Content

Introduction
- Observables and Evolutions
- Duration Calculus (DC)
- Semantical Correctness Proofs
- DC Decidability
- DC Implementables
- PLC-Automata

- Timed Automata (TA), Uppaal
- Networks of Timed Automata
- Region/Zone-Abstraction
- TA model-checking
- Extended Timed Automata
- Undecidability Results

\[
\begin{align*}
\text{obs} : \text{Time} &\rightarrow \mathcal{P}(\text{obs}) \\
\langle \text{obs}_0, \nu_0 \rangle, t_0 &\xrightarrow{\lambda_0} \langle \text{obs}_1, \nu_1 \rangle, t_1 \ldots
\end{align*}
\]

- Automatic Verification...
  ...whether a TA satisfies a DC formula, observer-based

- Recent Results:
  - Timed Sequence Diagrams, or Quasi-equal Clocks,
    or Automatic Code Generation, or ...
Duration Calculus: Preview

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

Strangest operators:

- almost everywhere – Example: $[G]$
  (Holds in a given interval $[b, e]$ iff the gas valve is open almost everywhere.)
- chop – Example: $([\neg I] ; [I] ; [\neg I]) \implies \ell \geq 1$
  (Ignition phases last at least one time unit.)
- integral – Example: $\ell \geq 60 \implies \int L \leq \frac{\ell}{60}$
  (At most 5% leakage time within intervals of at least 60 time units.)

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

Content

- Symbols
  - predicate and function symbols
  - state variables and domain values
  - global (or logical) variables
- State Assertions
  - syntax
  - semantics
- Terms
  - syntax
  - rigid terms
  - intervals
  - semantics
  - remarks
Duration Calculus: Syntax Overview

We will introduce four syntactical categories (and abbreviations):

(i) Symbols:

\[
\begin{align*}
\text{true, false, =, <, >, \leq, \geq, f, g, X, Y, Z, d, x, y, z,} \\
p, q
\end{align*}
\]

(ii) State Assertions:

\[
P ::= 0 | 1 | X = d | \neg P_1 | P_1 \land P_2
\]

(iii) Terms:

\[
\theta ::= x | \ell | \int P | f(\theta_1, \ldots, \theta_n)
\]

(iv) Formulae:

\[
F ::= p(\theta_1, \ldots, \theta_n) | \neg F_1 | F_1 \land F_2 | \forall x \bullet F_1 | F_1 ; F_2
\]

(v) Abbreviations:

\[
[\cdot], [P], [P]^t, [P]^{\leq t}, \Diamond F, \Box F
\]
Symbols: Predicate Symbols

- We assume a set of predicate symbols to be given, typical elements $p, q$.
  - Each predicate symbol $p$ has an arity $n \in \mathbb{N}_0$; shorthand notation: $p/n$.
  - A predicate symbol $p/n$ is called a constant if and only if $n = 0$.

- In the following, we assume the following predicate symbols:
  - constants: true, false.
  - binary (i.e. $n = 2$): $=$, $<$, $\leq$, $\geq$.

- Semantical domains: truth values $\mathbb{B} = \{\text{tt, ff}\}$, and real numbers $\mathbb{R}$.

- The semantics of an $n$-ary predicate symbol $p$ is a function from $\mathbb{R}^n$ to $\mathbb{B}$, denoted $\hat{p}$, i.e. $\hat{p} : \mathbb{R}^n \rightarrow \mathbb{B}$.

- For constants (arity $n = 0$) we have $\hat{p} \in \mathbb{B}$.

