

Semantics of DC Formulas

Note

- $\mathcal{V}[x := d]$ is a valuation that satisfies

$$\mathcal{V}[x := d](y) = \begin{cases} d & (x = y), \\ \mathcal{V}(y) & (\text{otherwise}). \end{cases}$$

- The logical symbols $(\neg, \wedge, \forall, \exists)$ in RHSs are for the underlying mathematical logic used to describe the semantics.

Semantics of Some Derived Formulas

$$
\begin{array}{rcl}
\mathcal{I}[\![\Diamond F]\!](\mathcal{V},[b,e]) & = & \exists[m_1,m_2]\subseteq[b,e].\mathcal{I}[\![F]\!](\mathcal{V},[m_1,m_2]) \\
\mathcal{I}[\![\Box F]\!](\mathcal{V},[b,e]) & = & \forall[m_1,m_2]\subseteq[b,e].\mathcal{I}[\![F]\!](\mathcal{V},[m_1,m_2]) \\
\mathcal{I}[\![\lceil\rceil]\!](\mathcal{V},[b,e]) & = & b=e \\
\mathcal{I}[\![\lceil P\rceil]\!](\mathcal{V},[b,e]) & = & \int_b^e P_{\mathcal{I}}(t)dt = e-b \wedge b<e
\end{array}
$$

The last equation means that $P_{\mathcal{I}}(t)=1$ holds for t almost everywhere in $[b,e]$.

Rigid and Chop-Free

- A term is called *rigid* if it does not contain the length and integral operators.
- A formula is called *rigid* if it only contains rigid terms.
- A formula is called *chop-free* if it does not contain the chop operator (;).

Satisfiability, Realisability, Validity

$$\mathcal{I}, \mathcal{V}, [b, e] \models F \overset{\text{def}}{\iff} \mathcal{I}[\![F]\!](\mathcal{V}, [b, e]) = \mathrm{tt}$$

$$\mathcal{I}, \mathcal{V} \models F \overset{\text{def}}{\iff} \forall [b, e] \in \mathrm{Intv}.\ \mathcal{I}, \mathcal{V}, [b, e] \models F$$

$$\mathcal{I} \models F \overset{\text{def}}{\iff} \forall \mathcal{V} \in \mathrm{Val}.\ \mathcal{I}, \mathcal{V} \models F$$

$$\models F \overset{\text{def}}{\iff} \forall \mathcal{I} \in \mathrm{Interp}.\ \mathcal{I} \models F$$

- F is *satisfiable* iff $\mathcal{I}, \mathcal{V}, [b, e] \models F$ for some $\mathcal{I}$, $\mathcal{V}$ and $[b, e]$.
- F is *realisable* iff $\mathcal{I}, \mathcal{V} \models F$ for some $\mathcal{I}$ and $\mathcal{V}$.
- F is *valid* iff $\models F$.

Satisfiability, Realisability, Validity

Duality

- F is satisfiable iff $\neg F$ is not valid.
- F is valid iff $\neg F$ is not satisfiable.

Other Properties

- F is realizable if F is valid, but not vice versa.
- F is satisfialble if F is realizable, but not vice versa.

Gas Burner

Requirement and Design Constraints

$$\mathrm{Req} \overset{\text{def}}{\iff} \Box(\ell \geq 60 \Rightarrow \int L \leq \ell/20)$$

$$\text{Des-1} \overset{\text{def}}{\iff} \Box(\lceil L \rceil \Rightarrow \ell \leq 1)$$

$$\text{Des-2} \overset{\text{def}}{\iff} \Box(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil \Rightarrow \ell > 30)$$

Theorem 2.16

$$\models \text{Des-1} \land \text{Des-2} \Rightarrow \mathrm{Req}$$

Gas Burner

Requirement

$$\text{Req-1} \overset{\text{def}}{\Longleftrightarrow} \Box(\ell \leq 30 \Rightarrow \textstyle\int L \leq 1)$$

