# Lecture 14

## Decidable Theories

Logical theories, quantifier elimination, unbounded dense linear orders, linear arithmetic over the rationals, Presburger arithmetic

*Introduction to Logic for Computer Science*

Prof Hongseok Yang
KAIST

These slides are minor variants of those made by Prof Worrell and Dr Haase for their logic course at Oxford.

## Logical theories

A **theory** $\mathcal{T}$ is a set of sentences closed under semantic entailment:

$$\mathcal{T} \models F \text{ implies } F \in \mathcal{T}.$$

A theory is **complete** if either $F \in \mathcal{T}$ or $\neg F \in \mathcal{T}$ for any $F$.

## Logical theories

A **theory** $\mathcal{T}$ is a set of sentences closed under semantic entailment:

$$\mathcal{T} \models F \text{ implies } F \in \mathcal{T}.$$

A theory is **complete** if either $F \in \mathcal{T}$ or $\neg F \in \mathcal{T}$ for any $F$.

Two recipes for generating theories:

- Pick a $\sigma$-structure $\mathcal{A}$. Define

  $$\mathrm{Th}(\mathcal{A}) = \{F \,:\, \mathcal{A} \models F \text{ and } F \text{ is a sentence}\}.$$

  $\mathrm{Th}(\mathcal{A})$ is called the **theory of** $\mathcal{A}$.

- Pick a set of **axioms** $\mathcal{S}$ (i.e., a set of setences). Define

  $$\mathcal{T} = \{F \,:\, \mathcal{S} \models F \text{ and } F \text{ is a sentence}\}.$$

## Logical theories

A **theory** $\mathcal{T}$ is a set of sentences closed under semantic entailment:

$$\mathcal{T} \models F \text{ implies } F \in \mathcal{T}.$$

A theory is **complete** if either $F \in \mathcal{T}$ or $\neg F \in \mathcal{T}$ for any $F$.

Two recipes for generating theories:

- Pick a $\sigma$-structure $\mathcal{A}$. Define

  $$\mathrm{Th}(\mathcal{A}) = \{F : \mathcal{A} \models F \text{ and } F \text{ is a sentence}\}.$$

  $\mathrm{Th}(\mathcal{A})$ is called the **theory of** $\mathcal{A}$.

- Pick a set of **axioms** $\mathcal{S}$ (i.e., a set of setences). Define

  $$\mathcal{T} = \{F : \mathcal{S} \models F \text{ and } F \text{ is a sentence}\}.$$

Ex1: Prove that both recipes give theories.

Ex2: Which one always generates a complete theory?

Ex3: Give an example of an incomplete theory.

**Example (Structure-based Theory)**

The theory of **linear arithmetic over the rationals** is

$$\mathcal{T}_{LAR} = \mathrm{Th}(\mathbb{Q}, 1, +, \{c\,\cdot\}_{c\in\mathbb{Q}}, <).$$

It tells the truth of the following sentences:

- The system of linear inequalities $A\boldsymbol{x} \le \boldsymbol{b}$ has no solution.
- Every solution of $A\boldsymbol{x} \le \boldsymbol{b}$ is also a solution of $C\boldsymbol{x} \le \boldsymbol{d}$.

### Example (Structure-based Theory)

The theory of **linear arithmetic over the rationals** is

$$\mathcal{T}_{LAR} = \mathrm{Th}(\mathbb{Q}, 1, +, \{c\,\cdot\,\}_{c\in\mathbb{Q}}, <).$$

It tells the truth of the following sentences:

- The system of linear inequalities $A\boldsymbol{x} \leq \boldsymbol{b}$ has no solution.
- Every solution of $A\boldsymbol{x} \leq \boldsymbol{b}$ is also a solution of $C\boldsymbol{x} \leq \boldsymbol{d}$.