- Examples:
  - $\hat{\text{true}} = \text{tt}$, $\hat{\text{false}} = \text{ff}$.
  - $\hat{=} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B}$, $\hat{=}(a, b) = \text{tt}$, iff $a = b$.
    - $\hat{=}(3, 17) = \text{ff}$, $\hat{=}(2, 2) = \text{tt}$.
Once Again: Syntax vs. Semantics

- **Predicate symbols** are principally freely chosen, we could also consider the following ones:
  - ⊗/1
  - ✧/3
  - ≥/2

- To semantically work with a **predicate symbol**, we need to define a meaning.
  One possible choice:
  - \( \hat{\otimes} : \mathbb{R} \rightarrow \mathbb{B} \)
    \[ \hat{\otimes}(a) = \begin{cases} 
    \text{tt}, & \text{if } a \in \mathbb{N} \text{ and digit sum of } a \text{ equals 27} \\
    \text{ff}, & \text{otherwise} 
    \end{cases} \]
  - \( \hat{\bowtie} : \mathbb{R} \times \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B} \)
    \[ \hat{\bowtie}(a, b, c) = \begin{cases} 
    \text{tt}, & \text{if } ax^2 + bx + c = 0 \text{ has at least one solution} \\
    \text{ff}, & \text{otherwise} 
    \end{cases} \]
  - \( \hat{\geq} : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{B} \)
    \[ \hat{\geq}(a, b) = \begin{cases} 
    \text{tt}, & \text{if } a \geq b \\
    \text{ff}, & \text{otherwise} 
    \end{cases} \]

Same Game: Function Symbols

- **true, false, =, <, >, ≤, ≥, f, g, X, Y, Z, d, x, y, z,**

- We assume a set of **function symbols** to be given, typical elements \( f, g \).
  - Each **function symbol** \( f \) has an **arity** \( n \in \mathbb{N}_0 \); shorthand notation: \( f/n \).
  - A function symbol \( f/n \) is called a **constant** if and only if \( n = 0 \).

- In the following, we assume the following **function symbols**:  
  - **constants**: \( i/0 \) for each \( i \in \mathbb{N}_0 \).
  - **binary** (i.e. \( n = 2 \)): \( +, \cdot \).

- The semantics of an \( n \)-ary **function symbol** \( f \)
  is a function from \( \mathbb{R}^n \) to \( \mathbb{R} \), denoted \( \hat{f} \), i.e. \( \hat{f} : \mathbb{R}^n \rightarrow \mathbb{R} \).
  - For constants (arity \( n = 0 \)) we have \( \hat{f} \in \mathbb{R} \).

- **Examples**:
  - \( 0 = 0 \in \mathbb{R}, 27 = 27 \in \mathbb{R} \).
  - \( + : \mathbb{R} \times \mathbb{R} \rightarrow \mathbb{R} \), \( + (a, b) = a + b \),
    \( + (1, 2) = 3 \).
To better distinguish syntax from semantics, we could choose to work with the following symbols for natural numbers:

- **Syntax:**
  - zero, one, two, ..., twentyseven, ...
  (all with arity 0)

- **Semantics:**
  - \( \hat{\text{zero}} = 0 \in \mathbb{R} \),
  - \( \hat{\text{one}} = 1 \in \mathbb{R} \),
  - \( \hat{\text{two}} = 2 \in \mathbb{R} \),
  - \( \vdots \),
  - \( \hat{\text{twentyseven}} = 27 \in \mathbb{R} \),
  - \( \vdots \)
Symbols: State Variables and Domain Values

true, false, =, <, >, ≤, ≥, f, g, X, Y, Z, d, x, y, z

- We assume a set ‘Obs’ of state variables or observables, typical elements X, Y, Z.
- Each state variable X has a finite (semantical) domain D(X) = {d_1, ..., d_n}.
- A state variable with domain {0, 1} is called boolean observable.

- For each domain {d_1, ..., d_n} of a state variable in ‘Obs’ we assume
  - symbols d_1, ..., d_n
  - with d_i = d_i, 1 ≤ i ≤ n.

- Example:
  - state variable F ("flame sensor"). domain D(F) = {0, 1}.
    symbols 0, 1 with \( \hat{0} = 0 \in \mathbb{N}_0 \), \( \hat{1} = 1 \in \mathbb{N}_0 \).
  - state variable T ("traffic lights"). domain D(T) = {red, green},
    symbols red, green with with \( \hat{\text{red}} = \text{red} \in D(T) \), \( \hat{\text{green}} = \text{green} \in D(T) \).