Lemma 2.17

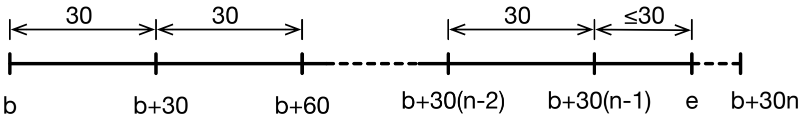
$$\models \text{Req-1} \Rightarrow \text{Req}$$

Lemma 2.19

$$\models \text{Des-1} \land \text{Des-2} \Rightarrow \text{Req-1}$$

Proof of Lemma 2.17

Consider an interval $[b, e]$ of length $\ell = e - b \geq 60$ and let $n = \lceil (e-b)/30 \rceil$ so that $n - 1 < (e-b)/30 \leq n$. We split $[b, e]$ into n adjacent subintervals.



We will show that

$$20 \int_b^e L_{\mathcal{I}}(t)dt \leq \ell.$$

Proof of Lemma 2.17 (cont'd)

$$
\begin{aligned}
& 20 \int_b^e L_{\mathcal{I}}(t)dt \\
&= 20 \left(\sum_{i=0}^{n-2} \int_{b+30i}^{b+30(i+1)} L_{\mathcal{I}}(t)dt + \int_{b+30(n-1)}^{e} L_{\mathcal{I}}(t)dt \right) \\
&\leq 20 \left(\sum_{i=0}^{n-2} 1 + 1 \right) && (\because \text{Req-1}) \\
&= 20n \\
&< 20 \left(\frac{e-b}{30} + 1 \right) && (\because n-1 < \frac{e-b}{30}) \\
&= \frac{2}{3}(e-b) + 20 \\
&\leq e-b && (\because e-b \geq 60) \\
&= \ell
\end{aligned}
$$

Proof of Lemma 2.19

$$
\begin{array}{ll}
 & \ell \leq 30 \\
\Rightarrow & \ell \leq 30 \wedge (\neg\Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil) \vee \Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil)) \\
\Rightarrow & \neg\Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil) \vee (\ell \leq 30 \wedge \Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil)) \\
\Rightarrow & \lceil\rceil \vee \lceil L \rceil ; (\lceil\rceil \vee \lceil \neg L \rceil) \vee \lceil \neg L \rceil ; (\lceil\rceil \vee \lceil L \rceil) \vee \lceil \neg L \rceil ; \lceil L \rceil ; \lceil \neg L \rceil \vee \\
 & (\ell \leq 30 \wedge \Diamond(\lceil L \rceil ; \lceil \neg L \rceil ; \lceil L \rceil)) \\
\Rightarrow & \lceil\rceil \vee \lceil L \rceil ; (\lceil\rceil \vee \lceil \neg L \rceil) \vee \lceil \neg L \rceil ; (\lceil\rceil \vee \lceil L \rceil) \vee \lceil \neg L \rceil ; \lceil L \rceil ; \lceil \neg L \rceil \\
\Rightarrow & \lceil\rceil \vee (\int L \leq 1) ; (\lceil\rceil \vee \lceil \neg L \rceil) \vee \lceil \neg L \rceil ; (\lceil\rceil \vee (\int L \leq 1)) \vee \\
 & \lceil \neg L \rceil ; (\int L \leq 1) ; \lceil \neg L \rceil \\
\Rightarrow & (\int L = 0) \vee (\int L \leq 1) ; (\int L = 0) \vee \\
 & (\int L = 0) ; ((\int L = 0) \vee (\int L \leq 1)) \vee \\
 & (\int L = 0) ; (\int L \leq 1) ; (\int L = 0) \\
\Rightarrow & \int L \leq 1
\end{array}
$$

Duration Calculus

Proof Rules

Proof Rules and Axioms

Proof Rules

$$\frac{F_1 \quad F_2 \quad \cdots \quad F_n}{G}$$

G can be *derived* from $F_1, F_2, \ldots, F_n$. $F_1, F_2, \ldots, F_n$ and G are called *premises* and *conclusion* of the rule.

