Duration Calculus
Introduction

Michael R. Hansen

mrh@imm.dtu.dk

Informatics and Mathematical Modelling

Technical University of Denmark

Overview

- Background
- Short introduction
- Decidability results
- Undecidability results

Background

- Provable Correct Systems (ProCoS, ESPRIT BRA 3104)
  Bjørner Langmaack Hoare Olderog
- Project case study: Gas Burner Sørensen Ravn Rischel

Background

- Provable Correct Systems (ProCoS, ESPRIT BRA 3104)
  Bjørner Langmaack Hoare Olderog
- Project case study: Gas Burner Sørensen Ravn Rischel

  Intervals properties

  Timed Automata, Real-time Logic, Metric Temporal Logic, Explicit Clock Temporal, . . ., Alur, Dill, Jahanian, Mok, Koymans, Harel, Lichtenstein, Pnueli, . . .

Background

- Provable Correct Systems (ProCoS, ESPRIT BRA 3104)
  Bjørner Langmaack Hoare Olderog
- Project case study: Gas Burner Sørensen Ravn Rischel

  Intervals properties

  Timed Automata, Real-time Logic, Metric Temporal Logic, Explicit Clock Temporal, . . ., Alur, Dill, Jahanian, Mok, Koymans, Harel, Lichtenstein, Pnueli, . . .

  Duration of states

  Duration Calculus Zhou Hoare Ravn 91

Background

- Provable Correct Systems (ProCoS, ESPRIT BRA 3104)
  Bjørner Langmaack Hoare Olderog
- Project case study: Gas Burner Sørensen Ravn Rischel

  Intervals properties

  Timed Automata, Real-time Logic, Metric Temporal Logic, Explicit Clock Temporal, . . ., Alur, Dill, Jahanian, Mok, Koymans, Harel, Lichtenstein, Pnueli, . . .

  Duration of states

  Duration Calculus Zhou Hoare Ravn 91

  — an Interval Temporal Logic Halpern Moszkowski Manna

Background

- Provable Correct Systems (ProCoS, ESPRIT BRA 3104)
  Bjørner Langmaack Hoare Olderog
- Project case study: Gas Burner Sørensen Ravn Rischel
  Intervals properties
  Timed Automata, Real-time Logic, Metric Temporal Logic, Explicit Clock Temporal, . . ., Alur, Dill, Jahanian, Mok, Koymans, Harel, Lichtenstein, Pnueli, . . .
  Duration of states
  Duration Calculus Zhou Hoare Ravn 91
  — an Interval Temporal Logic Halpern Moszkowski Manna
- Logical Calculi, Applications, Mechanical Support
- Duration Calculus: A formal approach to real-time systems
  Zhou Chaochen and Michael R. Hansen
  Springer 2004

Gas Burner example: Requirements

State variables modelling Gas and Flame:

$$\mathrm{G}, \mathrm{F} : \mathbb{T}\text{ime} \rightarrow \{0, 1\}$$

State expression modelling that gas is Leaking

$$\mathrm{L} \mathrel{\widehat{=}} \mathrm{G} \land \neg \mathrm{F}$$

Requirement

- Gas must at most be leaking 1/20 of the elapsed time

$$(e - b) \geq 60\,\mathrm{s} \;\Rightarrow\; 20 \int_b^e \mathrm{L}(t)dt \leq (e - b)$$

Gas Burner example: Design decisions

- Leaks are detectable and stoppable within 1s:

$$\forall c, d : b \leq c < d \leq e.(\mathrm{L}[c, d] \Rightarrow (d - c) \leq 1\,\mathrm{s})$$

where

$$P[c, d] \;\widehat{=}\; \int_c^d P(t) = (d - c) > 0$$

which reads "P holds throughout $[c, d]$"

Gas Burner example: Design decisions

- Leaks are detectable and stoppable within 1s:

  $$\forall c, d : b \le c < d \le e.(\mathrm{L}[c, d] \;\Rightarrow\; (d - c) \le 1\,\mathrm{s})$$

  where

  $$P[c, d] \;\widehat{=}\; \int_c^d P(t) = (d - c) > 0$$

  which reads "P holds throughout $[c, d]$"

- At least 30s between leaks:

  $$\forall c, d, r, s : b \le c < r < s < d \le e. \\ (\mathrm{L}[c, r] \wedge \neg \mathrm{L}[r, s] \wedge \mathrm{L}[s, d]) \;\Rightarrow\; (s - r) \ge 30\,\mathrm{s}$$

Interval Logic [Halpern Moszkowski Manna 83]

Terms: $\theta \ ::= \ x \mid v \mid \theta_1 + \theta_n \mid \ \ldots$ Temporal Variable

Formulas: $\phi \ ::= \ \theta_1 = \theta_n \mid \neg\phi \mid \phi \vee \psi \mid \phi \frown \psi \mid (\exists x)\phi \mid \ \ldots$ chop

Interval Logic [Halpern Moszkowski Manna 83]

Terms: $\theta \ ::= \ x \mid v \mid \theta_1 + \theta_n \mid \ \ldots$ Temporal Variable

$v : \mathbb{I}\text{ntv} \rightarrow \mathbb{R}$

Formulas: $\phi \ ::= \ \theta_1 = \theta_n \mid \neg\phi \mid \phi \vee \psi \mid \phi \frown \psi \mid (\exists x)\phi \mid \ \ldots$ chop