### Example (Axiom-based Theory)

The theory $\mathcal{T}_{UDLO}$ of **unbounded dense linear orders** is the set of sentences entailed by the following set of axioms:

$$\begin{array}{ll}
F_1 & \forall x\, \forall y\, (x < y \rightarrow \neg(x = y \vee y < x)) \\
F_2 & \forall x\, \forall y\, \forall z\, (x < y \wedge y < z \rightarrow x < z) \\
F_3 & \forall x\, \forall y\, (x < y \vee y < x \vee x = y) \\
F_4 & \forall x\, \forall y\, (x < y \rightarrow \exists z\, (x < z \wedge z < y)) \\
F_5 & \forall x\, \exists y\, \exists z\, (y < x < z)\,.
\end{array}$$

# Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

## Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

Ex: Do you think $\mathcal{T}_{LAR}$ is decidable? What about $\mathcal{T}_{UDLO}$?

## Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

Ex: Do you think $\mathcal{T}_{LAR}$ is decidable? What about $\mathcal{T}_{UDLO}$?

Ans: Both are decidable. Proved by **quantifier-elimination**.

## Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

Ex: Do you think $\mathcal{T}_{LAR}$ is decidable? What about $\mathcal{T}_{UDLO}$?

Ans: Both are decidable. Proved by **quantifier-elimination**.

A theory $\mathcal{T}$ **admits quantifier-elimination** if for any $\exists x\, F$ with $F$ quantifier-free, there is a quantifier-free formula $G$ such that

$$\mathcal{T} \models \exists x\, F \leftrightarrow G.$$

$\mathcal{T}$ has a **quantifier-elimination procedure** if $G$ is computable.

## Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

Ex: Do you think $\mathcal{T}_{LAR}$ is decidable? What about $\mathcal{T}_{UDLO}$?

Ans: Both are decidable. Proved by **quantifier-elimination**.

A theory $\mathcal{T}$ **admits quantifier-elimination** if for any $\exists x\, F$ with $F$ quantifier-free, there is a quantifier-free formula $G$ such that

$$\mathcal{T} \models \exists x\, F \leftrightarrow G.$$

$\mathcal{T}$ has a **quantifier-elimination procedure** if $G$ is computable.

**Theorem**

*A theory $\mathcal{T}$ is decidable if $\mathcal{T}$ has (i) a quantifier-elimination (QE) procedure, and (ii) a procedure for deciding $F \in \mathcal{T}$ for quantifier-free (QF) sentences $F$.*

## Decidable theories

A theory $\mathcal{T}$ is **decidable** if there is an algorithm for checking $F \in \mathcal{T}$ for every sentence $F$.

Ex: Do you think $\mathcal{T}_{LAR}$ is decidable? What about $\mathcal{T}_{UDLO}$?

Ans: Both are decidable. Proved by **quantifier-elimination**.

A theory $\mathcal{T}$ **admits quantifier-elimination** if for any $\exists x\, F$ with $F$ quantifier-free, there is a quantifier-free formula $G$ such that

$$\mathcal{T} \models \exists x\, F \leftrightarrow G.$$

$\mathcal{T}$ has a **quantifier-elimination procedure** if $G$ is computable.

**Theorem**

*A theory $\mathcal{T}$ is decidable if $\mathcal{T}$ has (i) a quantifier-elimination (QE) procedure, and (ii) a procedure for deciding $F \in \mathcal{T}$ for quantifier-free (QF) sentences $F$.*

Ex: Prove the theorem.

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

$\mathcal{T}_{UDLO}$ has (i) a QE proc., and (ii) a decision proc. for QF sentences.

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

$\mathcal{T}_{UDLO}$ has (i) a QE proc., and (ii) a decision proc. for QF sentences.

Ex1: Design a decision procedure for QF sentences.

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

$\mathcal{T}_{UDLO}$ has (i) a QE proc., and (ii) a decision proc. for QF sentences.

Ex1: Design a decision procedure for QF sentences.

Ans1: QF sentences are boolean combinations of **true** and **false**. Compute their truth values using truth tables.

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

$\mathcal{T}_{UDLO}$ has (i) a QE proc., and (ii) a decision proc. for QF sentences.

Ex1: Design a decision procedure for QF sentences.

Ans1: QF sentences are boolean combinations of **true** and **false**.
Compute their truth values using truth tables.

Ex2: Design a QE procedure.

**Theorem**

*The theory $\mathcal{T}_{UDLO}$ of unbounded dense linear orders is decidable.*

$\mathcal{T}_{UDLO}$ has (i) a QE proc., and (ii) a decision proc. for QF sentences.

