02917 Advanced Topics in Embedded Systems

Model Checking for Duration Calculus
using Presburger Arithmetic

Michael R. Hansen

DTU Informatics
Department of Informatics and Mathematical Modelling

Overview

- Overview of fundamental (un)decidability results
- A basic decidability results
  – with non-elementary complexity
- Towards efficient model checking based on approximations
  Using Presburger Arithmetic: first-order logic of natural numbers with addition
- Decision procedure for Presburger Arithmetic

Overview

- Overview of fundamental (un)decidability results
- A basic decidability results
  – with non-elementary complexity
- Towards efficient model checking based on approximations
  Using Presburger Arithmetic: first-order logic of natural numbers with addition
- Decision procedure for Presburger Arithmetic

Overview

- Overview of fundamental (un)decidability results
- A basic decidability results
  – with non-elementary complexity
- Towards efficient model checking based on approximations
  Using Presburger Arithmetic: first-order logic of natural numbers with addition
- Decision procedure for Presburger Arithmetic

Overview

- Overview of fundamental (un)decidability results
- A basic decidability results
  – with non-elementary complexity
- Towards efficient model checking based on approximations
  Using Presburger Arithmetic: first-order logic of natural numbers with addition
- Decision procedure for Presburger Arithmetic

Overview

- Overview of fundamental (un)decidability results
- A basic decidability results
  – with non-elementary complexity
- Towards efficient model checking based on approximations
  Using Presburger Arithmetic: first-order logic of natural numbers with addition
- Decision procedure for Presburger Arithmetic

Zhou Hansen Sestoft 93

Restricted Duration Calculus:

- $\lceil S \rceil$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Decidable result for both discrete and continuous time

Seemingly small extensions give undecidable subsets

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

Basic Decidability Properties of Duration Calculus

Zhou Hansen Sestoft 93

Restricted Duration Calculus:

- $\lceil S \rceil$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Decidable result for both discrete and continuous time

Seemingly small extensions give undecidable subsets

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

Basic Decidability Properties of Duration Calculus

Zhou Hansen Sestoft 93

Restricted Duration Calculus:

- $\lceil S \rceil$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Decidable result for both discrete and continuous time 🙂

Seemingly small extensions give undecidable subsets 🙁

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

Basic Decidability Properties of Duration Calculus

Zhou Hansen Sestoft 93

Restricted Duration Calculus:

- $\lceil S \rceil$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Decidable result for both discrete and continuous time

Seemingly small extensions give undecidable subsets

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

Basic Decidability Properties of Duration Calculus

Zhou Hansen Sestoft 93

Restricted Duration Calculus:

- $\lceil S \rceil$
- $\neg\phi,\ \phi \vee \psi,\ \phi \frown \psi$

Satisfiability is reduced to emptiness of regular languages

Decidable result for both discrete and continuous time

Seemingly small extensions give undecidable subsets

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

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.

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

- $\mathcal{L}(\phi)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\phi) \neq \emptyset$
- Satisfiability problem for *RDC* is decidable

non-elementary complexity

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.

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

- $\mathcal{L}(\phi)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\phi) \neq \emptyset$
- Satisfiability problem for *RDC* is decidable

non-elementary complexity

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.

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

- $\mathcal{L}(\phi)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\phi) \neq \emptyset$
- Satisfiability problem for *RDC* is decidable

non-elementary complexity

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.

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

- $\mathcal{L}(\phi)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\phi) \neq \emptyset$
- Satisfiability problem for *RDC* is decidable

non-elementary complexity

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.

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

- $\mathcal{L}(\phi)$ is regular
- ϕ is satisfiable iff $\mathcal{L}(\phi) \neq \emptyset$
- Satisfiability problem for *RDC* is decidable

non-elementary complexity 🙁

Status

- A DC fragment based on (propositional logic and chop) with almost no notion of duration has non-elementary complexity. No existing tool is used on a daily basis.

- Fragments (propositional logic and chop) having simple notions of duration are undecidable.