$\phi : \mathbb{I}\text{ntv} \rightarrow \{\text{tt,ff}\}$

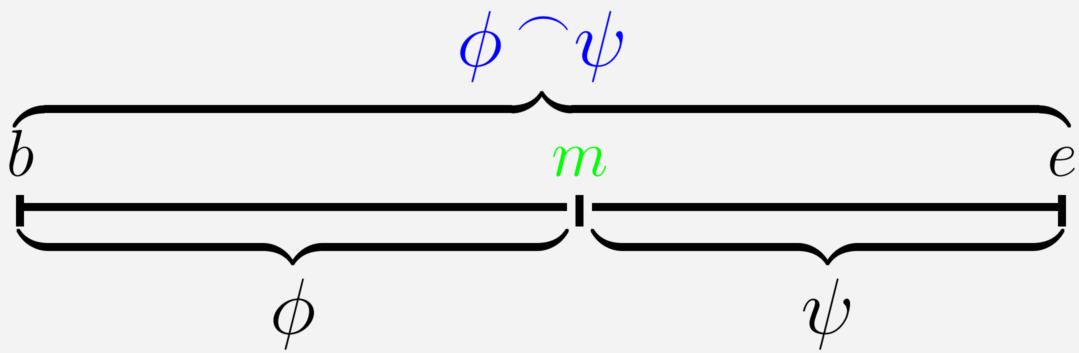
Terms: $\theta \ ::= \ x \mid v \mid \theta_1 + \theta_n \mid \ \ldots$ Temporal Variable

v : $\mathbb{I}$ntv $\longrightarrow \mathbb{R}$

Formulas: $\phi \ ::= \ \theta_1 = \theta_n \mid \neg\phi \mid \phi \vee \psi \mid \phi {}^\frown \psi \mid (\exists x)\phi \mid \ \ldots$ chop

ϕ : $\mathbb{I}$ntv $\longrightarrow \{$tt,ff$\}$

Chop:



for some $m : b \leq m \leq e$

Interval Logic [Halpern Moszkowski Manna 83]

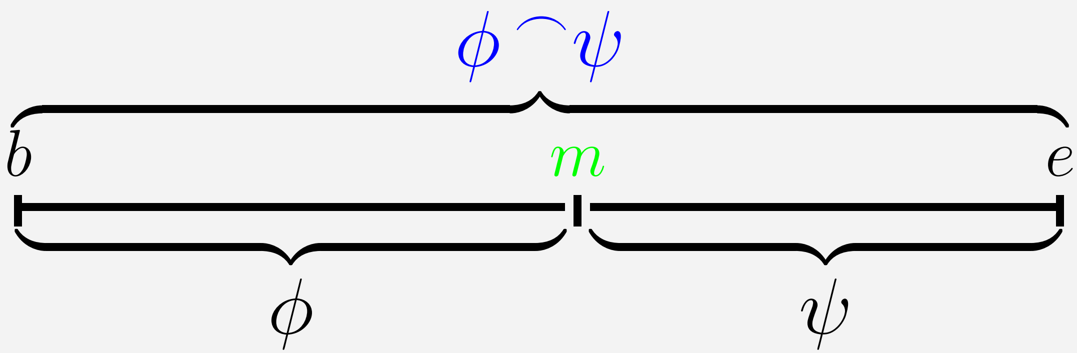
Terms: $\theta \ ::= \ x \mid v \mid \theta_1 + \theta_n \mid \ \ldots$ Temporal Variable

$v : \mathbb{I}\text{ntv} \rightarrow \mathbb{R}$

Formulas: $\phi \ ::= \ \theta_1 = \theta_n \mid \neg\phi \mid \phi \vee \psi \mid \phi \frown \psi \mid (\exists x)\phi \mid \ \ldots$ chop

$\phi : \mathbb{I}\text{ntv} \rightarrow \{\text{tt},\text{ff}\}$

Chop:



for some $m : b \leq m \leq e$

In DC: $\mathbb{I}\text{ntv} = \{ \ [a, b] \mid a, b \in \mathbb{R} \wedge a \leq b\}$

Duration Calculus [Zhou Hoare Ravn 91]

- State variables $P: \mathbb{T}\text{ime} \rightarrow \{0,1\}$ Finite Variablilty
- State expressions $S ::= 0 \mid 1 \mid P \mid \neg S \mid S_1 \vee S_2$

  $S: \mathbb{T}\text{ime} \rightarrow \{0,1\}$ pointwise defined

Duration Calculus [Zhou Hoare Ravn 91]

- State variables $P: \mathbb{T}\text{ime} \longrightarrow \{0, 1\}$ Finite Variablilty
- State expressions $S ::= 0 \mid 1 \mid P \mid \neg S \mid S_1 \vee S_2$

