Satisfiability Checking

07 First-order logic

Prof. Dr. Erika Ábrahám

RWTH Aachen University
Informatik 2
LuFG Theory of Hybrid Systems

WS 22/23

First-order logic

- We have seen that natural languages are not well-suited for correct reasoning.
- Propositional logic is useful but sometimes not expressive enough for modeling.

First-order (FO) logic is a framework with the syntactical ingredients:

1. Theory symbols: constants, variables, function symbols
2. Lifting from theory to the logical level: predicate symbols
3. Logical symbols: Logical connectives and quantifiers

- 3 is fixed

一阶逻辑和命题逻辑的不同之处在于，
一阶逻辑包含量词。

- Fixing 1 and 2 gives different FO instances

一阶逻辑不同于单纯的“命题逻辑”（Proposition Logic），因为，一阶逻辑里面使用了大量quantifier. ∃“和”∀“就是一阶逻辑的”限量词“（Quantifier）

Constants, variables, function symbols, terms

Theory symbols: constants, variables, function symbols

Example:

Constants:	$0, 1$
Variables:	$x, y, z, \ldots$
Function symbol	binary $+$

Terms (theory expressions) are inductively defined by the following rules:

1. All constants and variables are terms.
2. If $t_1, \ldots, t_n$ $(n > 0)$ are terms and f an n-ary function symbol then $f(t_1, \ldots, t_n)$ is a term. **constant和variable是terms, 由他们构成的函数f(c/v)还是terms**

Only strings obtained by finitely many applications of these rules are terms.

Example terms: 0, x, $+(0, 1)$, $+(x, 1)$, $+(x, +(y, 1))$ $(0 + 1)$, $(x + 1)$, $(x + (y + 1))$

Predicates lift terms from the theory to the logical level.

Example predicate symbols: binary $\geq, >, =, <, \leq$ comparable

(Theory) constraints are inductively defined by the following rule:

1. If P is an n-ary predicate symbol and $t_1, \ldots, t_n$ are terms then $P(t_1, \ldots, t_n)$ is a constraint.

Only strings obtained by finitely many applications of this rule are constraints.

Example constraints: $x < (x + 1)$, $((x + 1) + y) = ((x + y) + 1)$

Logical connectives and quantifiers, formulas

first order logic：对于单一变量，不是变量的set

- Logical connectives: unary $\neg$, binary $\wedge, \vee, \rightarrow, \leftrightarrow, \ldots$
- Universal quantifier $\forall$ ("for all"), existential quantifier $\exists$ ("exists")

(Well-formed) formulas are inductively defined by the following rules:

1. If c is a constraint then c is a formula (called atomic formula).
2. If φ is a formula then $(\neg\varphi)$ is a formula.
3. If φ and ψ are formulas then $(\varphi \wedge \psi)$ is a formula.
4. Similar rules apply to other binary logical connectives.
5. If φ is a formula and x is a variable, then $(\forall x.\ \varphi)$ and $(\exists x.\ \varphi)$ are formulas.

Only expressions which can be obtained by finitely many applications of these rules are formulas.

Example formulas:

- $x < (x+1)$ (atomic formula)
- $(\neg\ x < 0)$
- $(x < (x+1)\ \wedge\ ((x+1)+y) = ((x+y)+1))$
- $\forall x.\ \exists y.\ y = (x+1)$

Example

Assume the argumentation:

1. All men are mortal.
2. Socrates is a man.
3. Therefore, Socrates is mortal.

We can formalize it by defining

Constants:	*Socrates*
Variables:	x
Predicate symbols:	unary *isMen*, *isMortal*

Formalization:

1. $\forall x.\ \mathit{isMen}(x) \rightarrow \mathit{isMortal}(x)$
2. $\mathit{isMen}(\mathit{Socrates})$
3. $\mathit{isMortal}(\mathit{Socrates})$

Some remarks and notation

元数, e.g. 一元函数, 二元函数

- Constants can also be seen as function symbols of arity 0.
- Sometimes equality ($=$) is included as a logical symbol.
- Note: the logical connectives negation ($\neg$) and conjunction ($\wedge$) and the existential quantifier ($\exists$) would be sufficient, the remaining syntax ($\vee, \rightarrow, \leftrightarrow, \ldots, \forall$) are syntactic sugar.

We omit parentheses whenever we may restore them through operator precedence (with left-to-right binding for several occurrences of the same operator):

可以通过优先级忽略括号

binds stronger

强 弱

$$\xleftarrow{\hspace{8em}}$$

$$\neg \quad \wedge \quad \vee \quad \rightarrow \quad \leftrightarrow \quad \exists \quad \forall$$

Thus, we write:

$$
\begin{array}{lll}
\neg\neg a & \text{for} & (\neg(\neg a)), \\
\exists a.\ \exists b.\ (a \wedge b \rightarrow P(a,b)) & \text{for} & \exists a.\ \exists b.\ ((a \wedge b) \rightarrow P(a,b))
\end{array}
$$

Free and bound variable occurrences

A variable occurrence in a formula φ is an occurrence of a variable in an atomic sub-formula (constraint) of φ.

