Content

- **RDC** $+\ell = x, \forall x$ in Continuous Time

  - Outline of the proof
  
  - Recall: two-counter machines (2-CM)
    
    - states and commands (syntax)
    
    - configurations and computations (semantics)

  - Encoding configurations in DC
    
    - initial configuration of a 2-CM

  - Encoding transitions in DC
    
    - increment counter,
    
    - decrement counter,
    
    - and some helper formulae.

  - Satisfiability and Validity

  - Discussion
Decidability Results for Realisability: Overview

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Discrete Time</th>
<th>Continuous Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDC</td>
<td><strong>decidable</strong></td>
<td><strong>decidable</strong></td>
</tr>
<tr>
<td>RDC + ( \ell = r )</td>
<td>decidable for ( r \in \mathbb{N} )</td>
<td>undecidable for ( r \in \mathbb{R}^+ )</td>
</tr>
<tr>
<td>RDC + ( \int P_1 = \int P_2 )</td>
<td>undecidable</td>
<td>undecidable</td>
</tr>
<tr>
<td>RDC + ( \ell = x, \forall x )</td>
<td>undecidable</td>
<td><strong>undecidable</strong></td>
</tr>
<tr>
<td>DC</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Decidability Results for RDC in Continuous Time
Recall: Restricted DC (RDC)

\[ F ::= [P] \mid \neg F_1 \mid F_1 \lor F_2 \mid F_1 ; F_2 \]

where \( P \) is a state assertion with **boolean** observables only.

From now on: “RDC + \( \ell = x, \forall x \)”

\[ F ::= [P] \mid \neg F_1 \mid F_1 \lor F_2 \mid F_1 ; F_2 \mid \ell = 1 \mid \ell = x \mid \forall x \bullet F_1 \]
Theorem 3.10.
The realisability from 0 problem for DC with continuous time is undecidable, not even semi-decidable.

Theorem 3.11.
The satisfiability problem for DC with continuous time is undecidable.
Sketch: Proof of Theorem 3.10

Reduce divergence of two-counter machines to realisability from 0:

- Given a two-counter machine $\mathcal{M}$ with final state $q_{\text{fin}}$,
- construct a DC formula $F(\mathcal{M}) := encoding(\mathcal{M})$
- such that

  $\mathcal{M}$ diverges if and only if the DC formula

  $$F(\mathcal{M}) \land \neg \diamondsuit [q_{\text{fin}}]$$

  is realisable from 0.

- If realisability from 0 was (semi-)decidable, divergence of two-counter machines would be (which it isn’t).
Two-Counter Machines
Recall: Two-counter machines

A **two-counter** machine is a structure

\[ M = (Q, q_0, q_{\text{fin}}, \text{Prog}) \]

where

- \( Q \) is a finite set of **states**,  
- comprising the **initial state** \( q_0 \) and the **final state** \( q_{\text{fin}} \)  
- \( \text{Prog} \) is the **machine program**, i.e. a finite set of **commands** of the form

\[
q : \text{inc}_i : q' \quad \text{and} \quad q : \text{dec}_i : q', q'', \quad i \in \{1, 2\}.
\]

\[
q : x_1 := x_1 + 1; \quad \text{goto } q' \quad \quad q : \text{if } (x_1 = 0) \\
q : x_2 := x_2 + 1; \quad \text{goto } q' \quad \quad \text{else} \quad \text{goto } q' \\
q : x_1 := x_1 - 1; \quad \text{goto } q''
\]

- We assume **deterministic** 2CM: for each \( q \in Q \), at most one command starts in \( q \),  
  and \( q_{\text{fin}} \) is the only state where no command starts.
• a configuration of \( \mathcal{M} \) is a triple \( K = (q, n_1, n_2) \in Q \times \mathbb{N}_0 \times \mathbb{N}_0 \).