Ex1: Design a decision procedure for QF sentences.

Ans1: QF sentences are boolean combinations of **true** and **false**. Compute their truth values using truth tables.

Ex2: Design a QE procedure.

Hint: Given $\exists x\, F$ for a quantifier-free $F$, the proc. works as follows:

1. Transform $\exists x\, F$ to an equivalent $\bigvee_i ((\exists x\, G_i) \wedge H_i)$ where $H_i$ is QF and $G_i$ is conjunction of $x < y$ or $y < x$ for some variable $y \neq x$.

2. Transform $\exists x\, G$ to an equivalent quantifier-free $G'$.

Find out how to do both steps.

1. Transform $\exists x\, F$ to an equivalent $\bigvee_i ((\exists x\, G_i) \wedge H_i)$ where $H_i$ is QF and $G_i$ is conjunction of $x < y$ or $y < x$ for some variable $y$.

2. Transform $\exists x\, G$ to an equivalent quantifier-free $G'$.

How to do the step 2?

1. Transform $\exists x\, F$ to an equivalent $\bigvee_i((\exists x\, G_i) \wedge H_i)$ where $H_i$ is QF and $G_i$ is conjunction of $x < y$ or $y < x$ for some variable $y$.

2. Transform $\exists x\, G$ to an equivalent quantifier-free $G'$.

How to do the step 2?

Assume that $G \equiv (l_1 < x \wedge \ldots \wedge l_m < x \wedge x < u_1 \wedge \ldots \wedge x < u_n)$.

1. Transform $\exists x\, F$ to an equivalent $\bigvee_i ((\exists x\, G_i) \wedge H_i)$ where $H_i$ is QF and $G_i$ is conjunction of $x < y$ or $y < x$ for some variable $y$.

2. Transform $\exists x\, G$ to an equivalent quantifier-free $G'$.

How to do the step 2?

Assume that $G \equiv (l_1 < x \wedge \ldots \wedge l_m < x \wedge x < u_1 \wedge \ldots \wedge x < u_n)$.

If $m = 0$ or $n = 0$, then $\mathcal{T}_{UDLO} \models (\exists x\, F) \leftrightarrow \textbf{true}$. Thus, $G' = \textbf{true}$.

1. Transform $\exists x\, F$ to an equivalent $\bigvee_i ((\exists x\, G_i) \wedge H_i)$ where $H_i$ is QF and $G_i$ is conjunction of $x < y$ or $y < x$ for some variable $y$.

2. Transform $\exists x\, G$ to an equivalent quantifier-free $G'$.

How to do the step 2?

Assume that $G \equiv (l_1 < x \wedge \ldots \wedge l_m < x \wedge x < u_1 \wedge \ldots \wedge x < u_n)$.

If $m = 0$ or $n = 0$, then $\mathcal{T}_{UDLO} \models (\exists x\, F) \leftrightarrow \textbf{true}$. Thus, $G' = \textbf{true}$.

Otherwise,

$$\mathcal{T}_{UDLO} \models (\exists x\, F) \leftrightarrow \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} l_i < u_j.$$

Thus, $G' = \bigwedge_{i=1}^{m} \bigwedge_{j=1}^{n} l_i < u_j$.

**Theorem**

*The theory*

$$\mathcal{T}_{LAR} = \mathrm{Th}(\mathbb{Q}, 1, +, \{c\,\cdot\,\}_{c \in \mathbb{Q}}, <)$$

*of linear arithmetic over the rationals is decidable.*

**Theorem**

*The theory*

$$\mathcal{T}_{LAR} = \mathrm{Th}(\mathbb{Q}, 1, +, \{c\cdot\}_{c\in\mathbb{Q}}, <)$$

*of linear arithmetic over the rationals is decidable.*

Ex: Prove the theorem. Hint: The proof is very similar to the one for the decidability of $\mathcal{T}_{UDLO}$, which we have just studied.

# Presburger arithmetic

**Figure:** Mojzesz Presburger (1904 - 1943).

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ is commonly known as **Presburger arithmetic**.

Natural numbers, not rationals. Only addition. No multiplication.

