Advanced Logic

http://www-sop.inria.fr/members/Martin.Avanzini/teaching/2021/AL/

Martin Avanzini

Summer Semester 2021

Last Lecture

1. the set of WMSO formulas over $\mathcal{V}_1, \mathcal{V}_2$ is given by the following grammar:

$$\phi, \psi ::= \top \mid \bot \mid x < y \mid X(x) \mid \phi \vee \psi \mid \neg\phi \mid \exists x.\phi \mid \exists X.\phi$$

   – first-order variables $\mathcal{V}_1$ range over $\mathbb{N}$ and second-order variables $\mathcal{V}_2$ range over finite sets over $\mathbb{N}$

2. a WMSO formula ϕ over second-order variables $\{P_\mathsf{a} \mid \mathsf{a} \in \Sigma\}$ defines a language

$$L(\phi) \triangleq \{w \in \Sigma^* \mid \underline{w} \vDash \phi\}$$

3. WMSO definable languages are regular, and vice verse

4. Satisfiability and validity decidable in $2^{2^{\cdot^{\cdot^{\cdot^{2^c}}}}}$, the height of this tower essentially depends on quantifiers; this bound cannot be improved

   – in practice, satisfiability/validity often feasible, even for bigger formulas

Today's Lecture

- ⋆ Presburger arithmetic
- ⋆ the tool `MONA`

Presburger Arithmetic

Presburger Arithmetic

- **Presburger Arithmetic** refers to the first-order theory over $(\mathbb{N}, \{0, +, <\})$
- named in honor of Mojżesz Presburger, who introduced it in 1929
- formulas in this logic are derivable from the following grammar:

$$s, t ::= 0 \mid x \mid s + t$$

$$\phi, \psi ::= \top \mid \bot \mid s = t \mid s < t \mid \phi \wedge \psi \mid \neg\psi \mid \exists x.\phi$$

  where x is a first-order variable
- valuations map first-order variables to $\mathbb{N}$

Presburger Arithmetic

- Presburger Arithmetic refers to the first-order theory over $(\mathbb{N}, \{0, +, <\})$
- named in honor of Mojżesz Presburger, who introduced it in 1929
- formulas in this logic are derivable from the following grammar:

$$s, t ::= 0 \mid x \mid s + t$$

$$\phi, \psi ::= \top \mid \bot \mid s = t \mid s < t \mid \phi \wedge \psi \mid \neg\psi \mid \exists x.\phi$$

  where x is a first-order variable
- valuations map first-order variables to $\mathbb{N}$

Applications

Presburger Arithmetic employed — due to the balance between expressiveness and algorithmic properties — e.g. in automated theorem proving and static program analysis

Examples

- ⋆ m is even: ?

Examples

- m is even: $\exists n.m = n + n$, or shorthand $\exists n.m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible

Examples

- m is even: $\exists n.m = n + n$, or shorthand $\exists n.m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- m equals 1: ?

Examples

- ⋆ m is even: $\exists n. m = n + n$, or shorthand $\exists n. m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- ⋆ m equals 1: $\forall n. n < m \to n = 0$

Examples

- m is even: $\exists n. m = n + n$, or shorthand $\exists n. m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- m equals 1: $\forall n. n < m \rightarrow n = 0$
- $m = r \bmod 5$: ?

Examples

- m is even: $\exists n.m = n + n$, or shorthand $\exists n.m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- m equals 1: $\forall n.n < m \rightarrow n = 0$
- $m = r \bmod 5$: $\exists n.r < 5 \wedge m = 5 \cdot n + r$

Examples

- m is even: $\exists n. m = n + n$, or shorthand $\exists n. m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- m equals 1: $\forall n. n < m \rightarrow n = 0$
- $m = r \bmod 5$: $\exists n. r < 5 \wedge m = 5 \cdot n + r$
- the system of linear equations

$$
\begin{aligned}
m + n &= 13 \\
m - n &= 1
\end{aligned}
$$

has a solution: ?