So what about tool support?

Status

- A DC fragment based on (propositional logic and chop) with almost no notion of duration has non-elementary complexity. No existing tool is used on a daily basis.

- Fragments (propositional logic and chop) having simple notions of duration are undecidable.

So what about tool support?

Status

- A DC fragment based on (propositional logic and chop) with almost no notion of duration has non-elementary complexity. No existing tool is used on a daily basis.
- Fragments (propositional logic and chop) having simple notions of duration are undecidable.

So what about tool support?

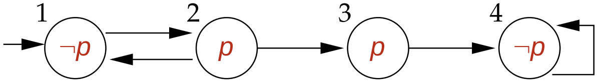
An example

Given Kripke structure K and a (certain kind of) DC formula ϕ.

- Does $K \models \phi$ hold?

every trace tr satisfies ϕ
non-elementary

Example: A simple Kripke structure K:



Problem: $K \models \Box(\ell < 4 \Rightarrow \int p < 3)$? YES

- Example run:

$$tr = (1:\neg p)\,(2:p)\,(1:\neg p)\,(2:p)\,(1:\neg p)\,(2:p)\,(3:\_)$$

satisfies $\Box(\ell < 4 \Rightarrow \int p < 3)$

Branching-time approximations for efficient verification
FränzleHansen 08,09

An example

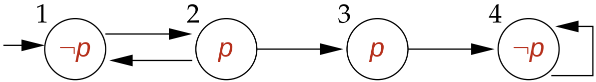
Given Kripke structure K and a (certain kind of) DC formula ϕ.

- Does $K \models \phi$ hold?

every trace tr satisfies ϕ

non-elementary

Example: A simple Kripke structure K:



Problem: $K \models \Box(\ell < 4 \Rightarrow \int p < 3)$?

YES

- Example run:

$$tr = (1 : \neg p)\ (2 : p)\ (1 : \neg p)\ (2 : p)\ (1 : \neg p)\ (2 : p)\ (3 : \_)$$

satisfies $\Box(\ell < 4 \Rightarrow \int p < 3)$

Branching-time approximations for efficient verification

FränzleHansen 08,09

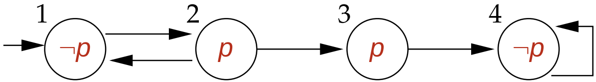
An example

Given Kripke structure K and a (certain kind of) DC formula ϕ.

- Does $K \models \phi$ hold?

every trace tr satisfies ϕ
non-elementary

Example: A simple Kripke structure K:



Problem: $K \models \Box(\ell < 4 \Rightarrow \int p < 3)$? YES

- Example run:

$$tr = (1 : \neg p)\,(2 : p)\,(1 : \neg p)\,(2 : p)\,(1 : \neg p)\,(2 : p)\,(3 : \_)$$

satisfies $\Box(\ell < 4 \Rightarrow \int p < 3)$

Branching-time approximations for efficient verification
FränzleHansen 08,09

An example

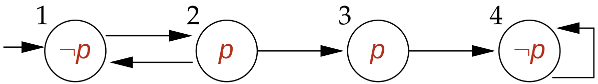
Given Kripke structure K and a (certain kind of) DC formula ϕ.

- Does $K \models \phi$ hold?

every trace tr satisfies ϕ

non-elementary

Example: A simple Kripke structure K:



Problem: $K \models \Box(\ell < 4 \Rightarrow \int p < 3)$? YES

- Example run:

$$tr = (1 : \neg p)\,(2 : p)\,(1 : \neg p)\,(2 : p)\,(1 : \neg p)\,(2 : p)\,(3 : \_)$$

satisfies $\Box(\ell < 4 \Rightarrow \int p < 3)$

Branching-time approximations for efficient verification

FränzleHansen 08,09

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N} \qquad \textit{a multiset}$$