  $S : \mathbb{T}\text{ime} \longrightarrow \{0, 1\}$ pointwise defined

- Durations $\int S : \mathbb{I}\text{ntv} \longrightarrow \mathbb{R}$ defined on $[b, e]$ by

  $$\int_b^e S(t)dt$$

  – Temporal variables with a structure

Example: Gas Burner

Requirement

$$\ell \geq 60 \;\Rightarrow\; 20\int \mathrm{L} \leq \ell$$

Design decisions

$$D_1 \;\widehat{=}\; \Box(\lceil\!\lceil \mathrm{L} \rceil\!\rceil \Rightarrow \ell \leq 1)$$

$$D_2 \;\widehat{=}\; \Box((\lceil\!\lceil \mathrm{L} \rceil\!\rceil \frown \lceil\!\lceil \neg\mathrm{L} \rceil\!\rceil \frown \lceil\!\lceil \mathrm{L} \rceil\!\rceil) \Rightarrow \ell \geq 30)$$

where ℓ denotes the *length* of the interval, and

$\Diamond\phi$	$\widehat{=}$	true $\frown \phi \frown$ true	"for some sub-interval: ϕ"
$\Box\phi$	$\widehat{=}$	$\neg\Diamond\neg\phi$	"for all sub-intervals: ϕ"
$\lceil\!\lceil P \rceil\!\rceil$	$\widehat{=}$	$\int P = \ell \wedge \ell > 0$	"P holds throughout a non-point inter

Example: Gas Burner

Requirement

$$\ell \geq 60 \;\Rightarrow\; 20\int \mathrm{L} \leq \ell$$

Design decisions

$$D_1 \;\widehat{=}\; \Box(\llbracket \mathrm{L} \rrbracket \Rightarrow \ell \leq 1)$$

$$D_2 \;\widehat{=}\; \Box((\llbracket \mathrm{L} \rrbracket {}^\frown \llbracket \neg \mathrm{L} \rrbracket {}^\frown \llbracket \mathrm{L} \rrbracket) \Rightarrow \ell \geq 30)$$

where ℓ denotes the *length* of the interval, and

$\Diamond\phi$	$\widehat{=}$	true $^\frown\phi^\frown$true	"for some sub-interval: ϕ"
$\Box\phi$	$\widehat{=}$	$\neg\Diamond\neg\phi$	"for all sub-intervals: ϕ"
$\llbracket P \rrbracket$	$\widehat{=}$	$\int P = \ell \,\wedge\, \ell > 0$	"P holds throughout a non-point inter

succinct formulation — no interval endpoints

Correctness — informal interval reasoning

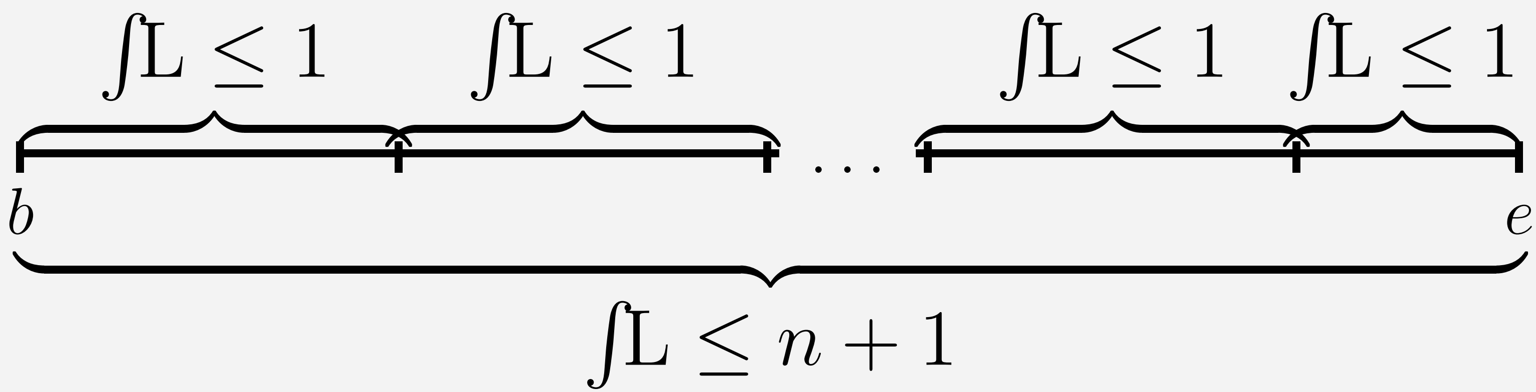
We must establish: $(D_1 \ \wedge \ D_2) \Rightarrow \ \ell \geq 60 \ \Rightarrow \ 20 \int \mathrm{L} \leq \ell$

Correctness — informal interval reasoning

We must establish: $(D_1 \ \wedge \ D_2) \Rightarrow \ \ell \geq 60 \ \Rightarrow \ 20 \int \mathrm{L} \leq \ell$