• The transition relation \( \vdash \) on configurations is defined as follows:

<table>
<thead>
<tr>
<th>Command</th>
<th>Semantics: ( K \vdash K' )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( q : inc_1 : q' ) ( q : dec_1 : q', q'' )</td>
<td>( (q, n_1, n_2) \vdash (q', n_1 + 1, n_2) ) ( (q, 0, n_2) \vdash (q', 0, n_2) ) ( (q, n_1 + 1, n_2) \vdash (q'', n_1, n_2) )</td>
</tr>
<tr>
<td>( q : inc_2 : q' ) ( q : dec_2 : q', q'' )</td>
<td>( (q, n_1, n_2) \vdash (q', n_1, n_2 + 1) ) ( (q, n_1, 0) \vdash (q', n_1, 0) ) ( (q, n_1, n_2 + 1) \vdash (q'', n_1, n_2) )</td>
</tr>
</tbody>
</table>

• The (!) computation of \( \mathcal{M} \) is a finite sequence of the form \( \text{"\( \mathcal{M} \) halts"} \)

\[
K_0 = (q_0, 0, 0) \vdash K_1 \vdash K_2 \vdash \cdots \vdash (q_{\text{fin}}, n_1, n_2)
\]

or an infinite sequence of the form \( \text{"\( \mathcal{M} \) diverges"} \)

\[
K_0 = (q_0, 0, 0) \vdash K_1 \vdash K_2 \vdash \ldots
\]
2CM Example

- $M = (Q, q_0, q_{\text{fin}}, \text{Prog})$
- commands of the form $q : \text{inc}_i : q' \text{ and } q : \text{dec}_i : q', q''$, $i \in \{1, 2\}$
- configuration $K = (q, n_1, n_2) \in Q \times \mathbb{N}_0 \times \mathbb{N}_0$.

<table>
<thead>
<tr>
<th>Command</th>
<th>Semantics: $K \vdash K'$</th>
</tr>
</thead>
</table>
| $q : \text{inc}_1 : q'$ | $(q, n_1, n_2) \vdash (q', n_1 + 1, n_2)$  
| $q : \text{dec}_1 : q', q''$ | $(q, 0, n_2) \vdash (q', 0, n_2)$  
|                       | $(q, n_1 + 1, n_2) \vdash (q'', n_1, n_2)$               |
| $q : \text{inc}_2 : q'$ | $(q, n_1, n_2) \vdash (q', n_1, n_2 + 1)$  
| $q : \text{dec}_2 : q', q''$ | $(q, n_1, 0) \vdash (q', n_1, 0)$  
|                       | $(q, n_1, n_2 + 1) \vdash (q'', n_1, n_2)$               |

$M_1$
- $Q = \{q_0, q_1, q_{\text{fin}}\}$
- $\text{Prog} = \{q_0 : \text{inc}_1 : q_1, q_1 : \text{inc}_1 : q_{\text{fin}}\}$
- $\{q_0, 0, 0\} \overset{1}{\Rightarrow} \{q_0, 1, 0\} \overset{2}{\Rightarrow} \{q_{\text{fin}}, 2, 0\}$ (M1 halts)

$M_2$
- $Q = \{q_0, q_{\text{fin}}\}$
- $\text{Prog} = \{q_0 : \text{inc}_2 : q_0\}$
- $\{q_0, 0, 0\} \overset{4}{\Rightarrow} \{q_0, 0, 1\} \overset{6}{\Rightarrow} \{q_0, 0, 2\} \ldots$ (M2 diverges)
Reduction to 2-CM: Idea
Reducing Divergence to DC realisability: Idea In Pictures

2CM $\mathcal{M}$ diverges

iff

exists $\pi : K_0 \vdash K_1 \vdash \ldots$

iff

exists interpretation $\mathcal{I}$ of $\pi$

```
\[\mathcal{I} \vdash_0 F(\mathcal{M}) \land \neg \Diamond [q_{fin}]\]
```

$F(\mathcal{M})$ intuitively specifies:

- $[0, d]$ encodes $(q_0, 0, 0)$,
- each $[n \cdot d, (n + 1) \cdot d]$ encodes a configuration,
- $[n \cdot d, (n + 1) \cdot d]$ and $[(n + 1) \cdot d, (n + 2) \cdot d]$ are in $\vdash$-relation,
- if $q_{fin}$ is reached, we stay there
Reducing Divergence to DC realisability: Idea

- A single configuration $K$ of $\mathcal{M}$ can be encoded in an interval of length 4; being an encoding interval can be characterised by a DC formula.