Each variable occurence in a formula is either bound or free, defined inductively by the following rules:

- Any occurrence of any variable in any atomic formula is free.
- A variable occurrence in φ is free in $(\neg\varphi)$ iff it is free in φ.
- A variable occurrence in φ is free in $(\varphi \wedge \psi)$ iff it is free in φ, and analogously for the symmetric case and all other binary Boolean connectives.
- An occurrence of a variable x in φ is free in $(\exists y.\ \varphi)$ iff x is free in φ and x is a symbol different from y.
- An analogous rule holds with $\forall$ in place of $\exists$.
- A variable occurrence is bound iff it is not free.

Free and bound variable occurrences

Examples:

- In

  $$P(z) \vee \forall x.\ (P(x) \rightarrow Q(z))$$

  z is occurs free in $P(z)$ and in $Q(z)$, whereas x occurs bound in $P(x)$.

- In

  $$Q(z) \vee \forall z.P(z)$$

  the first occurrence of z is free whereas its second occurrence is bound.

Signature Σ, Σ-formula, Σ-sentence

The non-logical symbols 包含predicates and individual constants. These include symbols that, in an interpretation, may stand for individual $\boxed{\text{constants, variables, functions, or predicates.}}$

- A signature Σ fixes the set of non-logical symbols (up to variables).
- A Σ-formula is a formula with non-logical symbols from Σ.
- A Σ-sentence is a Σ-formula without free variable occurrences.

In the previous example: $\Sigma = (\mathit{Socrates}, \mathit{isMen}(\cdot), \mathit{isMortal}(\cdot))$ with

- $\mathit{Socrates}$ a constant and
- isMen and $\mathit{isMortal}$ unary predicate symbols.

类似的, logical symbols包含truth-functional connectives (such as "and", "or", "not", "implies", and logical equivalence) and the symbols for the quantifiers "for all" and "there exists".

The formulas

1. $\forall x.\ \mathit{isMen}(x) \rightarrow \mathit{isMortal}(x)$
2. $\mathit{isMen}(\mathit{Socrates})$
3. $\mathit{isMortal}(\mathit{Socrates})$

are Σ-sentences (all occurrences of the only variable x are bound).

Exercise A

Assume the following signature Σ:

- 0 and 2 are constants;
- x, y, z are variables;
- $*$ is a binary function;
- $>$ and $=$ are binary predicates.

How many occurrences of x are free in the following Σ-formula?

$$x = \boxed{0} \vee \boxed{\forall y.\ (x > (2 * x))} \vee \boxed{\forall x.\ \neg((x = y) \rightarrow (\forall x.\ x > z)))}$$

- 0
- 1
- 2
- 3
- 4
- 5

Further examples

- $\Sigma = \{0, 1, +, >\}$
  - $0, 1$ are constant symbols
  - $+$ is a binary function symbol
  - $>$ is a binary predicate symbol

- Examples of Σ-sentences:

$$\exists x.\ \forall y.\ x > y$$
$$\forall x.\ \exists y.\ x > y$$
$$\forall x.\ x + 1 > x$$
$$\forall x.\ \neg(x + 0 > x \lor x > x + 0)$$

Further examples

- $\Sigma = \{0, 1, +, *, <, \mathit{isPrime}\}$
  - $0, 1$ constant symbols
  - $+, *$ binary function symbols
  - $<$ binary predicate symbol
  - $\mathit{isPrime}$ unary predicate symbol

- An example Σ-sentence:

$$\forall n.\ (1 < n \rightarrow (\exists p.\ \mathit{isPrime}(p) \land n < p \land p < 2 * n))$$

Further examples

- Let $\Sigma = \{0, 1, +, =\}$ where $0, 1$ are constants, $+$ is a binary function symbol and $=$ a binary predicate symbol.
- Let $\varphi = \exists x.\ x + 0 = 1$ a Σ-formula.
- Q: Is φ true?
- A: So far these are only symbols, strings. No meaning yet.
- Q: What do we need to fix for the semantics?
- A: We need a domain for the variables. Let's say $\mathbb{N}_0$.
- Q: Is φ true in $\mathbb{N}_0$?
- A: Depends on the interpretation of '$+$' and '$=$'!

Bonus exercise 9

Assume a signature Σ consisting of theory constants a and b, unary function f, binary function g, unary predicate p, binary predicate q (and allowing variables names x and y).

Which of the following are (well-formed) Σ-formulas?