Correctness — informal interval reasoning

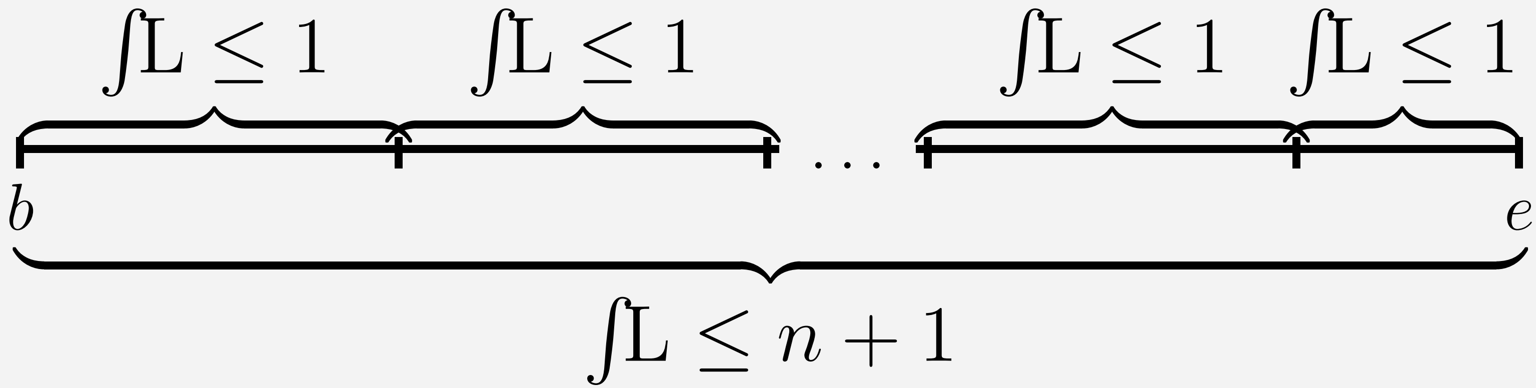
Note that $(D_1 \wedge D_2) \;\Rightarrow \Box(\ell \leq 30 \Rightarrow \int\mathrm{L} \leq 1)$:



We must establish: $(D_1 \;\wedge\; D_2) \Rightarrow \;\ell \geq 60 \;\Rightarrow\; 20\int\mathrm{L} \leq \ell$

Correctness — informal interval reasoning

Note that $(D_1 \wedge D_2) \;\Rightarrow \Box(\ell \leq 30 \Rightarrow \int\mathrm{L} \leq 1)$:



Since $n \geq 2 \Rightarrow 20 \cdot (n+1) \leq 30 \cdot n$ we have

$$(D_1 \;\wedge\; D_2) \Rightarrow \;\ell \geq 60 \;\Rightarrow\; 20\int\mathrm{L} \leq \ell$$

Decidability

[Zhou Hansen Sestoft 93]

Restricted Duration Calculus :

- $\llbracket S \rrbracket$
- $\neg\phi,\ \phi \vee \psi,\ \phi ^\frown \psi$

Satisfiability is reduced to emptiness of regular languages

Both for discrete and continuous time

Decidability

[Zhou Hansen Sestoft 93]

Restricted Duration Calculus :

- $[\![S]\!]$
- $\neg\phi,\ \phi \vee \psi,\ \phi\frown\psi$

Satisfiability is reduced to emptiness of regular languages

Both for discrete and continuous time

Skakkebæk Sestoft 94, Pandya 01, Fränzle 02, Gomez Bowman 03

Decidability

[Zhou Hansen Sestoft 93]

Restricted Duration Calculus :

- $\llbracket S \rrbracket$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Both for discrete and continuous time

Skakkebæk Sestoft 94, Pandya 01, Fränzle 02, Gomez Bowman 03

Even small extensions give undecidable subsets

RDC_1 (Cont. time)	RDC_2	RDC_3
• $\ell = r,\ \llbracket S \rrbracket$ • $\neg\phi,\ \phi\vee\psi, \phi\frown\psi$	• $\int S_1 = \int S_2$ • $\neg\phi,\ \phi\vee\psi, \phi\frown\psi$	• $\ell = x,\ \llbracket S \rrbracket$ • $\neg\phi,\ \phi\vee\psi, \phi\frown\psi,\ (\exists x)\phi$

Discrete-Time Duration Calculus

- For an interpretation

$$\mathcal{I} \; : \; SVar \longrightarrow (\mathbb{T}\text{ime} \longrightarrow \{0, 1\})$$

the discontinuity points of each $P_{\mathcal{I}}$ must belong to $\mathbb{N}$.

Discrete-Time Duration Calculus

- For an interpretation

$$\mathcal{I} \; : \; SVar \rightarrow (\mathbb{T}\text{ime} \rightarrow \{0, 1\})$$

the discontinuity points of each $P_{\mathcal{I}}$ must belong to $\mathbb{N}$.

- We consider only *discrete intervals*

$$[b, e] \in \mathbb{I}\text{ntv}$$

where $b, e \in \mathbb{N}$.

Discrete-Time Duration Calculus

- For an interpretation

$$\mathcal{I} \; : \; SVar \rightarrow (\mathbb{T}\text{ime} \rightarrow \{0,1\})$$

the discontinuity points of each $P_{\mathcal{I}}$ must belong to $\mathbb{N}$.

- We consider only *discrete intervals*

$$[b,e] \in \mathbb{I}\text{ntv}$$

where $b, e \in \mathbb{N}$.

- The semantics of chop is

$$\mathcal{I},[b,e] \models \phi \frown \psi \;\text{ iff }\; \left\{ \begin{array}{l} \mathcal{I},[b,m] \models \phi \text{ and } \mathcal{I},[m,e] \models \psi, \\ \text{for some } m \in [b,e] \text{ where } m \in \mathbb{N} \end{array} \right\}$$

Discrete- vs Continuous-Time DC

- The formula

$$\ell = 1 \iff \llbracket 1 \rrbracket \wedge \neg(\llbracket 1 \rrbracket \frown \llbracket 1 \rrbracket)$$

is valid for discrete time; but not for continuous time

Discrete- vs Continuous-Time DC

- The formula

  $$\ell = 1 \iff \llbracket 1 \rrbracket \wedge \neg(\llbracket 1 \rrbracket \frown \llbracket 1 \rrbracket)$$

  is valid for discrete time; but not for continuous time

- The formula

  $$\llbracket S \rrbracket \Rightarrow (\llbracket S \rrbracket \frown \llbracket S \rrbracket)$$

  is valid for continuous time; but not for discrete time

Restricted Duration Calculus (*RDC*)

1. if S is a state expression, then $\llbracket S \rrbracket \in RDC$, and
2. if $\phi, \psi \in RDC$, then $\neg\phi, \phi \vee \psi, \phi \frown \psi \in RDC$.