- An interpretation on ‘Time’ encodes the computation of $\mathcal{M}$ if
  - each interval $[4n, 4(n + 1)], n \in \mathbb{N}_0$, encodes a configuration $K_n$,
  - each two subsequent intervals $[4n, 4(n + 1)]$ and $[4(n + 1), 4(n + 2)], n \in \mathbb{N}_0$, encode configurations $K_n \vdash K_{n+1}$ in transition relation.

- Being an encoding of the run can be characterised by a DC formula $F(\mathcal{M})$.

- Then $\mathcal{M}$ diverges if and only if $F(\mathcal{M}) \land \neg \Diamond [q_{\text{fin}}]$ is realisable from 0.
Encoding Configurations
Encoding Configurations

- We use $\text{Obs} = \{\text{obs}\}$ with $\mathcal{D}(\text{obs}) = \mathcal{Q}_M \cup \{C_1, C_2, B, X\}$.

Examples:
- $K = (q, 2, 3)$

\[
\left( \begin{array}{c}
[q] \\
\wedge
\end{array} \right) \left( \begin{array}{c}
[B]; [C_1]; [B]; [C_1]; [B] \\
\wedge \\
\ell = 1
\end{array} \right) \left( \begin{array}{c}
[X] \\
\wedge \\
\ell = 1
\end{array} \right) \left( \begin{array}{c}
[B]; [C_2]; [B]; [C_2]; [B] \\
\wedge \\
\ell = 1
\end{array} \right)
\]
We use $\text{Obs} = \{\text{obs}\}$ with $D(\text{obs}) = Q_M \cup \{C_1, C_2, B, X\}$.

**Examples:**

- $K = (q, 2, 3)$

\[
\left(\begin{array}{c} \lceil q \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil B \rceil ; \lceil C_1 \rceil ; \lceil B \rceil ; \lceil C_1 \rceil ; \lceil B \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil X \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil B \rceil ; \lceil C_2 \rceil ; \lceil B \rceil ; \lceil C_2 \rceil ; \lceil B \rceil ; \lceil C_2 \rceil ; \lceil B \rceil \\
\wedge \\
\ell = 1 \end{array}\right)
\]

- $K_0 = (q_0, 0, 0)$

\[
\left(\begin{array}{c} \lceil q_0 \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil B \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil X \rceil \\
\wedge \\
\ell = 1 \end{array}\right) ;
\left(\begin{array}{c} \lceil B \rceil \\
\wedge \\
\ell = 1 \end{array}\right)
\]

or, using abbreviations, $\lceil q_0 \rceil^1 ; \lceil B \rceil^1 ; \lceil X \rceil^1 ; \lceil B \rceil^1$. 
Formula Construction for Given 2-CM
Construction of $F(\mathcal{M})$

In the following, we give DC formulae describing

- the **initial configuration**: $\textit{init}$,
- the **general form of configurations**: $\textit{keep}$,
- the **transitions between configurations**: $F(q: \textit{inc}_i : q')$ and $F(q: \textit{dec}_i : q')$,
- the handling of the **final state**.

$F(\mathcal{M})$ is the conjunction of all these formulae:

$$F(\mathcal{M}) = \textit{init} \land \textit{keep} \land \ldots$$

$$\land \bigwedge_{q: \textit{inc}_i : q' \in \text{Prog}} F(q: \textit{inc}_i : q')$$

$$\land \bigwedge_{q: \textit{dec}_i : q' \in \text{Prog}} F(q: \textit{dec}_i : q')$$
**Initial and General Configurations**

\[
\text{init} : \iff (\ell \geq 4 \implies [q_0]^{1} ; [B]^{1} ; [X]^{1} ; [B]^{1} ; \text{true})
\]