The *counting semantics:* $K[\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}, \mathrm{false}\}$ if K is not consistent with m, i, j
- $K[\![\phi]\!]_c\, i\, j\, m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N}$$ *a multiset*

The *counting semantics:* $K[\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K[\![\phi]\!]_c\ i\ j\ m = \{\mathrm{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\ i\ j\ m = \{\mathrm{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\ i\ j\ m = \{\mathrm{true}, \mathrm{false}\}$ if K is not consistent with m, i, j
- $K[\![\phi]\!]_c\ i\ j\ m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N}$ *a multiset*

The *counting semantics:* $K [\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K [\![\phi]\!]_c\ i\ j\ m = \{\text{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K [\![\phi]\!]_c\ i\ j\ m = \{\text{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K [\![\phi]\!]_c\ i\ j\ m = \{\text{true}, \text{false}\}$ if K is not consistent with m, i, j
- $K [\![\phi]\!]_c\ i\ j\ m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N}$ *a multiset*

The *counting semantics:* $K[\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}, \mathrm{false}\}$ if K is not consistent with m, i, j
- $K[\![\phi]\!]_c\, i\, j\, m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N} \qquad\qquad \textit{a multiset}$$

The *counting semantics:* $K[\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}, \mathrm{false}\}$ if K is not consistent with m, i, j
- $K[\![\phi]\!]_c\, i\, j\, m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N}$$ *a multiset*

The *counting semantics:* $K[\![\phi]\!]_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K[\![\phi]\!]_c\, i\, j\, m = \{\mathrm{true}, \mathrm{false}\}$ if K is not consistent with m, i, j
- $K[\![\phi]\!]_c\, i\, j\, m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Branching-time approximations: Counting semantics

Ideas:

- All traces between two vertices are treated uniformly
- Add information on the frequency of visits to vertices.

$$m : \mathrm{Mset} = V \xrightarrow{\text{part}} \mathbb{N} \qquad\qquad \textit{a multiset}$$

The *counting semantics:* $K\llbracket\phi\rrbracket_c : V \to V \to \mathrm{Mset} \to 2^{\mathbb{B}}$:

- $K\llbracket\phi\rrbracket_c\, i\, j\, m = \{\text{true}\}$ when $tr \models_K \phi$, for any run tr from i to j which is *consistent* with m
- $K\llbracket\phi\rrbracket_c\, i\, j\, m = \{\text{false}\}$ when $tr \not\models_K \phi$, for any tr from i to j which is *consistent* with m
- $K\llbracket\phi\rrbracket_c\, i\, j\, m = \{\text{true}, \text{false}\}$ if K is not consistent with m, i, j
- $K\llbracket\phi\rrbracket_c\, i\, j\, m = \emptyset$ otherwise.

We aim at a symbolic treatment of multisets

Model checking: Main idea

Given K, ϕ and a vector $\overline{m} = dom\, m$ of variables.

We mark situation pairs (i, j), with $(\psi, b, lin(\overline{m}))$, $b \in \{\text{true}, \text{false}\}$, where ψ is a subformula of ϕ

- and $lin(\overline{m})$ is a side-condition (Presburger formula) with $\overline{m}$ as free variables.

Key properties for marking $(\psi, true, lin(\overline{m}))$:

- $$\text{true} \in K[\![\psi]\!]\, i\, j\, m \quad \text{for any } \overline{m} \text{ satisfying } lin(\overline{m})$$

There are similar properties for a marking $(\psi, false, lin(\overline{m}))$

Model checking: Main idea

Given K, ϕ and a vector $\overline{m} = \mathit{dom}\, m$ of variables.

We mark situation pairs (i, j), with $(\psi, b, \mathit{lin}(\overline{m}))$, $b \in \{\text{true}, \text{false}\}$, where ψ is a subformula of ϕ

- and $\mathit{lin}(\overline{m})$ is a side-condition (Presburger formula) with $\overline{m}$ as free variables.