Interpretation of State Variables

- The last semantical domain we consider is
  - the set Time of points in time,
  - mostly, Time = \( \mathbb{R}_0^+ \) (continuous / dense),
  - sometimes Time = \( \mathbb{N}_0 \) (discrete time).

- The semantics of a state variable is time-dependent.

It is given by an interpretation \( I \), i.e. a mapping

\[
I : \text{Obs} \rightarrow (\text{Time} \rightarrow D), \quad D = \bigcup_{X \in \text{Obs}} D(X),
\]

assigning to each state variable \( X \in \text{Obs} \) a function

\[
I(X) : \text{Time} \rightarrow D(X)
\]

such that \( I(X)(t) \in D(X) \) denotes the value that \( X \) has at time \( t \in \text{Time} \).

- For convenience, we shall abbreviate \( I(X) \) to \( X_I \).
Let $\text{Obs} = \{obs_1, \ldots, obs_n\}$ be a set of state variables.

**Evolution** (over time) of Obs:

$$\pi : \text{Time} \rightarrow \mathcal{D}(\text{obs}_1) \times \cdots \times \mathcal{D}(\text{obs}_n).$$

**Interpretation** of Obs:

$$I : \text{Obs} \rightarrow (\text{Time} \rightarrow \mathcal{D}).$$

Both, $\pi$ and $I$, represent the same timed behaviour if,

- for all $t \in \text{Time},$
  - $I(\text{obs}_i)(t) = \pi(t) \downarrow i, \quad 1 \leq i \leq n,$ or
  - $\pi(t) = (I(\text{obs}_1)(t), \ldots, I(\text{obs}_n)(t)) = (\text{obs}_{12}(t), \ldots, \text{obs}_{n2}(t)).$

---

**Example: Evolutions vs. Interpretation of State Variables**

- $\text{obs}_1 = H, \text{obs}_2 = G, \text{obs}_3 = I, \text{obs}_4 = F$
- $\pi(t) = (1, 1, 0, 1), \quad I(H)(t) = H(t) = \pi(t) \downarrow 1 = 1,$
  $I(I)(t) = I_2(t) = \pi(t) \downarrow 3 = 0.$
- $I : \text{Obs} \rightarrow (\text{Time} \rightarrow \mathcal{D})$: 
  $$\text{Time} \rightarrow \mathcal{D}$$
Predicate / Function Symbols vs. State Variables

\[ true, false, =, <, >, \leq, \geq, \ f, g, \ X, Y, Z, \ d, \ x, y, z, \]

**Note:**
- The choice of function and predicate symbols introduced earlier, i.e.
  - \( true, false, =, <, >, \leq, \geq, \)
  - \( 0, 1, \ldots, \)
  - \( +, \cdot, \)
and their semantics, i.e.
- \( true \) is the truth value \( \mathbf{t} \in \mathbf{B}, \)
- \( \equiv : \mathbb{R}^2 \rightarrow \mathbf{B} \) is the equality relation on real numbers,
- \( \hat{0} \) is the (real) number zero from \( \mathbb{R}, \)
- \( + : \mathbb{R}^2 \rightarrow \mathbb{R} \) is the addition function on real numbers,
is fixed throughout the lecture.

- The choice of observables and their domains depends on the system we want to describe.

**Symbols: Global Variables**

\[ true, false, =, <, >, \leq, \geq, \ f, g, \ X, Y, Z, \ d, \ x, y, z, \]

- We assume a set ‘GVar’ of global (or logical) variables, typical elements \( x, y, z. \)
- The semantics of a global variable is given by a valuation, i.e. a mapping
  \[ V : \text{GVar} \rightarrow \mathbb{R} \]
assigning to each global variable \( x \in \text{GVar} \) a real number \( V(x) \in \mathbb{R}. \)
We use Val to denote the set of all valuations, i.e. \( \text{Val} = (\text{GVar} \rightarrow \mathbb{R}). \)
Global variables are fixed over time in system evolutions.
\[
\begin{align*}
G\mathcal{V}_{\mathcal{W}} &= \{ x, y \} \\
\mathcal{V}_1 &= \{ x \mapsto 0, \ y \mapsto 7 \} \\
\mathcal{V}_2 &= \{ x \mapsto 3.14, \ y \mapsto 2.7 \}
\end{align*}
\]
Symbols: Overview