Examples

- m is even: $\exists n.m = n + n$, or shorthand $\exists n.m = 2 \cdot n$
  - generally, multiplication by constant $c \in \mathbb{N}$ permissible
- m equals 1: $\forall n.n < m \rightarrow n = 0$
- $m = r \bmod 5$: $\exists n.r < 5 \wedge m = 5 \cdot n + r$
- the system of linear equations

$$\begin{aligned} m + n &= 13 \\ m - n &= 1 \end{aligned}$$

  has a solution: $\exists m.\exists n.m + n = 13 \wedge m = 1 + n$

A Decision Procedure for Presburger Arithmetic

General Idea

1. encode natural numbers as binary words (lsb-first order)
   - assignments $\alpha : \mathcal{V} \to \{0, \ldots, 2^m\}$ over $\{x_1, \ldots, x_n\}$ become binary matrices $\underline{\alpha} \in \{0,1\}^{(m,n)}$

	$\alpha(x_i)$	$\underline{\alpha}$
x_1	7	
x_2	1	$\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$
x_3	3	

A Decision Procedure for Presburger Arithmetic

General Idea

1. encode natural numbers as binary words (lsb-first order)
   - assignments $\alpha : \mathcal{V} \to \{0, \ldots, 2^m\}$ over $\{x_1, \ldots, x_n\}$ become binary matrices $\underline{\alpha} \in \{0,1\}^{(m,n)}$

	$\alpha(x_i)$	$\underline{\alpha}$
x_1	7	
x_2	1	$\begin{pmatrix}1\\1\\1\end{pmatrix}\begin{pmatrix}0\\0\\1\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}\begin{pmatrix}1\\0\\0\end{pmatrix}$
x_3	3	

2. for formula ϕ, define a DFA $\mathcal{A}_\phi$ recognizing precisely codings $\underline{\alpha}$ of valuations α making ϕ become true

Language of a Formula

let us denote by $\hat{\mathsf{L}}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{\mathsf{L}}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \models \phi\}$$

Language of a Formula

let us denote by $\hat{L}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{L}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \models \phi\}$$

Lemma

For any formula ϕ in Presburger Arithmetic, $\hat{L}(\phi)$ is regular.

Language of a Formula

let us denote by $\hat{\mathsf{L}}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{\mathsf{L}}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \models \phi\}$$

Lemma

For any formula ϕ in Presburger Arithmetic, $\hat{\mathsf{L}}(\phi)$ is regular.

Proof Outline.

By induction on the structure of ϕ, we construct a DFA $\mathcal{A}_\phi$ recognizing $\hat{\mathsf{L}}(\phi)$.

Language of a Formula

let us denote by $\hat{\mathsf{L}}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{\mathsf{L}}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \vDash \phi\}$$

Lemma

For any formula ϕ in Presburger Arithmetic, $\hat{\mathsf{L}}(\phi)$ is regular.

Proof Outline.

By induction on the structure of ϕ, we construct a DFA $\mathcal{A}_\phi$ recognizing $\hat{\mathsf{L}}(\phi)$.

- $\phi = \top$, $\phi = \bot$: In these cases $\hat{\mathsf{L}}(\phi)$ is easily seen to be regular.
- $\phi = (s < t)$ or $\phi = (s = t)$: A corresponding automaton can be constructed (next slide).

Language of a Formula

let us denote by $\hat{L}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{L}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \vDash \phi\}$$

Lemma

For any formula ϕ in Presburger Arithmetic, $\hat{L}(\phi)$ is regular.

Proof Outline.

By induction on the structure of ϕ, we construct a DFA $\mathcal{A}_\phi$ recognizing $\hat{L}(\phi)$.

- ⋆ $\phi = \top$, $\phi = \bot$: In these cases $\hat{L}(\phi)$ is easily seen to be regular.
- ⋆ $\phi = (s < t)$ or $\phi = (s = t)$: A corresponding automaton can be constructed (next slide).
- ⋆ $\phi = \neg\phi$ or $\phi = \psi_1 \wedge \psi_2$ From the induction hypothesis, using DFA-complementation and DFA-union.

Language of a Formula

let us denote by $\hat{\mathsf{L}}(\phi)$ the language of coded valuations making ϕ true:

$$\hat{\mathsf{L}}(\phi) \triangleq \{\underline{\alpha} \mid \alpha \models \phi\}$$

Lemma

For any formula ϕ in Presburger Arithmetic, $\hat{\mathsf{L}}(\phi)$ is regular.

Proof Outline.

By induction on the structure of ϕ, we construct a DFA $\mathcal{A}_\phi$ recognizing $\hat{\mathsf{L}}(\phi)$.

- ⋆ $\phi = \top$, $\phi = \bot$: In these cases $\hat{\mathsf{L}}(\phi)$ is easily seen to be regular.
- ⋆ $\phi = (s < t)$ or $\phi = (s = t)$: A corresponding automaton can be constructed (next slide).
- ⋆ $\phi = \neg\phi$ or $\phi = \psi_1 \wedge \psi_2$ From the induction hypothesis, using DFA-complementation and DFA-union.
- ⋆ $\phi = \forall x.\psi$: From induction hypothesis, using homomorphism application to project out x and “repairing final states”, as in the case of WMSO.