\[
\text{keep} : \iff \Box ( [Q]^{1} ; [B \lor C_1]^{1} ; [X]^{1} ; [B \lor C_2]^{1} ; \ell = 4 ) \\
\implies (\ell = 4 ; [Q]^{1} ; [B \lor C_1]^{1} ; [X]^{1} ; [B \lor C_2]^{1})
\]

where \( Q := \neg (X \lor C_1 \lor C_2 \lor B) \).
Auxiliary Formula Pattern copy

\[ \text{copy}(F, \{P_1, \ldots, P_n\}) : \equiv \]

\[ \forall c, d \bullet \square ((F \land \ell = c) \land (\lceil P_1 \lor \cdots \lor P_n \rceil \land \ell = d) \land \lceil P_1 \rceil \land \ell = 4 \implies (\ell = c + d + 4 \land \lceil P_1 \rceil) \]

\[ \land \cdots \]

\[ \forall c, d \bullet \square ((F \land \ell = c) \land (\lceil P_1 \lor \cdots \lor P_n \rceil \land \ell = d) \land \lceil P_n \rceil \land \ell = 4 \implies \ell = c + d + 4 \land \lceil P_n \rceil \]

\[ \square (F \land \lceil P_1 \lor \cdots \lor P_n \rceil \land \lceil P_1 \rceil \land \ell = c \land \ell = d \land \ell = 4 \implies \ell = c + d + 4 \land \lceil P_1 \rceil) \]
\( q : \text{inc}_1 : q' \ (\text{Increment}) \)

(i) Change state

\[ \square([q]^1 ; [B \lor C_1]^1 ; [X]^1 ; [B \lor C_2]^1 ; \ell = 4 \implies \ell = 4 ; [q']^1 ; \text{true}) \]
$$q : inc_1 : q' \quad (\text{Increment})$$

(i) Change state

\[
\square([q]^1; [B \lor C_1]^1; [X]^1; [B \lor C_2]^1; \ell = 4 \implies \ell = 4; [q']^1; true)
\]

\[
\square\left(\begin{array}{cccc}
[q] & [B \lor C_1] & [X] & [B \lor C_2] \\
\ell = 1 & \ell = 1 & \ell = 1 & \ell = 1 \\
\implies & & & \ell = 4 \\
\ell = 4 & & \ell = 1 & true
\end{array}\right)
\]

(ii) Increment counter

\[
\forall d \bullet \square([q]^1; [B]^d; (\ell = 0 \lor [C_1]; [\neg X]); [X]^1; [B \lor C_2]^1; \ell = 4 \\
\implies \ell = 4; [q']^1; ([B]; [C_1]; [B] \land \ell = d); true
\]

\[
\forall d \bullet \square\left(\begin{array}{cccc}
q & [B] & [X] & [B \lor C_2] \\
\ell = 1 & \ell = d & \ell = 1 & \ell = 1 \\
\implies & & & \ell = 4 \\
\ell = 4 & & \ell = 1 & \ell = d \\
\end{array}\right)
\]
$q : \text{inc}_1 : q'$ (Increment)

(i) Keep rest of first counter

$$\text{copy}(\lceil q \rceil^1 ; \lceil B \lor C \rceil^1 ; \lceil C \rceil^1, \{B, C\})$$

(ii) Leave second counter unchanged

$$\text{copy}(\lceil q \rceil^1 ; \lceil B \lor C \rceil^1 ; \lceil X \rceil^1, \{B, C\})$$
\(q : dec_1 : q', q''\) (Decrement)

(i) If zero

\[\square([-q^1] ; [B]^1 ; [X]^1 ; [B \lor C_2]^1 ; \ell = 4 \implies \ell = 4 ; [q']^1 ; [B]^1 ; true)\]

(ii) Decrement counter

\[\forall d \bullet \square([-q^1] ; ([B] ; [C_1] \land \ell = d) ; [B] ; [B \lor C_1] ; [X]^1 ; [B \lor C_2]^1 ; \ell = 4 \implies \ell = 4 ; [q'']^1 ; [B]^d ; true)\]