<table>
<thead>
<tr>
<th>Syntax</th>
<th>Semantics (meaning)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>predicate symbols</strong></td>
<td></td>
</tr>
<tr>
<td>true, false, =, &lt;, &gt;, ≤, ≥</td>
<td>true = ( \text{tt} \in \mathbb{B} ), ( \equiv : \mathbb{R}^2 \rightarrow \mathbb{B} )</td>
</tr>
<tr>
<td><strong>function symbols</strong></td>
<td></td>
</tr>
<tr>
<td>( f / n, g )</td>
<td>( \hat{f} : \mathbb{R}^n \rightarrow \mathbb{R} )</td>
</tr>
<tr>
<td><strong>state variables</strong></td>
<td></td>
</tr>
<tr>
<td>( X, Y, Z )</td>
<td>( \mathcal{I}(X) : \text{Time} \rightarrow \mathcal{D}(X) )</td>
</tr>
<tr>
<td><strong>domain values</strong></td>
<td></td>
</tr>
<tr>
<td>( d )</td>
<td>( \hat{d} \in \mathcal{D}(X) )</td>
</tr>
<tr>
<td><strong>global variables</strong></td>
<td></td>
</tr>
<tr>
<td>( x, y, z )</td>
<td>( \forall(x) \in \mathbb{R} )</td>
</tr>
</tbody>
</table>

Duration Calculus: State Assertions
We will introduce **four syntactical categories** (and **abbreviations**):

(i) **Symbols:**
\[
\begin{align*}
& p, q, \\
& \text{true, false, } =, <, \leq, \geq, \\
& f, g, \quad X, Y, Z, \quad d, \quad x, y, z,
\end{align*}
\]

(ii) **State Assertions:**
\[
P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \land P_2
\]

(iii) **Terms:**
\[
\theta ::= x \mid \ell \mid \int P \mid f(\theta_1, \ldots, \theta_n)
\]

(iv) **Formulae:**
\[
F ::= p(\theta_1, \ldots, \theta_n) \mid \neg F_1 \mid F_1 \land F_2 \mid \forall x \bullet F_1 \mid F_1 ; F_2
\]

(v) **Abbreviations:**
\[
\begin{align*}
& \lceil \rceil, \quad \lceil P \rceil, \quad \lceil P \rceil^t, \quad \lceil P \rceil^{\leq t}, \quad \Diamond F, \quad \Box F
\end{align*}
\]

---

**State Assertions: Syntax**

- The set of **state assertions** is defined by the following grammar:
\[
P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \land P_2
\]

where
- \(X \in \text{Obs}\) is a state variable.
- \(d\) denotes a value from \(X\)'s domain.

We shall use \(P, Q, R\) to denote state assertions.

- Here, '0', '1', '='; '¬', and '∧'
  are like **keywords** (or terminal symbols) in programming languages.

**Abbreviations:**

- We shall write \(X\) instead of \(\lceil X = 1 \rceil\) if \(X\) is **boolean**, i.e. if \(\mathcal{D}(X) = \{0, 1\}\).
- Assume the usual precedence: '¬' binds stronger than '∧'
- Define \(\lor, \implies, \iff\) as usual.
State Assertions: Examples

\[ P ::= 0 \mid 1 \mid X = d \mid \neg P_1 \mid P_1 \land P_2 \]

Observables \( F, G, D(F) = \{0, 1\}, D(G) = \{0, 1, 2\}. \)

- \( 0 \mid \odot \)
- \( F = 1 \mid \odot \)
- \( F \mid \odot \) abbrev.
- \( \neg (F = 1) \mid \odot, \odot \)
- \( \neg F \mid \odot \) abbrev.
- \( G \mid \times \)
- \( G = 2, \neg F = 2 \times \)
- \( F = G \mid \times \) typoing
- \( F = 1 \land G = 1 \mid \odot \)
- \( \neg (F = 1 \land G = 1) \mid \odot \)
- \( \neg (F = 1 \land G = 1) \) \( \neg F = 1 \land G = 1, \) \( (\neg F = 1) \land G = 1 \) \( \odot \)

State Assertions: Semantics

- The semantics of state assertion \( P \) is a function
  \[ I[P] : \text{Time} \to \{0, 1\}, \]
  i.e., \( I[P](t) \) denotes the truth value of \( P \) at time \( t \in \text{Time} \).