Axioms

$$\frac{\qquad\qquad}{G}$$

Proof rules with empty premises. We write G to denote an axmiom.

Ex: Propositional Logic (Hilbert Style)

Syntax

$$\phi ::= p \mid \neg\phi \mid \phi \Rightarrow \phi$$

Axioms

$$\phi \Rightarrow (\psi \Rightarrow \phi) \tag{A1}$$

$$(\phi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\phi \Rightarrow \chi)) \tag{A2}$$

$$(\neg\phi \Rightarrow \neg\psi) \Rightarrow (\psi \Rightarrow \phi) \tag{A3}$$

Inference Rules

$$\frac{\phi \quad \phi \Rightarrow \psi}{\psi} \tag{MP}$$

Ex: Theorem

$$\phi \Rightarrow \phi$$

Proof

$1 : \phi \Rightarrow ((\phi \Rightarrow \phi) \Rightarrow \phi)$ (A1)

$2 : (\phi \Rightarrow ((\phi \Rightarrow \phi) \Rightarrow \phi)) \Rightarrow (\phi \Rightarrow (\phi \Rightarrow \phi) \Rightarrow (\phi \Rightarrow \phi)$ (A2)

$3 : (\phi \Rightarrow (\phi \Rightarrow \phi)) \Rightarrow (\phi \Rightarrow \phi)$ (1,2,MP)

$4 : \phi \Rightarrow (\phi \Rightarrow \phi)$ (A1)

$5 : \phi \Rightarrow \phi$ (4,3,MP)

Ex: Derived Rule

$$\frac{\phi \Rightarrow \psi \quad \psi \Rightarrow \chi}{\phi \Rightarrow \chi}$$

Proof

$$
\begin{array}{lr}
1 : \psi \Rightarrow \chi & \text{(premise)} \\
2 : (\phi \Rightarrow \chi) \Rightarrow (\phi \Rightarrow (\psi \Rightarrow \chi)) & \text{(A1)} \\
3 : \phi \Rightarrow (\psi \Rightarrow \chi) & \text{(1,2,MP)} \\
4 : (\phi \Rightarrow (\psi \Rightarrow \chi)) \Rightarrow ((\phi \Rightarrow \psi) \Rightarrow (\psi \Rightarrow \chi)) & \text{(A2)} \\
5 : (\phi \Rightarrow \psi) \implies (\phi \Rightarrow \chi) & \text{(3,4,MP)} \\
6 : \phi \Rightarrow \psi & \text{(premise)} \\
7 : \phi \Rightarrow \chi & \text{(6,5,MP)}
\end{array}
$$

Predicate Calculus

- Modus Ponens:
$$\frac{F \quad F \Rightarrow G}{G}.$$
- $\forall$-Introduction:
$$\frac{F}{\forall x.F}.$$
- $\forall$-Elimination:
$$\frac{\forall x.F}{F[x := \theta]}$$

  where F is chop-free or θ is rigid.

Equality

- Reflexivity:
$$x = x.$$
- Symmetry:
$$x = y \Rightarrow y = x.$$
- Transitivity:
$$(x = y \wedge y = z) \Rightarrow x = z.$$
- Leibniz Property:
$$(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \Rightarrow f(x_1, \ldots, x_n) = f(y_1, \ldots, y_n),$$
$$(x_1 = y_1 \wedge \cdots \wedge x_n = y_n) \Rightarrow p(x_1, \ldots, x_n) = p(y_1, \ldots, y_n).$$

Interval Logic (1)

- Length-Pos:
$$\ell \geq 0.$$
- Chop-Asm:
$$((F; G); H) \Leftrightarrow (F; (G; H)).$$
- Chop-Overlay:
$$((F; G_1) \wedge \neg(F; G_2)) \Rightarrow (F; (G_1 \wedge \neg G_2)),$$
$$((G_1; F) \wedge \neg(G_2; F)) \Rightarrow ((G_1 \wedge \neg G_2); F).$$
- Chop-Elim:
$$(F; G) \Rightarrow F,$$
$$(G; F) \Rightarrow F$$

where F is a rigid formula.