Restricted Duration Calculus (*RDC*)

1. if S is a state expression, then $[\![S]\!] \in RDC$, and
2. if $\phi, \psi \in RDC$, then $\neg\phi, \phi \vee \psi, \phi \frown \psi \in RDC$.

Expressiveness of *RDC* for Discrete Time:

$$
\begin{array}{lcl}
\ell = 0 & \iff & \neg[\![1]\!] \\
\int S = 0 & \iff & [\![\neg S]\!] \vee \ell = 0 \\
\ell = 1 & \iff & [\![1]\!] \wedge \neg([\![1]\!] \frown [\![1]\!]) \\
\int S = 1 & \iff & (\int S = 0) \frown ([\![S]\!] \wedge \ell = 1) \frown (\int S = 0) \\
\int S = k+1 & \iff & (\int S = k) \frown (\int S = 1) \\
\int S \geq k & \iff & (\int S = k) \frown \text{true} \\
\int S > k & \iff & (\int S \geq k) \wedge \neg(\int S = k) \\
\int S \leq k & \iff & \neg(\int S > k) \\
\int S < k & \iff & (\int S \leq k) \wedge \neg(\int S = k)
\end{array}
$$

Decidability of *RDC* for Discrete Time

Satisfiability is reduced to emptiness of regular languages

Decidability of *RDC* for Discrete Time

Satisfiability is reduced to emptiness of regular languages

Idea: $a \in \Sigma$ describes a piece of an interpretation, e.g. $P_1 \wedge \neg P_2 \wedge P_3$

Decidability of *RDC* for Discrete Time

Satisfiability is reduced to emptiness of regular languages

Idea: $a \in \Sigma$ describes a piece of an interpretation, e.g. $P_1 \wedge \neg P_2 \wedge P_3$

Discrete time — one letter corresponds to one time unit

$$
\begin{aligned}
\mathcal{L}(\llbracket S \rrbracket) &= (DNF(S))^+ \\
\mathcal{L}(\varphi \vee \psi) &= \mathcal{L}(\varphi) \cup \mathcal{L}(\psi) \\
\mathcal{L}(\neg\varphi) &= \Sigma^* \setminus \mathcal{L}(\varphi) \\
\mathcal{L}(\varphi \frown \psi) &= \mathcal{L}(\varphi)\, \mathcal{L}(\psi)
\end{aligned}
$$

Decidability of *RDC* for Discrete Time

Satisfiability is reduced to emptiness of regular languages

Idea: $a \in \Sigma$ describes a piece of an interpretation, e.g. $P_1 \wedge \neg P_2 \wedge P_3$

Discrete time — one letter corresponds to one time unit

$$\begin{array}{rcl}
\mathcal{L}(\llbracket S \rrbracket) & = & (DNF(S))^+ \\
\mathcal{L}(\varphi \vee \psi) & = & \mathcal{L}(\varphi) \cup \mathcal{L}(\psi) \\
\mathcal{L}(\neg\varphi) & = & \Sigma^* \setminus \mathcal{L}(\varphi) \\
\mathcal{L}(\varphi \frown \psi) & = & \mathcal{L}(\varphi)\, \mathcal{L}(\psi)
\end{array}$$

- $\mathcal{L}(\llbracket \phi \rrbracket)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\llbracket \phi \rrbracket \neq \emptyset$

Example

- Is the formula $(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket$ valid for discrete time?

Example

- Is the formula $(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket$ valid for discrete time?
- $\Sigma = \{\{P\}, \{\}\}$.

Example

- Is the formula $(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket$ valid for discrete time?
- $\Sigma = \{\{P\}, \{\}\}$.
- We have

$$
\begin{array}{ll}
 & (\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket \text{ is valid} \\
\text{iff} & \neg((\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket) \text{ is not satisfiable} \\
\text{iff} & (\llbracket P \rrbracket \frown \llbracket P \rrbracket) \wedge \neg \llbracket P \rrbracket \text{ is not satisfiable} \\
\text{iff} & \mathcal{L}_1(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \cap \mathcal{L}_1(\neg \llbracket P \rrbracket) = \{\} \\
\text{iff} & \{\{P\}^i \mid i \geq 2\} \cap (\Sigma^* \setminus \{\{P\}^i \mid i \geq 1\}) = \{\}
\end{array}
$$

The last equality holds.

Example

- Is the formula $(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket$ valid for discrete time?
- $\Sigma = \{\{P\}, \{\}\}$.
- We have

$$
\begin{array}{ll}
 & (\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket \text{ is valid} \\
\text{iff} & \neg((\llbracket P \rrbracket \frown \llbracket P \rrbracket) \Rightarrow \llbracket P \rrbracket) \text{ is not satisfiable} \\
\text{iff} & (\llbracket P \rrbracket \frown \llbracket P \rrbracket) \wedge \neg \llbracket P \rrbracket \text{ is not satisfiable} \\
\text{iff} & \mathcal{L}_1(\llbracket P \rrbracket \frown \llbracket P \rrbracket) \cap \mathcal{L}_1(\neg \llbracket P \rrbracket) = \{\} \\
\text{iff} & \{\{P\}^i \mid i \geq 2\} \cap (\Sigma^* \setminus \{\{P\}^i \mid i \geq 1\}) = \{\}
\end{array}
$$

  The last equality holds.