Key properties for marking $(\psi, \mathit{true}, \mathit{lin}(\overline{m}))$:

- $$\text{true} \in K[\![\psi]\!]\, i\, j\, m \quad \text{for any } \overline{m} \text{ satisfying } \mathit{lin}(\overline{m})$$

There are similar properties for a marking $(\psi, \mathit{false}, \mathit{lin}(\overline{m}))$

Model checking: Main idea

Given K, ϕ and a vector $\overline{m} = dom\, m$ of variables.

We mark situation pairs (i, j), with $(\psi, b, lin(\overline{m}))$, $b \in \{\text{true}, \text{false}\}$, where ψ is a subformula of ϕ

- and $lin(\overline{m})$ is a side-condition (Presburger formula) with $\overline{m}$ as free variables.

Key properties for marking $(\psi, true, lin(\overline{m}))$:

- $$\text{true} \in K[\![\psi]\!]\, i\, j\, m \quad \text{for any } \overline{m} \text{ satisfying } lin(\overline{m})$$

There are similar properties for a marking $(\psi, false, lin(\overline{m}))$

Model checking: Main idea

Given K, ϕ and a vector $\overline{m} = \mathit{dom}\, m$ of variables.

We mark situation pairs (i, j), with $(\psi, b, \mathit{lin}(\overline{m}))$, $b \in \{\text{true}, \text{false}\}$, where ψ is a subformula of ϕ

- and $\mathit{lin}(\overline{m})$ is a side-condition (Presburger formula) with $\overline{m}$ as free variables.

Key properties for marking $(\psi, \mathit{true}, \mathit{lin}(\overline{m}))$:

- $$\text{true} \in K[\![\psi]\!]\, i\, j\, m \quad \text{for any } \overline{m} \text{ satisfying } \mathit{lin}(\overline{m})$$

There are similar properties for a marking $(\psi, \mathit{false}, \mathit{lin}(\overline{m}))$

A Condition for Consistency

$C(K, i_0, j_0, \overline{m})$ is a system of linear equations:

m-consistency wrt. K, i_0 and j_0
is equivalent to satisfiability of $C(K, i_0, j_0, \overline{m})$

Variables:

- x_i for every $i \in V$,
- x_{ij} for every edge $(i, j) \in E$
- $\overline{m}[k]$ for every $k \in \mathrm{dom}\ m$

Main ideas:

- The "inflow" is the same as the "outflow" for any vertex k.
- i_0 has an extra inflow of 1 and j_0 has an extra outflow of 1.
- $x_k = \overline{m}[k]$ for every $k \in \mathrm{dom}\ m$.

For example, for every $k \in V \setminus \{i_0, j_0\}$:

$$\Sigma_{(l,k)\in E}\, x_{lk} = x_k \quad \text{and} \quad x_k = \Sigma_{(k,l)\in E}\, x_{kl}$$

A Condition for Consistency

$C(K, i_0, j_0, \overline{m})$ is a system of linear equations:

m-consistency wrt. K, i_0 and j_0
is equivalent to satisfiability of $C(K, i_0, j_0, \overline{m})$

Variables:

- x_i for every $i \in V$,
- x_{ij} for every edge $(i, j) \in E$
- $\overline{m}[k]$ for every $k \in \mathrm{dom}\ m$

Main ideas:

- The "inflow" is the same as the "outflow" for any vertex k.
- i_0 has an extra inflow of 1 and j_0 has an extra outflow of 1.
- $x_k = \overline{m}[k]$ for every $k \in \mathrm{dom}\ m$.

For example, for every $k \in V \setminus \{i_0, j_0\}$:

$$\Sigma_{(l,k)\in E}\, x_{lk} = x_k \quad \text{and} \quad x_k = \Sigma_{(k,l)\in E}\, x_{kl}$$

A Condition for Consistency

$C(K, i_0, j_0, \overline{m})$ is a system of linear equations:

m-consistency wrt. K, i_0 and j_0
is equivalent to satisfiability of $C(K, i_0, j_0, \overline{m})$

Variables:

- x_i for every $i \in V$,
- x_{ij} for every edge $(i, j) \in E$
- $\overline{m}[k]$ for every $k \in \mathrm{dom}\ m$

Main ideas:

- The "inflow" is the same as the "outflow" for any vertex k.
- i_0 has an extra inflow of 1 and j_0 has an extra outflow of 1.
- $x_k = \overline{m}[k]$ for every $k \in \mathrm{dom}\ m$.