- The value \( I[P](t) \) is defined inductively over the structure of \( P \):
  \[
  \begin{align*}
  I[0](t) &= 0 \\
  I[1](t) &= 1 \\
  I[X = d](t) &= \begin{cases} 1, & \text{if } X_I(t) = d \\ 0, & \text{otherwise} \end{cases} \\
  I[\neg P_1](t) &= 1 - I[P_1](t) \\
  I[P_1 \land P_2](t) &= \begin{cases} 1, & \text{if } I[P_1](t) = 1, i \in \{i, 2\} \\ 0, & \text{otherwise} \end{cases}
  \end{align*}
  \]
State Assertions: Notes

• If $X$ is a boolean observer, the following equalities hold:

$$I[X](t) = I[X = 1](t) = I(X)(t) = X_I(t).$$

• $I[P]$ is also called interpretation of $P$.

We shall write $P_I$ as a shorthand notation.

• Here, the state assertions 0 and 1 are treated like boolean values (like tt and ff), it will become clear in a minute, why 0, 1 is a better choice than tt and ff.

State Assertions: Example

• Interpretation $I$ of boolean observables $G$ and $F$:

• Consider state assertion $L := G \land \neg F$.

• $L_I(1.2) = 1$, because

• $L_I(2) = 0$, because

• Interpretation of $L$ as timing diagram:
Duration Calculus: Terms

Duration Calculus: Overview

We will introduce four syntactical categories (and abbreviations):

(i) Symbols:
\[ p,q, true, false, =, <, >, \leq, \geq, f, g, X, Y, Z, d, x, y, z, \]

(ii) State Assertions:
\[ P ::= 0 | 1 | X = d | \neg P_1 | P_1 \land P_2 \]

(iii) Terms:
\[ \theta ::= x | \ell | \int P | f(\theta_1, \ldots, \theta_n) \]

(iv) Formulae:
\[ F ::= p(\theta_1, \ldots, \theta_n) | \neg F_1 | F_1 \land F_2 | \forall x \bullet F_1 | F_1 ; F_2 \]

(v) Abbreviations:
\[ [\_], [P], [P]^t, [P]^{\leq t}, \Diamond F, \Box F \]
• **Duration terms** (or DC terms, or just terms) are defined by the following grammar:

\[
\theta ::= x \mid \ell \mid P \mid f(\theta_1, \ldots, \theta_n)
\]

where

- \(x\) is a **global variable** from GVar,
- \(P\) is a **state assertion**, and
- \(f\) a **function symbol** (of arity \(n\)).

- ‘\(\ell\)’ and ‘\(f\)’ are like **keywords** (or terminal symbols) in programming languages.
- \(\ell\) is called **length operator**, \(f\) is called **integral operator**.

**Notation:** we may write function symbols in **infix notation** as usual, i.e. we may write \(\theta_1 + \theta_2\) instead of \(+(\theta_1; \theta_2)\).

**Definition 1.** [Rigid]

A term **without** length and integral operators is called **rigid**.
Towards Semantics of Terms: Intervals

- Let $b, e \in \text{Time}$ be points in time s.t. $b \leq e$. Then $[b, e]$ denotes the **closed interval** $\{x \in \text{Time} \mid b \leq x \leq e\}$.

- We use 'Intv' to denote the set of **closed intervals** in the time domain, i.e.
  \[
  \text{Intv} := \{[b, e] \mid b, e \in \text{Time}\}.
  \]

- **Closed intervals** of the form $[b, b]$ are called **point intervals**.