Interval Logic (2)

- Chop-Ex:
$$((\exists x.F); G) \Rightarrow \exists x.(F; G),$$
$$(G; (\exists x.F)) \Rightarrow \exists x.(G; F)$$

  where $x \notin \mathit{free}(G)$.
- Chop-Length:
$$(F; (\ell = x)) \Rightarrow \neg((\neg F); (\ell = x)),$$
$$((\ell = x); F) \Rightarrow \neg((\ell = x); (\neg F)).$$
- Add-Length:
$$(x \geq 0 \wedge y \geq 0) \Rightarrow ((\ell = x + y) \Leftrightarrow (\ell = x); (\ell = y))$$

Free Variables

$free(\theta)/free(F)$: the set of free (global) variables in θ/F.

$$
\begin{aligned}
free(x) &= \{x\} \\
free(\ell) &= \emptyset \\
free(\int P) &= \emptyset \\
free(f(\theta_1,\ldots,\theta_n)) &= \textstyle\bigcup_{i=1}^{n} free(\theta_i) \\
free(p(\theta_1,\ldots,\theta_n)) &= \textstyle\bigcup_{i=1}^{n} free(\theta_i) \\
free(\neg F) &= free(F) \\
free(F \wedge G) &= free(F) \cup free(G) \\
free(\forall x.F) &= free(F) \setminus \{x\} \\
free(F; G) &= free(F) \cup free(G)
\end{aligned}
$$

Interval Logic (3)

- Chop-Pnt:

$$F \Rightarrow (F; (\ell = 0)),$$
$$F \Rightarrow ((\ell = 0); F).$$

- Neccessary:

$$\frac{F}{\neg((\neg F); G)}, \qquad \frac{F}{\neg(G; (\neg F))}.$$

- Chop-Mon:

$$\frac{F \Rightarrow G}{(F; H) \Rightarrow (G; H)}, \qquad \frac{F \Rightarrow G}{(H; F) \Rightarrow (H; G)}.$$

Theorems

- Box-Impl:
$$\Box(F \Rightarrow G) \Rightarrow (\Box F \Rightarrow \Box G).$$
- Box-Elim:
$$\Box F \Rightarrow F.$$
- Box-Trans:
$$\Box F \Rightarrow \Box\Box F.$$
- Box-Intro:
$$\frac{F}{\Box F}.$$

Durations

- Dur-Zero:
$$\int 0 = 0.$$
- Dur-One:
$$\int 1 = \ell.$$
- Dur-Pos:
$$\int P \geq 0.$$
- Dur-Add:
$$\int P + \int Q = \int P \wedge Q + \int P \vee Q.$$
- Dur-Chop:
$$(\int P = x);(\int P = y) \Rightarrow \int P = x + y.$$
- Dur-Logic:
$$\int P = \int Q$$
where $P \Leftrightarrow Q$ is a tautology.

Theorems

- P-Mon:
$$\lceil P \rceil \Rightarrow \lceil Q \rceil$$
where $P \Rightarrow Q$ is a tautology.
- P-Chop:
$$\lceil P \rceil ; \lceil P \rceil \Leftrightarrow \lceil P \rceil .$$
- P-Box:
$$\lceil P \rceil \Rightarrow \Box(\lceil\rceil \vee \lceil P \rceil).$$
- P-Neg:
$$\neg \lceil P \rceil \Leftrightarrow (\lceil\rceil \vee \Diamond \lceil \neg P \rceil)$$
- P-And:
$$\lceil P \wedge Q \rceil \Leftrightarrow \lceil P \rceil \wedge \lceil Q \rceil$$