For example, for every $k \in V \setminus \{i_0, j_0\}$:

$$\Sigma_{(l,k)\in E}\, x_{lk} = x_k \quad \text{and} \quad x_k = \Sigma_{(k,l)\in E}\, x_{kl}$$

A Condition for Consistency

$C(K, i_0, j_0, \overline{m})$ is a system of linear equations:

m-consistency wrt. K, i_0 and j_0
is equivalent to satisfiability of $C(K, i_0, j_0, \overline{m})$

Variables:

- x_i for every $i \in V$,
- x_{ij} for every edge $(i, j) \in E$
- $\overline{m}[k]$ for every $k \in \mathrm{dom}\ m$

Main ideas:

- The "inflow" is the same as the "outflow" for any vertex k.
- i_0 has an extra inflow of 1 and j_0 has an extra outflow of 1.
- $x_k = \overline{m}[k]$ for every $k \in \mathrm{dom}\ m$.

For example, for every $k \in V \setminus \{i_0, j_0\}$:

$$\Sigma_{(l,k)\in E}\, x_{lk} \quad = \quad x_k \quad \text{and} \quad x_k \quad = \quad \Sigma_{(k,l)\in E}\, x_{kl}$$

Model checking: Case $\psi = \int S < k$

Markings for (i, j):

- $\left(\int S < k, \text{true}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] < k\right)\right)$
- $\left(\int S < k, \text{false}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] \geq k\right)\right)$

This is easily generalized to formulas of the form:

$$\Sigma_{i=1}^{n} c_i \int S_i \lhd k$$

where $\lhd \in \{<, \leq, =, \geq, >\}$.

Model checking: Case $\psi = \int S < k$

Markings for (i, j):

- $\left(\int S < k, \text{true}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] < k\right)\right)$
- $\left(\int S < k, \text{false}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] \geq k\right)\right)$

This is easily generalized to formulas of the form:

$$\Sigma_{i=1}^{n} c_i \int S_i \lhd k$$

where $\lhd \in \{<, \leq, =, \geq, >\}$.

Model checking: Case $\psi = \int S < k$

Markings for (i, j):

- $\left(\int S < k, \text{true}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] < k\right)\right)$
- $\left(\int S < k, \text{false}, \left(C(K, i, j, \overline{m}, \overline{e}) \wedge \sum_{v \in \mathrm{dom}\, m, v \models S} m[v] \geq k\right)\right)$

This is easily generalized to formulas of the form:

$$\Sigma_{i=1}^{n} c_i \int S_i \lhd k$$

where $\lhd \in \{<, \leq, =, \geq, >\}$.

Model checking: Case $\phi = \psi_1 \wedge \psi_2$

Markings for (i, j):

- The true marking is $(\psi_1 \wedge \psi_2, \text{true}, \mu \wedge \nu)$ iff (i, j) is marked with $(\psi_1, \text{true}, \mu)$ and $(\psi_2, \text{true}, \nu)$
- The false marking is $(\psi_1 \wedge \psi_2, \text{false}, \mu \vee \nu)$ iff (i, j) is marked with $(\psi_1, \text{false}, \mu)$ and $(\psi_2, \text{false}, \nu)$

Model checking: Case $\phi = \psi_1 \wedge \psi_2$

Markings for (i,j):