# Simple number theory in Presburger arithmetic

**Example**

Every natural number is odd or even:

$$\forall x \, \exists y \, (x = y + y \lor x = y + y + 1)\,.$$

## Simple number theory in Presburger arithmetic

**Example**

Every natural number is odd or even:

$$\forall x \, \exists y \, (x = y + y \vee x = y + y + 1) \, .$$

Ex: Consider the following Chicken McNugget problem.

> *Given $a_1, \ldots, a_n \in \mathbb{N}$, is there some $c \in \mathbb{N}$ such that all numbers greater than $c$ can be represented as a non-negative linear combination of $a_1, \ldots, a_n$?*

Express this problem for given $a_1, \ldots, a_n$ in Presburger arithmetic.

# Simple number theory in Presburger arithmetic

**Example**

Every natural number is odd or even:

$$\forall x\, \exists y\, (x = y + y \vee x = y + y + 1)\,.$$

Ex: Consider the following Chicken McNugget problem.

*Given $a_1, \ldots, a_n \in \mathbb{N}$, is there some $c \in \mathbb{N}$ such that all numbers greater than $c$ can be represented as a non-negative linear combination of $a_1, \ldots, a_n$?*

Express this problem for given $a_1, \ldots, a_n$ in Presburger arithmetic.

Ans:

$$\exists x\, \forall y\, (x < y \rightarrow (\exists z_1 \ldots \exists z_n\, (y = a_1 \cdot z_1 + \cdots + a_n \cdot z_n))).$$

Here $a_i \cdot z_i$ is an abbreviation for $\underbrace{z_i + \ldots + z_i}_{a_i \text{ copies}}$.

## Decidability of Presburger arithmetic

**Theorem**

*Presburger arithmetic* $\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ *is decidable.*

# Decidability of Presburger arithmetic

**Theorem**

*Presburger arithmetic* $\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ *is decidable.*

Can't prove it using QE.

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ does not have quantifier elimination. For instance, $y$ cannot be eliminated from $\exists y\,(x = y + y)$.

## Decidability of Presburger arithmetic

**Theorem**

*Presburger arithmetic* $\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ *is decidable.*

Can't prove it using QE.

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ does not have quantifier elimination. For instance, $y$ cannot be eliminated from $\exists y\, (x = y + y)$.

Solution: extend the signature with unary divisibility relations $c \mid \cdot$ for all $c > 0$ such that

$$c \mid n \text{ iff there is } k \in \mathbb{N} \text{ such that } n = k \cdot c.$$

**Theorem**

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <, \{c \mid \cdot\}_{c>0})$ *has a QE procedure.*

## Decidability of Presburger arithmetic

**Theorem**

*Presburger arithmetic* $\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ *is decidable.*

Can't prove it using QE.

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <)$ does not have quantifier elimination. For instance, $y$ cannot be eliminated from $\exists y\,(x = y + y)$.

Solution: extend the signature with unary divisibility relations $c \mid \cdot$ for all $c > 0$ such that

$$c \mid n \text{ iff there is } k \in \mathbb{N} \text{ such that } n = k \cdot c.$$

**Theorem**

$\mathrm{Th}(\mathbb{N}, 0, 1, +, <, \{c \mid \cdot\}_{c>0})$ *has a QE procedure.*

Ex: Use the theorem and prove the decid. of Presburger arithmetic.

The QE procedure for $\mathrm{Th}(\mathbb{N}, 0, 1, +, <, \{c \mid \cdot\}_{c>0})$ works in two steps.

The QE procedure for $\mathrm{Th}(\mathbb{N}, 0, 1, +, <, \{c \mid \cdot\}_{c>0})$ works in two steps.

1. Transform $\exists x\, F$ to an equivalent $\bigvee_l ((\exists x\, F_l) \wedge F'_l)$ where $F'_l$ is QF and $F_l$ has the form

$$F_l = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

2. Transform $\exists x\, F_l$ to an equivalent quantifier-free $G_l$.

The QE procedure for $\mathrm{Th}(\mathbb{N}, 0, 1, +, <, \{c \mid \cdot\}_{c>0})$ works in two steps.