Theorems

- P-Chop-Neg:
$$\neg(\lceil P \rceil ; \text{true}) \Leftrightarrow \lceil\rceil \vee \lceil \neg P \rceil ; \text{true}$$
$$\neg(\text{true} ; \lceil P \rceil) \Leftrightarrow \lceil\rceil \vee \text{true} ; \lceil \neg P \rceil$$
- P-Chop-And:
$$((\lceil P \rceil ; \text{true}) \wedge (\lceil Q \rceil ; \text{true})) \Leftrightarrow \lceil P \wedge Q \rceil ; \text{true}$$
$$((\text{true} ; \lceil P \rceil) \wedge (\text{true} ; \lceil Q \rceil)) \Leftrightarrow \text{true} ; \lceil P \wedge Q \rceil$$
- P-Chop-Or:
$$((\lceil P \rceil ; \text{true}) \vee (\lceil Q \rceil ; \text{true})) \Leftrightarrow \lceil P \vee Q \rceil ; \text{true}$$
$$((\text{true} ; \lceil P \rceil) \vee (\text{true} ; \lceil Q \rceil)) \Leftrightarrow \text{true} ; \lceil P \vee Q \rceil$$

Induction Rules

- Induction-R:

$$\begin{array}{lrcl} (1) & \lceil\rceil & \Rightarrow & F \\ (2) & F;\lceil P\rceil & \Rightarrow & F \\ (3) & F;\lceil\neg P\rceil & \Rightarrow & F \\ \hline (4) & & F & \end{array}$$

- Induction-L:

$$\begin{array}{lrcl} (1) & \lceil\rceil & \Rightarrow & F \\ (2) & \lceil P\rceil;F & \Rightarrow & F \\ (3) & \lceil\neg P\rceil;F & \Rightarrow & F \\ \hline (4) & & F & \end{array}$$

Example

F holds for any P.

$$F \overset{\text{def}}{\iff} \lceil\rceil \vee (\text{true}; \lceil P \rceil) \vee (\text{true}; \lceil \neg P \rceil)$$

Proof

$$
\begin{array}{lr}
1 : \lceil\rceil \Rightarrow F & \text{(PL)} \\
2 : F \Rightarrow \text{true} & \text{(PL)} \\
3 : F; \lceil P \rceil \Rightarrow \text{true}; \lceil P \rceil & \text{(2,Chop-Mon)} \\
4 : \text{true}; \lceil P \rceil \Rightarrow F & \text{(PL)} \\
5 : F; \lceil P \rceil \Rightarrow F & \text{(3,4,MP)} \\
6 : F; \lceil \neg P \rceil \Rightarrow \text{true}; \lceil \neg P \rceil & \text{(2,Chop-Mon)} \\
7 : \text{true}; \lceil \neg P \rceil \Rightarrow F & \text{(PL)} \\
8 : F; \lceil \neg P \rceil \Rightarrow F & \text{(6,7,MP)} \\
9 : F & \text{(1,5,8,Induction-R)}
\end{array}
$$

Soundness

Soundness of Axioms and Proof Rules

For any interpretation $\mathcal{I}$, $(\mathcal{I} \models F_1) \wedge \cdots \wedge (\mathcal{I} \models F_n)$ implies $\mathcal{I} \models G$ if

$$\frac{F_1 \quad \cdots \quad F_n}{G}$$

is an inference rule of DC.

Soundness of Induction Rules

For all interpretation $\mathcal{I}$,

- $\mathcal{I} \models \lceil\rceil \Rightarrow F$,
- $\mathcal{I} \models F; \lceil P \rceil \Rightarrow F$, and
- $\mathcal{I} \models F; \lceil \neg P \rceil \Rightarrow F$

always imply $\mathcal{I} \models F$.

Summary

- Syntax
- Semantics
- Proof Rules