- The true marking is $(\psi_1 \wedge \psi_2, \text{true}, \mu \wedge \nu)$ iff (i,j) is marked with $(\psi_1, \text{true}, \mu)$ and $(\psi_2, \text{true}, \nu)$
- The false marking is $(\psi_1 \wedge \psi_2, \text{false}, \mu \vee \nu)$ iff (i,j) is marked with $(\psi_1, \text{false}, \mu)$ and $(\psi_2, \text{false}, \nu)$

Model checking: Case $\phi = \neg\psi$

Markings for (i, j):

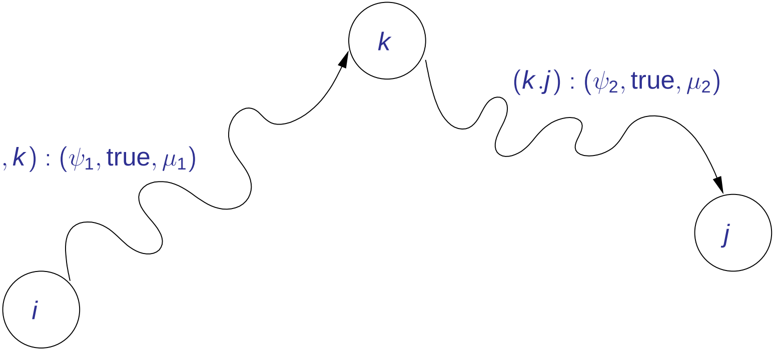
- The true marking is $(\neg\psi, \text{true}, \mu)$ iff (i, j) is marked with $(\psi, \text{false}, \mu)$
- The false marking is $(\neg\psi, \text{false}, \nu)$ iff (i, j) is marked with (ψ, true, ν)

Model checking: Case $\phi = \neg\psi$

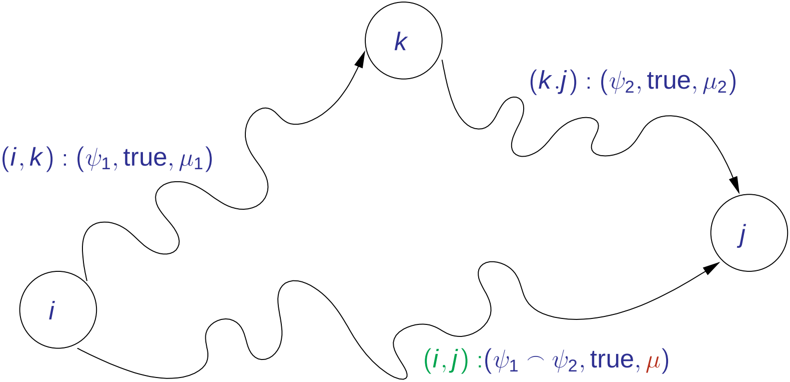
Markings for (i, j):

- The true marking is $(\neg\psi, \text{true}, \mu)$ iff (i, j) is marked with $(\psi, \text{false}, \mu)$
- The false marking is $(\neg\psi, \text{false}, \nu)$ iff (i, j) is marked with (ψ, true, ν)