---

Terms: Semantics

- The **semantics** of a term $\theta$ is a function
  \[
  I[\theta] : \text{Val} \times \text{Intv} \to \mathbb{R},
  \]
  that is, $I[\theta]$ maps a pair consisting of a **valuation** and an **interval** to a real number.

- $I[\theta](V, [b, e])$ is called
  - the **value** (or interpretation) of $\theta$
  - under interpretation $I$ and valuation $V$
  - in the **interval** $[b, e]$.

- The value $I[\theta](V, [b, e])$ is defined **inductively** over the structure of $\theta$:
  \[
  \begin{align*}
  I[x](V, [b, e]) &= V(x) \\
  I[\ell](V, [b, e]) &= e - b \\
  I[\int P](V, [b, e]) &= \frac{1}{b - e} \int_{b}^{e} P(x) \, dx \\
  I[f(\theta_1, \ldots, \theta_n)](V, [b, e]) &= I[f](V, [b, e]) (I[\theta_1](V, [b, e]), \ldots, I[\theta_n](V, [b, e]))
  \end{align*}
  \]
Consider the term $\theta = x \cdot \int L$. 

- $I[\theta](\mathcal{V}, [0.5, 3.25]) = I[(x \cdot \int L)\mathcal{V}, [0.5, 3.25])$
- $= \begin{cases} 
  \mathcal{V}(x) & \text{if } x \in \mathbb{Q} \\
  \int_{0.5}^{3.25} L(t) \, dt & \text{if } x \not\in \mathbb{Q}
\end{cases} 

= \begin{cases} 
  20 & \text{if } x \in \mathbb{Q} \\
  \int_{0.5}^{3.25} L(t) \, dt & \text{if } x \not\in \mathbb{Q}
\end{cases} 

= 20 \cdot 1.25 = 25$

- $I[\theta](\mathcal{V}, [1.5, 1.5]) = \emptyset$

Terms: Is the Semantics Well-defined?

- So, $I[\int P](\mathcal{V}, [b, e]) = \int_{b}^{e} P_{X}(t) \, dt$ – but does the integral always exist?

- IOW: is there a $P_{X}$ which is not (Riemann-)integrable? Yes. For instance

$$P_{X}(t) = \begin{cases} 
  1 & \text{if } t \in \mathbb{Q} \\
  0 & \text{if } t \not\in \mathbb{Q}
\end{cases}$$

- To exclude such functions, DC considers only interpretations $\mathcal{I}$ satisfying the following condition of finite variability:

  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 a function $X_{\mathcal{I}}$ is of finite variability if and only if, on each interval $[b, e]$, the function $X_{\mathcal{I}}$ has only finitely many points of discontinuity.
Remark 2.5. The semantics $\mathcal{I}[\theta]$ of a term is insensitive against changes of the interpretation $\mathcal{I}$ at individual time points.

More formally:

- Let $\mathcal{I}_1, \mathcal{I}_2$ be interpretations of $\text{Obs}$ such that $\mathcal{I}_1(X)(t) = \mathcal{I}_2(X)(t)$ for all $X \in \text{Obs}$ and all $t \in \text{Time} \setminus \{t_0, \ldots, t_n\}$. Then $\mathcal{I}_1[\theta](V, [b, e]) = \mathcal{I}_2[\theta](V, [b, e])$ for all terms $\theta$ and intervals $[b, e]$.

Remark 2.6. The semantics $\mathcal{I}[\theta](V, [b, e])$ of a rigid term does not depend on the interval $[b, e]$.
Tell Them What You’ve Told Them...

- **State assertions** over
  - state variables (or observables), and
  - predicate symbols
  are evaluated at points in time.
  The semantics of a state assertion is a truth value.

- **Terms** are evaluated over intervals and can
  - measure the accumulated duration of a state assertion,
  - measure the length of intervals, and
  - use function symbols.
  The semantics of a term is a real number.

- The value of rigid terms is independent from the considered interval.
- The semantics of terms is insensitive against changes at finitely many points in time.