- Therefore, the formula is valid for discrete time.

Undecidability [Zhou Hansen Sestoft 93]

Restricted Duration Calculus :

- $\llbracket S \rrbracket$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Both for discrete and continuous time

Appearrently small extensions give undecidable subsets

RDC_1 (Cont. time)	RDC_2	RDC_3
• $\ell = r,\ \llbracket S \rrbracket$ • $\neg\phi,\ \phi\vee\psi,\ \phi\frown\psi$	• $\int S_1 = \int S_2$ • $\neg\phi,\ \phi\vee\psi,\ \phi\frown\psi$	• $\ell = x,\ \llbracket S \rrbracket$ • $\neg\phi,\ \phi\vee\psi,\ \phi\frown\psi,\ (\exists x)\phi$

Two-Counter Machines

- A *two-counter machine* has an *initial label* q_0, two *counters* c_1 and c_2 which can hold arbitrary natural numbers from $\mathbb{N} = \{0, 1, 2, \ldots\}$, and a finite set of labeled *instructions* m_i.

Two-Counter Machines

- A *two-counter machine* has an *initial label* q_0, two *counters* c_1 and c_2 which can hold arbitrary natural numbers from $\mathbb{N} = \{0, 1, 2, \ldots\}$, and a finite set of labeled *instructions* m_i.
- A *configuration* s has the form: (q, n_1, n_2), where q is the current label, and n_1 and n_2 are the values of c_1 and c_2, respectively.

Two-Counter Machines

- A *two-counter machine* has an *initial label* q_0, two *counters* c_1 and c_2 which can hold arbitrary natural numbers from $\mathbb{N} = \{0, 1, 2, \ldots\}$, and a finite set of labeled *instructions* m_i.
- A *configuration* s has the form: (q, n_1, n_2), where q is the current label, and n_1 and n_2 are the values of c_1 and c_2, respectively.
- Instructions for c_1 (and similarly for c_2):

Instruction	s	$\Longrightarrow s'$
$q : c_1^+ \rightarrow q_j$	(q, n_1, n_2)	$\Longrightarrow (q_j, n_1 + 1, n_2)$
$q : c_1^- \rightarrow q_j, q_k$	$(q, 0, n_2)$	$\Longrightarrow (q_j, 0, n_2)$
$q : c_1^- \rightarrow q_j, q_k$	$(q, n_1 + 1, n_2)$	$\Longrightarrow (q_k, n_1, n_2)$

Two-Counter Machines

- A *two-counter machine* has an *initial label* q_0, two *counters* c_1 and c_2 which can hold arbitrary natural numbers from $\mathbb{N} = \{0, 1, 2, \ldots\}$, and a finite set of labeled *instructions* m_i.
- A *configuration* s has the form: (q, n_1, n_2), where q is the current label, and n_1 and n_2 are the values of c_1 and c_2, respectively.
- Instructions for c_1 (and similarly for c_2):

Instruction	s	$\Longrightarrow s'$
$q : c_1^+ \rightarrow q_j$	(q, n_1, n_2)	$\Longrightarrow (q_j, n_1 + 1, n_2)$
$q : c_1^- \rightarrow q_j, q_k$	$(q, 0, n_2)$	$\Longrightarrow (q_j, 0, n_2)$
$q : c_1^- \rightarrow q_j, q_k$	$(q, n_1 + 1, n_2)$	$\Longrightarrow (q_k, n_1, n_2)$

- Halting problem is undecidable

Two-Counter Machines

- A *two-counter machine* has an *initial label* q_0, two *counters* c_1 and c_2 which can hold arbitrary natural numbers from $\mathbb{N} = \{0, 1, 2, \ldots\}$, and a finite set of labeled *instructions* m_i.
- A *configuration* s has the form: (q, n_1, n_2), where q is the current label, and n_1 and n_2 are the values of c_1 and c_2, respectively.
- Instructions for c_1 (and similarly for c_2):

Instruction	s	$\Longrightarrow s'$
$q : c_1^+ \longrightarrow q_j$	(q, n_1, n_2)	$\Longrightarrow (q_j, n_1 + 1, n_2)$
$q : c_1^- \longrightarrow q_j, q_k$	$(q, 0, n_2)$	$\Longrightarrow (q_j, 0, n_2)$
$q : c_1^- \longrightarrow q_j, q_k$	$(q, n_1 + 1, n_2)$	$\Longrightarrow (q_k, n_1, n_2)$

- Halting problem is undecidable

Assume deterministic machine with one halting state q_{fin}

Undecidability 1: Continuous time only

1. the formula $\ell = r$ belongs to $RDC_1(r)$,
2. if S is a state expression, then $\llbracket S \rrbracket$ belongs to $RDC_1(r)$, and
3. if ϕ and ψ belong to $RDC_1(r)$, then so do $\neg\phi$, $\phi \vee \psi$, and $\phi{}^\frown\psi$.