1. Transform $\exists x\, F$ to an equivalent $\bigvee_l((\exists x\, F_l) \wedge F'_l)$ where $F'_l$ is QF and $F_l$ has the form

$$F_l = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

2. Transform $\exists x\, F_l$ to an equivalent quantifier-free $G_l$.

Step 1 is similar to what we did before. Will focus on step 2.

$$F_l = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

Goal: Eliminate quantifiers from $\exists x \, F_l$.

$$F_l = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

Goal: Eliminate quantifiers from $\exists x\, F_l$.

Let $b = \mathrm{lcm}\{a_i \mid i \in L \cup U \cup D\}$.

$$F_I = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

Goal: Eliminate quantifiers from $\exists x \, F_I$.

Let $b = \mathrm{lcm}\{a_i \mid i \in L \cup U \cup D\}$.

Ex1: Show that $\exists x \, F_I$ is equivalent to $\exists x \, H$ where

$$\begin{aligned} H = \bigwedge_{i \in L} \frac{b}{a_i} \cdot q_i(\vec{y}) < x \wedge \bigwedge_{j \in U} x < \frac{b}{a_j} \cdot p_j(\vec{y}) & \\ \wedge \bigwedge_{k \in D} \left( \frac{b}{a_k} \cdot c_k \right) \Bigm| \left( x + \frac{b}{a_k} \cdot r_k(\vec{y}) \right) \wedge b \mid x. & \end{aligned}$$

$$F_I = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

Goal: Eliminate quantifiers from $\exists x\, F_I$.

Let $b = \mathrm{lcm}\{a_i \mid i \in L \cup U \cup D\}$.

Ex1: Show that $\exists x\, F_I$ is equivalent to $\exists x\, H$ where

$$\begin{aligned} H = \bigwedge_{i \in L} \frac{b}{a_i} \cdot q_i(\vec{y}) < x \wedge \bigwedge_{j \in U} x < \frac{b}{a_j} \cdot p_j(\vec{y}) & \\ \wedge \bigwedge_{k \in D} \left( \frac{b}{a_k} \cdot c_k \right) \Bigm| \left( x + \frac{b}{a_k} \cdot r_k(\vec{y}) \right) \wedge b \mid x. & \end{aligned}$$

Define $c = \mathrm{lcm}\big(\{b\} \cup \{b \cdot c_k / a_k : k \in D\}\big)$.

$$F_l = \Big( \bigwedge_{i \in L} q_i(\vec{y}) < a_i \cdot x \wedge \bigwedge_{j \in U} a_j \cdot x < p_j(\vec{y}) \wedge \bigwedge_{k \in D} c_k \mid a_k \cdot x + r_k(\vec{y}) \Big).$$

Goal: Eliminate quantifiers from $\exists x\, F_l$.

Let $b = \mathrm{lcm}\{a_i \mid i \in L \cup U \cup D\}$.

Ex1: Show that $\exists x\, F_l$ is equivalent to $\exists x\, H$ where

$$\begin{aligned} H = & \bigwedge_{i \in L} \frac{b}{a_i} \cdot q_i(\vec{y}) < x \wedge \bigwedge_{j \in U} x < \frac{b}{a_j} \cdot p_j(\vec{y}) \\ & \wedge \bigwedge_{k \in D} \left( \frac{b}{a_k} \cdot c_k \right) \Bigm| \left( x + \frac{b}{a_k} \cdot r_k(\vec{y}) \right) \wedge b \mid x. \end{aligned}$$

Define $c = \mathrm{lcm}\big(\{b\} \cup \{b \cdot c_k / a_k : k \in D\}\big)$.

Ex2: Show that $\exists x\, H$ is equivalent to

$$\begin{cases} \bigvee_{0 \le m < c} H[m/x] & \text{if } L = \emptyset, \\ \bigvee_{i \in L} \bigvee_{1 \le m \le c} H[((b/a_i) \cdot q_i(\vec{y}) + m)/x] & \text{otherwise.} \end{cases}$$

# Time complexity of the decidability algorithm for Presburger arithmetic

**Theorem (Oppen)**

*Presburger arithmetic is decidable in time* $2^{2^{2^{O(n)}}}$.

# Good article on Presburger arithmetic

A suvival guide to Presburger arithmetic.

Written by Christoph Hasse. Published in ACM SIGLOG News 2018.

https://dl.acm.org/citation.cfm?id=3242964.