Case $\psi = \psi_1 \frown \psi_2$ — true marking



Case $\psi = \psi_1 \frown \psi_2$ true marking

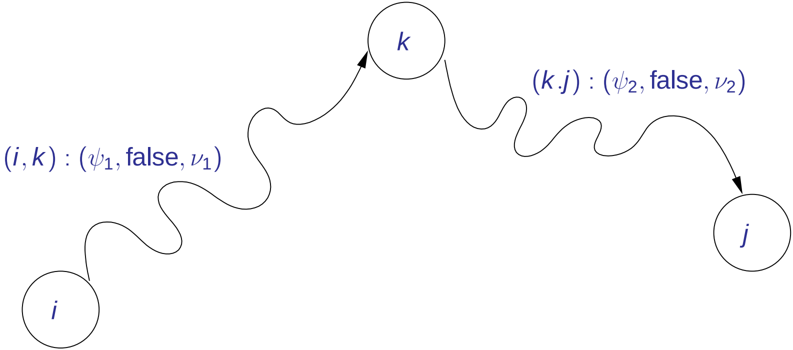


where μ is

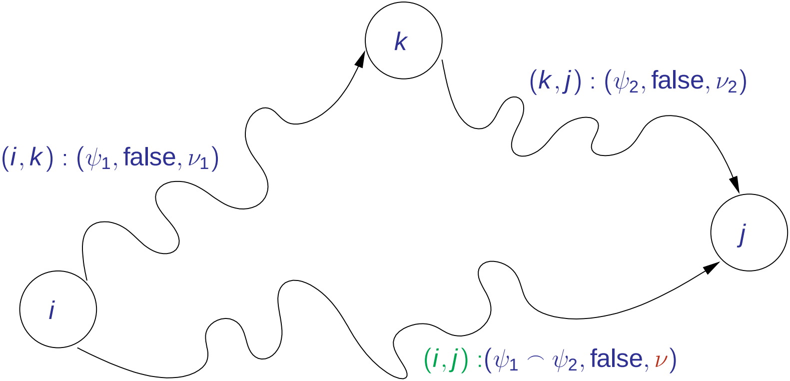
$$\bigvee_{k \in V} \left\{ \begin{array}{ll} & \exists \overline{m}_1, \overline{m}_2 : Split(k, \overline{m}, \overline{m}_1, \overline{m}_2) \\ \wedge & \forall \overline{m}_1, \overline{m}_2 : Split(k, \overline{m}, \overline{m}_1, \overline{m}_2) \Rightarrow (\mu_1[\overline{m}_1/\overline{m}] \wedge \mu_2[\overline{m}_2/\overline{m}]) \end{array} \right\}$$

and $Split(k, \overline{m}, \overline{m}_1, \overline{m2})$ is $\overline{m} = \overline{m}_1 + \overline{m}_2 \wedge C(i, k, \overline{m}_1) \wedge C(k, j, \overline{m}_2)$

Case $\psi = \psi_1 \frown \psi_2$ false marking



Case $\psi = \psi_1 \frown \psi_2$ false marking



where ν is

$$C(i, j, \overline{m}) \wedge \bigwedge_{k \in V} \forall \overline{m}_1, \overline{m}_2 : (Split(k, \overline{m}, \overline{m}_1, \overline{m}_2) \Rightarrow \nu_1[\overline{m}_1/\overline{m}] \vee \nu_2[\overline{m}_2/\overline{m}])$$

Model checking 2: Example. Simplified markings

For $\mathrm{dom}\ m = \{1, 2, 4\}$ and $\zeta = \int \text{true} < 4 \wedge \neg \int p < 3$.

Notice $\Box(\int \text{true} < 4 \Rightarrow \int p < 3) \iff \neg \Diamond \zeta$.

i, j	$C(m)$ after simplification	markings $(\psi, \text{false}, \eta)$ for $\psi =$ $\int \text{true} < 4$	$\neg \int p < 3$	ζ	$\Diamond \zeta$
$1, 1$	$m[1] = m[2]$	$m[1] > 2$	$m[1] < 3$	true	true
$1, 2$	$m[1] = m[2] + 1$	$m[1] > 2$	$m[1] < 3$	true	true
$1, 3$	$m[1] = m[2] > 0$	$m[1] > 1$	$m[1] < 3$	true	true
$1, 4$	$m[1] = m[2] > 0$	$m[1] > 1 \vee m[4] > 0$	$m[1] \leq 1$	true	true
$2, \{1, 3\}$	$m[2] = m[1] + 1$	$m[1] > 1$	$m[1] < 2$	true	true
		$\vdots$			
$3, \{1, 2\}$	false	true	true	true	true
$3, 3$	true	false	true	true	true
		$\vdots$			

Model checking 2: Example. Simplified markings

For $\mathrm{dom}\ m = \{1,2,4\}$ and $\zeta = \int \mathrm{true} < 4 \land \neg \int p < 3$.