Undecidability 1: Continuous time only

1. the formula $\ell = r$ belongs to $RDC_1(r)$,
2. if S is a state expression, then $[\![S]\!]$ belongs to $RDC_1(r)$, and
3. if ϕ and ψ belong to $RDC_1(r)$, then so do $\neg\phi$, $\phi \vee \psi$, and $\phi ^\frown \psi$.

Encoding of two-counter machine M:

- one state variable Q_i for each label q_i. Let $\mathcal{Q} = \{Q_0, \ldots, Q_{fin}\}$
- two state variables C_1 and C_2 to represent the counter values
- two auxiliary state variables B and L, used as delimiters

Undecidability 1: Continuous time only

1. the formula $\ell = r$ belongs to $RDC_1(r)$,
2. if S is a state expression, then $\llbracket S \rrbracket$ belongs to $RDC_1(r)$, and
3. if ϕ and ψ belong to $RDC_1(r)$, then so do $\neg\phi$, $\phi \vee \psi$, and $\phi \frown \psi$.

Encoding of two-counter machine M:

- one state variable Q_i for each label q_i. Let $\mathcal{Q} = \{Q_0, \ldots, Q_{\mathit{fin}}\}$
- two state variables C_1 and C_2 to represent the counter values
- two auxiliary state variables B and L, used as delimiters

A configuration (q, n_1, n_2) is encoded on an interval of length $4r$:

$$| \underbrace{Q}_{r} | \underbrace{Val_1}_{r} | \underbrace{L}_{r} | \underbrace{Val_2}_{r} |$$

where Val_j represents the value of counter c_j.

Undecidability 1 – Abbreviations

$$
\begin{array}{lcl}
\llbracket\ \rrbracket & \widehat{=} & \neg\llbracket 1 \rrbracket \\
\text{true} & \widehat{=} & \llbracket\ \rrbracket \vee \llbracket 1 \rrbracket \\
\ell < r & \widehat{=} & \neg((\ell = r)\frown\text{true}) \\
\ell = 4r & \widehat{=} & (\ell = 2r)\frown(\ell = 2r) \\
\llbracket S \rrbracket^r & \widehat{=} & \llbracket S \rrbracket \wedge (\ell = r) \\
\phi \leadsto \psi & \widehat{=} & \neg(\phi\frown\neg(\llbracket\ \rrbracket \vee \psi))
\end{array}
$$

- $\llbracket S \rrbracket^r$ reads "S has value one for a duration of r"
- $\phi \leadsto \psi$ reads "if the interval starts with ϕ, it must end immediately with $\llbracket\ \rrbracket$ or with ψ" — ϕ leads to ψ

Undecidability 1 – Continued

The interval describing Val_i has the following form:

$$|B|C_i|B|\cdots|B|C_i|B|$$

with n_i sections of C_i separated by B.

Undecidability 1 – Continued

The interval describing Val_i has the following form:

$$|B|C_i|B|\cdots|B|C_i|B|$$

with n_i sections of C_i separated by B.

The computation of M is simulated by a formula $F(M)$

For example, the following formula copies the C_1 sections to the same place in the next configuration.

$$\left(\llbracket Q_i \rrbracket^r \frown (\ell < r) \frown \llbracket C_1 \rrbracket \frown \left(\begin{array}{c} \llbracket C_1 \rrbracket \frown \textsf{true} \\ \wedge \\ \ell = 4r \end{array}\right)\right) \rightsquigarrow (\llbracket C_1 \rrbracket \frown \textsf{true})$$

exploits precision of length

Undecidability 1 – Continued

The interval describing Val_i has the following form:

$$|B|C_i|B|\cdots|B|C_i|B|$$

with n_i sections of C_i separated by B.

The computation of M is simulated by a formula $F(M)$

For example, the following formula copies the C_1 sections to the same place in the next configuration.

$$\left(\llbracket Q_i \rrbracket^r \frown (\ell < r) \frown \llbracket C_1 \rrbracket \frown \left(\begin{array}{c} \llbracket C_1 \rrbracket \frown \text{true} \\ \wedge \\ \ell = 4r \end{array} \right) \right) \leadsto (\llbracket C_1 \rrbracket \frown \text{true})$$

exploits precision of length

- M halts iff $F(M)$ is satisfiable
- satisfiability is undecidable for the subset under consideration

Remarks

- $\ell = 1$ is not expressible in RDC for continuous time.
- "Relaxing punctuality", replacing $\ell = r$ with $\ell < r$ does not give decidability.
- "Relaxing punctuality", replacing $\ell = r$ with $\ell > r$?

Undecidability 2: Discrete and Cont. Time

1. if S_1 and S_2 are state expressions, then $\int S_1 = \int S_2$ belongs to RDC_2, and
2. if ϕ and ψ belong to RDC_2, then so do $\neg\phi$, $\phi \vee \psi$ and $\psi \frown \psi$.

Undecidability 2: Discrete and Cont. Time

1. if S_1 and S_2 are state expressions, then $\int S_1 = \int S_2$ belongs to RDC_2, and
2. if ϕ and ψ belong to RDC_2, then so do $\neg\phi$, $\phi \vee \psi$ and $\psi \frown \psi$.

Encoding

1. two state variables C_i^+ and C_i^- for each counter c_i
2. state variables $\mathcal{Q} = \{Q_0, \ldots, Q_{fin}\}$ corresponding to the labels

Undecidability 2: Discrete and Cont. Time

1. if S_1 and S_2 are state expressions, then $\int S_1 = \int S_2$ belongs to RDC_2, and
2. if ϕ and ψ belong to RDC_2, then so do $\neg\phi$, $\phi \vee \psi$ and $\psi \frown \psi$.