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

- the automaton $\mathcal{A}_{s \,\leq\, t}$ recognizing $s \leq t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i \leq d$
    Intuition: $\mathrm{L}(\sum_i a_i \cdot x_i \leq d, \mathcal{A}_{s \,\leq\, t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i \leq d\}$

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

- the automaton $\mathcal{A}_{s \leq t}$ recognizing $s \leq t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i \leq d$
    Intuition: $L(\sum_i a_i \cdot x_i \leq d, \mathcal{A}_{s \leq t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i \leq d\}$
  - the initial state q_I is given by the representation of $s \leq t$

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

- the automaton $\mathcal{A}_{s \leq t}$ recognizing $s \leq t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i \leq d$
    Intuition: $\mathrm{L}(\sum_i a_i \cdot x_i \leq d, \mathcal{A}_{s \leq t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i \leq d\}$
  - the initial state q_I is given by the representation of $s \leq t$
  - the transition function δ is given by

$$\delta\left(\sum_i a_i \cdot x_i \leq d, \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}\right) \triangleq \sum_i a_i \cdot x_i \leq \left\lfloor \frac{1}{2}\left(d - \sum_i a_i \cdot b_i\right) \right\rfloor$$

since $\sum_i a_i \cdot (b_i + 2 \cdot x_i') \leq d \iff \sum_i a_i \cdot x_i' \leq \frac{1}{2} \cdot (d - \sum_i a_i \cdot b_i)$

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

- the automaton $\mathcal{A}_{s \leq t}$ recognizing $s \leq t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i \leq d$
    Intuition: $\mathrm{L}(\sum_i a_i \cdot x_i \leq d, \mathcal{A}_{s \leq t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i \leq d\}$
  - the initial state q_I is given by the representation of $s \leq t$
  - the transition function δ is given by

$$\delta\left(\sum_i a_i \cdot x_i \leq d, \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}\right) \triangleq \sum_i a_i \cdot x_i \leq \left\lfloor \frac{1}{2}\left(d - \sum_i a_i \cdot b_i\right) \right\rfloor$$

    since $\sum_i a_i \cdot (b_i + 2 \cdot x_i') \leq d \iff \sum_i a_i \cdot x_i' \leq \frac{1}{2} \cdot (d - \sum_i a_i \cdot b_i)$
  - final states are all those states $\sum_i a_i \cdot x_i \leq d$ with $0 \leq d$

Recognizing $s \leq t$

- an inequality $s \leq t$ can be represented as $\sum_i a_i \cdot x_i \leq b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 \leq x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 \leq 2$$

- the automaton $\mathcal{A}_{s \leq t}$ recognizing $s \leq t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i \leq d$
    Intuition: $\mathrm{L}(\sum_i a_i \cdot x_i \leq d, \mathcal{A}_{s \leq t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i \leq d\}$
  - the initial state q_I is given by the representation of $s \leq t$
  - the transition function δ is given by

$$\delta\left(\sum_i a_i \cdot x_i \leq d, \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}\right) \triangleq \sum_i a_i \cdot x_i \leq \left\lfloor \frac{1}{2}\left(d - \sum_i a_i \cdot b_i\right) \right\rfloor$$

    since $\sum_i a_i \cdot (b_i + 2 \cdot x_i') \leq d \iff \sum_i a_i \cdot x_i' \leq \frac{1}{2} \cdot (d - \sum_i a_i \cdot b_i)$
  - final states are all those states $\sum_i a_i \cdot x_i \leq d$ with $0 \leq d$

- finiteness: from initial state $\sum_i a_i \cdot x_i \leq d$, only $\sum_i a_i + d$ states reachable, hence the overall construction is finite

Recognizing $s < t$

- an inequality $s < t$ can be represented as $\sum_i a_i \cdot x_i < b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 < x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 < 2$$

- the automaton $\mathcal{A}_{s < t}$ recognizing $s < t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i < d$
    Intuition: $L(\sum_i a_i \cdot x_i < d, \mathcal{A}_{s < t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i < d\}$
  - the initial state q_I is given by the representation of $s < t$
  - the transition function δ is given by

$$\delta\left(\sum_i a_i \cdot x_i < d, \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}\right) \triangleq \sum_i a_i \cdot x_i < \left\lceil \frac{1}{2}\left(d - \sum_i a_i \cdot b_i\right)\right\rceil$$

    since $\sum_i a_i \cdot (b_i + 2 \cdot x_i^I) < d \iff \sum_i a_i \cdot x_i^I < \frac{1}{2} \cdot (d - \sum_i a_i \cdot b_i)$
  - final states are all those states $\sum_i a_i \cdot x_i < d$ with $0 < d$