(iii) Keep rest of first counter

\[copy([-q^1] ; [B] ; [C_1] ; [B_1], \{B, C_1\})\]

(iv) Leave second counter unchanged

\[copy([-q^1] ; [B \lor C_1] ; [X]^1, \{B, C_2\})\]
Final State

\( \text{copy}(\lceil q_{\text{fin}} \rceil^1; \lceil B \lor C_1 \rceil^1; \lceil X \rceil; \lceil B \lor C_2 \rceil^1, \{q_{\text{fin}}, B, X, C_1, C_2\}) \)

\[ \vdash \]

\( M \) diverges

if

\[ \vdash (M) \land \lnot \Delta[q_{\text{fin}}] \]

is realisable from 0
Satisfiability / Validity
Satisfiability

- Following Chaochen and Hansen (2004) we can observe that

\[ M \text{ halts if and only if } \text{ the DC formula } F(M) \land \Diamond \lceil q_{\text{fin}} \rceil \text{ is satisfiable.} \]

This yields

**Theorem 3.11.**
The satisfiability problem for DC with continuous time is undecidable.

(It is semi-decidable.)

- Furthermore, by taking the contraposition, we see

\[ M \text{ diverges if and only if } \text{ the DC formula } F(M) \land \neg \Diamond \lceil q_{\text{fin}} \rceil \text{ is not satisfiable.} \]

- Thus whether a DC formula is not satisfiable is not decidable, not even semi-decidable.
By Remark 2.13, $F$ is valid iff $\neg F$ is not satisfiable, so

**Corollary 3.12.** The validity problem for DC with continuous time is undecidable, not even semi-decidable.

This provides us with an alternative proof of Theorem 2.23 ("there is no sound and complete proof system for DC"): 

- **Suppose** there were such a calculus $C$.
- By Lemma 2.22 it is semi-decidable whether a given DC formula $F$ is a theorem in $C$.
- By the soundness and completeness of $C$, $F$ is a theorem in $C$ if and only if $F$ is valid.
- Thus it is semi-decidable whether $F$ is valid. **Contradiction.**
Note: the DC fragment defined by the following grammar is **sufficient** for the reduction

\[
F ::= [P] \mid \neg F_1 \mid F_1 \lor F_2 \mid F_1 ; F_2 \mid \ell = 1 \mid \ell = x \mid \forall x \cdot F_1,
\]

\(P\) a state assertion, \(x\) a global variable.

Formulae used in the reduction are abbreviations:

\[
\ell = 4 \iff \ell = 1 ; \ell = 1 ; \ell = 1 ; \ell = 1
\]

\[
\ell \geq 4 \iff \ell = 4 ; \text{true}
\]

\[
\ell = x + y + 4 \iff \ell = x ; \ell = y ; \ell = 4
\]

Length 1 is not necessary – we can use \(\ell = z\) instead, with fresh \(z\).

This is RDC augmented by “\(\ell = x\)” and “\(\forall x\)”, which we denote by \(\text{RDC} + \ell = x, \forall x\).
RDC $+\ell = x, \forall x$ in Continuous Time

- Outline of the proof
- Recall: two-counter machines (2-CM)
  - states and commands (syntax)
  - configurations and computations (semantics)

- Encoding configurations in DC
- initial configuration of a 2-CM

- Encoding transitions in DC
  - increment counter,
  - decrement counter,
  - and some helper formulae.

- Satisfiability and Validity
- Discussion
Tell Them What You’ve Told Them...

- For **Restricted DC** plus $\ell = x$ and $\forall x$ in continuous time:
  - **satisfiability** is **undecidable**.
  - **Proof idea**: reduce to halting problem of two-counter machines.

- For full DC, it doesn't get better.
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

\[ \text{obs} : \text{Time} \rightarrow \mathcal{D}(\text{obs}) \]

\[ \langle \text{obs}_0, \nu_0 \rangle, t_0 \xrightarrow{\lambda_0} \langle \text{obs}_1, \nu_1 \rangle, t_1 \ldots \]

- 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 …
References
References