Encoding

1. two state variables C_i^+ and C_i^- for each counter c_i
2. state variables $\mathcal{Q} = \{Q_0, \ldots, Q_{fin}\}$ corresponding to the labels

Idea

- the value of c_i is represented by the value of $\int C_i^+ - \int C_i^-$
- a computation $s_0\ s_1\ s_2\ \cdots$
  is represented by a sequence $|QE_0|C_0|QE_1|C_1|QE_2|C_2|\cdots$
  - QE_k is a state expression of $\mathcal{Q}$
  - C_k is a state expression of $\{C_1^+, C_2^+, C_1, C_2\}$

Undecidability 2: Abbreviations

$$
\begin{array}{lcl}
C^{\vee} & \widehat{=} & C_1^+ \vee C_1^- \vee C_2^+ \vee C_2^- \\
\llbracket S \rrbracket & \widehat{=} & (\int S = \int 1) \wedge \neg(\int 0 = \int 1) \\
Incr_1 & \widehat{=} & \llbracket C_1^+ \wedge \neg(C_1^- \vee C_2^+ \vee C_2^-) \rrbracket \\
\int S > 0 & \widehat{=} & \Diamond \llbracket S \rrbracket \\
\llbracket\ \rrbracket & \widehat{=} & \neg \llbracket 1 \rrbracket \\
\textsf{true} & \widehat{=} & \llbracket\ \rrbracket \vee \llbracket 1 \rrbracket \\
\phi \leadsto \psi & \widehat{=} & \neg(\phi \frown \neg(\llbracket\ \rrbracket \vee \psi)) \\
\Diamond_p \phi & \widehat{=} & \phi \frown \textsf{true} \qquad \text{reads: “for some prefix interval: } \phi\text{”} \\
\Box_p \phi & \widehat{=} & \neg(\Diamond_p(\neg \phi)) \qquad \text{reads: “for all prefix intervals: } \phi\text{”.}
\end{array}
$$

Undecidability 2: Encoding Instructions

The instruction $q_j : c_i^+ \rightarrow q_k$ is encoded as follows

$$\left(\left(\left(\begin{array}{c} \llbracket\ \rrbracket \\ \vee \\ (\text{true} \frown \llbracket C^\vee \rrbracket) \end{array} \right) \frown \left(\begin{array}{c} (\llbracket Q_j \rrbracket \frown \llbracket C^\vee \rrbracket) \\ \wedge \\ \int Q_j = \int C^\vee \end{array} \right) \right) \begin{array}{c} \\ \\ \\ \rightsquigarrow \end{array} \left(\begin{array}{l} \llbracket Q_k \rrbracket \\ \vee \\ (\llbracket Q_k \rrbracket \frown Incr_i) \\ \vee \\ (\llbracket Q_k \rrbracket \frown Incr_i \frown \llbracket Q^\vee \rrbracket \frown \text{true}) \end{array} \right) \right)$$

- remaining instructions follows similar pattern
- mutual exclusive sections and sections of equal size
- undecidability of RDC_2

Undecidability 3

1. if S is a state expression, then $[\![S]\!]$ belongs to RDC_3,
2. if x is a global variable, then $\ell = x$ belongs to RDC_3, and
3. if ϕ and ψ belong to RDC_3, then so do $\neg\phi$, $\phi \vee \psi$, $\phi \frown \psi$ and $(\exists x)\phi$, where x is any global variable.

A configuration of the machine is represented by a sequence of sections Q, L and C, all of the same length:

$$|Q|\underbrace{C|\cdots|C}_{n_1}|L_1|\underbrace{C|\cdots|C}_{n_2}|L_2|$$

The initial configuration, $(q_0, 0, 0)$, is represented by $|Q_0|L_1|L_2|$:

$$\exists x.([\![Q_0]\!] \wedge (\ell = x)) \frown ([\![L_1]\!] \wedge (\ell = x)) \frown ([\![L_2]\!] \wedge (\ell = x)) \frown \text{true}$$

Undecidability 3: Encoding Instructions

An abbreviation:

$$[\![S]\!]^x \;\widehat{=}\; ([\![\;]\!] \vee [\![S]\!]) \wedge (\ell = x)$$

An instruction $q_j : c_1^+ \rightarrow q_k$ transforms configurations as follows:

$$|Q_j|\underbrace{C|\cdots|C}_{n_1}|L_1|\underbrace{C|\cdots|C}_{n_2}|L_2| \Longrightarrow |Q_k|\underbrace{C|C|\cdots|C}_{n_1+1}|L_1|\underbrace{C|\cdots|C}_{n_2}|L_2|$$

Encoding

$$\forall x, y, z.\left(\begin{array}{l}([\![Q_j]\!]^x \frown [\![C]\!]^y \frown [\![L_1]\!]^x \frown [\![C]\!]^z \frown [\![L_2]\!]^x \frown (\ell = 4x + y + z))\\ \Rightarrow \\ (\ell = 3x + y + z) \frown [\![Q_k]\!]^x \frown [\![C]\!]^x \frown [\![C]\!]^y \frown [\![L_1]\!]^x \frown [\![C]\!]^z \frown [\![L_2]\!]^x\end{array}\right.$$

- undecidability of RDC_3