- finiteness: from initial state $\sum_i a_i \cdot x_i < d$, only $\sum_i a_i + d$ states reachable, hence the overall construction is finite

Recognizing $s = t$

- an inequality $s = t$ can be represented as $\sum_i a_i \cdot x_i = b$ where $a_i, b \in \mathbb{Z}$

$$2 \cdot x_1 = x_2 + 2 \quad \Longrightarrow \quad 2 \cdot x_1 - 1 \cdot x_2 = 2$$

- the automaton $\mathcal{A}_{s\,=\,t}$ recognizing $s = t$ is defined as follows
  - states Q are inequalities of the form $\sum_i a_i \cdot x_i = d$ plus trap-state q_{fail}
    Intuition: $L(\sum_i a_i \cdot x_i = d, \mathcal{A}_{s\,=\,t}) = \{\underline{\alpha} \mid \alpha \vDash \sum_i a_i \cdot x_i = d\}$
  - the initial state q_I is given by the representation of $s = t$
  - the transition function δ is given by

$$\delta\left(\sum_i a_i \cdot x_i = d, \begin{pmatrix} b_1 \\ \vdots \\ b_n \end{pmatrix}\right) \triangleq \begin{cases} \sum_i a_i \cdot x_i = \frac{1}{2}\left(d - \sum_i a_i \cdot b_i\right) & \text{if } d - \sum_i a_i \cdot b_i \text{ even,} \\ q_{fail} & \text{otherwise.} \end{cases}$$

    since $\sum_i a_i \cdot (b_i + 2 \cdot x_i') = d \iff \sum_i a_i \cdot x_i' = \frac{1}{2} \cdot (d - \sum_i a_i \cdot b_i)$
  - final states are all those states $\sum_i a_i \cdot x_i = d$ with $0 = d$

- finiteness: from initial state $\sum_i a_i \cdot x_i = d$, only $\sum_i a_i + d$ states reachable, hence the overall construction is finite

Decision Problems for Presburger Arithmetic

The Satisfiability Problem

- Given: formula ϕ
- Question: is there α s.t $\alpha \models \phi$?

The Validity Problem

- Given: formula ϕ
- Question: $\alpha \models \phi$ for all assignments α?

Decision Problems for Presburger Arithmetic

The Satisfiability Problem

- Given: formula ϕ
- Question: is there α s.t $\alpha \models \phi$?

The Validity Problem

- Given: formula ϕ
- Question: $\alpha \models \phi$ for all assignments α?

Theorem

Satisfiability and Validity are decidable for Presburger Arithmetic.

Theorem

For any formula ϕ, the constructed DFA recognizing $\hat{L}(\phi)$ has size $O(2^{2^n})$.

Peano Arithmetic

- ⋆ Peano's arithmetic is the first-order theory natural integers with vocabulary $\{+, \times, <\}$

Peano Arithmetic

- Peano's arithmetic is the first-order theory natural integers with vocabulary $\{+, \times, <\}$
- its existential fragment corresponds to the Diophantine equations, i.e., polynomial equations on integers
- Hilbert's 10th problem was to solve Diophantine equations

Peano Arithmetic

- Peano's arithmetic is the first-order theory natural integers with vocabulary $\{+, \times, <\}$
- its existential fragment corresponds to the Diophantine equations, i.e., polynomial equations on integers
- Hilbert's 10th problem was to solve Diophantine equations
- Youri Matiassevitch, drawing on the work of Julia Robinson, demonstrated that this was an undecidable problem

Skolem Arithmetic

- Skolem's arithmetic is the first order theory natural integers with the vocabulary $\{\times, =\}$

Skolem Arithmetic

- Skolem's arithmetic is the first order theory natural integers with the vocabulary $\{\times, =\}$
- Skolem's arithmetic is also decidable
- proof goes via reduction to tree automata, closely resembling the proof we have just seen for Presburger's arithmetic

The tool MONA

The MONA Project

https://www.brics.dk/mona/index.html

★ MONA is a WMSO (and more) model checker

- determines validity of formula
- or prints counter example

★ implemented through the outlined translation to finite automata