Notice $\Box(\int \mathrm{true} < 4 \Rightarrow \int p < 3) \iff \neg \Diamond \zeta$.

i, j	$C(m)$ after simplification	markings $(\psi, \mathrm{false}, \eta)$ for $\psi =$ $\int \mathrm{true} < 4$	$\neg \int p < 3$	ζ	$\Diamond \zeta$
$1,1$	$m[1] = m[2]$	$m[1] > 2$	$m[1] < 3$	true	true
$1,2$	$m[1] = m[2] + 1$	$m[1] > 2$	$m[1] < 3$	true	true
$1,3$	$m[1] = m[2] > 0$	$m[1] > 1$	$m[1] < 3$	true	true
$1,4$	$m[1] = m[2] > 0$	$m[1] > 1 \lor m[4] > 0$	$m[1] \leq 1$	true	true
$2,\{1,3\}$	$m[2] = m[1] + 1$	$m[1] > 1$	$m[1] < 2$	true	true
		$\vdots$			
$3,\{1,2\}$	false	true	true	true	true
$3,3$	true	false	true	true	true
		$\vdots$			

Summary

- Algorithm is correct.
- Procedure is 4-fold exponential.
  - Size of generated formula is exponential in the chop-depth.
  - Presburger formulas are checked in triple-exponential time.

Remember undecidable (non-elementary) starting point

- Preciseness when all chops is under same polarity and all conjunctions under the dual polarity.
- Quantifier elimination of side-condition is possible when all chops are in negative polarity. Procedure is then "just" 2-fold exponential.
- Prototype is implemented by William Pihl Heise in a using the solver Z3 as backend. The prototype has just been used on small examples. Algorithm seems promising.

Summary

- Algorithm is correct.
- Procedure is 4-fold exponential.
  - Size of generated formula is exponential in the chop-depth.
  - Presburger formulas are checked in triple-exponential time.

Remember undecidable (non-elementary) starting point

- Preciseness when all chops is under same polarity and all conjunctions under the dual polarity.
- Quantifier elimination of side-condition is possible when all chops are in negative polarity. Procedure is then ”just” 2-fold exponential.
- Prototype is implemented by William Pihl Heise in a using the solver Z3 as backend. The prototype has just been used on small examples. Algorithm seems promising.

Summary

- Algorithm is correct.
- Procedure is 4-fold exponential.
  - Size of generated formula is exponential in the chop-depth.
  - Presburger formulas are checked in triple-exponential time.

Remember undecidable (non-elementary) starting point

- Preciseness when all chops is under same polarity and all conjunctions under the dual polarity.
- Quantifier elimination of side-condition is possible when all chops are in negative polarity. Procedure is then ”just” 2-fold exponential.
- Prototype is implemented by William Pihl Heise in a using the solver Z3 as backend. The prototype has just been used on small examples. Algorithm seems promising.

Summary

- Algorithm is correct.
- Procedure is 4-fold exponential.
  - Size of generated formula is exponential in the chop-depth.
  - Presburger formulas are checked in triple-exponential time.

Remember undecidable (non-elementary) starting point

- Preciseness when all chops is under same polarity and all conjunctions under the dual polarity.
- Quantifier elimination of side-condition is possible when all chops are in negative polarity. Procedure is then ”just” 2-fold exponential.
- Prototype is implemented by William Pihl Heise in a using the solver Z3 as backend. The prototype has just been used on small examples. Algorithm seems promising.

Summary

- Algorithm is correct.
- Procedure is 4-fold exponential.
  - Size of generated formula is exponential in the chop-depth.
  - Presburger formulas are checked in triple-exponential time.

Remember undecidable (non-elementary) starting point

- Preciseness when all chops is under same polarity and all conjunctions under the dual polarity.
- Quantifier elimination of side-condition is possible when all chops are in negative polarity. Procedure is then ”just” 2-fold exponential.
- Prototype is implemented by William Pihl Heise in a using the solver Z3 as backend. The prototype has just been used on small examples. Algorithm seems promising.