(Multiple choice: please select all correct cases.)

unary 1 argument

- $x(a)$
- $(\forall x. (\exists x. p(a)))$
- $q(f)$

binary ，需要 2个 arg

unary ，需要 arg

- $(\neg(\forall x. p(x)))$
- $f(p(y(q)), p(y(q)))$
- $(\forall x. (\exists x. p(a)))$
- $q(y, f(y))$

Structures

- A Σ-structure is given by:
  - a domain D,
  - an interpretation I of the non-logical symbols in Σ that maps
    - each constant symbol to a domain element,
    - each function symbol of arity n to a function of type $D^n \to D$, and
    - each predicate symbol of arity n to a predicate of type $D^n \to \{0, 1\}$.
- To give meaning to formulas with free variable occurrences, we also need an assignment α that maps each variable (with a free occurrence) to a domain element.

Semantics

To give semantics to a logical system means to define a notion of truth for the formulas.

Semantics of terms and formulas under a structure $S = (D, I)$ and an assignment α:

Interpretation

constants:	$[\![c]\!]_{S,\alpha}$	$= I(c)$
variables:	$[\![x]\!]_{S,\alpha}$	$= \alpha(x)$
functions:	$[\![f(t_1, \ldots, t_n)]\!]_{S,\alpha}$	$= I(f)([\![t_1]\!]_{S,\alpha}, \ldots, [\![t_n]\!]_{S,\alpha})$
predicates:	$S, \alpha \models p(t_1, \ldots, t_n)$	iff $I(p)([\![t_1]\!]_{S,\alpha}, \ldots, [\![t_n]\!]_{S,\alpha})$

logical structure:

$S, \alpha \models \neg\varphi$	iff $S, \alpha \not\models \varphi$
$S, \alpha \models \varphi \wedge \psi$	iff $S, \alpha \models \varphi$ and $S, \alpha \models \psi$
$S, \alpha \models \exists x.\ \varphi$	iff there exists $v \in D$ such that $S, \alpha[x \mapsto v] \models \varphi$

Satisfiability, validity

- A Σ-formula φ is satisfiable if there exist a Σ-structure S and an assignment α that satisfy it.

  Notation: $S, \alpha \models \varphi$. For Σ-sentences we also write $S \models \varphi$.

- A Σ-formula φ is valid if it is satisfied by all Σ-structures and all assignments.

  Notation: $\models \varphi$.

Examples

- $\Sigma = \{0, 1, +, =\}$
- $\varphi = \exists x.\ x + 0 = 1$ a Σ-formula
- Q: Is φ satisfiable?
- A: Yes. Consider the structure S:
  - Domain: $\mathbb{N}_0$
  - Interpretation:
    - 0 and 1 are mapped to 0 and 1 in $\mathbb{N}_0$
    - $+$ means addition
    - $=$ means equality

S satisfies φ. S is said to be a model of φ.

Examples

- $\Sigma = \{0, 1, +, =\}$
- $\varphi = \exists x.\ x + 0 = 1$ a Σ-formula
- Q: Is φ valid?
- A: No. Consider the structure S':
  - Domain: $\mathbb{N}_0$
  - Interpretation:
    - 0 and 1 are mapped to 0 and 1 in $\mathbb{N}_0$
    - $+$ means multiplication
    - $=$ means equality

  S' does not satisfy φ.

Theories T, T-safisfiability and T-validity

- A Σ-theory T is defined by a set of Σ-sentences.

- A Σ-formula φ is T-satisfiable if there exists a structure that satisfies both the sentences of T and φ.
- A Σ-formula φ is T-valid if all structures that satisfy the sentences defining T also satisfy φ.

- The number of sentences that are necessary for defining a theory may be large or infinite.
- Instead, it is common to define a theory through a set of axioms.
- The theory is defined by these axioms and everything that can be inferred from them by a sound inference system.

Examples

- $\Sigma = \{0, 1, +, =\}$
- $\varphi = \exists x.\ x + 0 = 1$ a Σ-formula.
- We now define the Σ-theory T by the following axioms:
  1. $\forall x.\ x = x$ $\quad$ //$=$ must be reflexive
  2. $\forall x.\ \forall y.\ x + y = y + x$ $\quad$ //$+$ must be commutative
- Q: Is φ T-satisfiable?
- A: Yes, S is a model.
- Q: Is φ T-valid?
- A: No. S' satisfies the sentences in T but not φ.

Examples

- $\Sigma = \{0, 1, +, =\}$
- $\varphi = \exists x.\ x + 0 = 1$ a Σ-formula.
- We now define the Σ-theory T by the following axioms:
  1. $\forall x.\ x = x$ ($=$ is reflexive)
  2. $\forall x, y, z.\ ((x = y \wedge y = z) \rightarrow x = z)$ ($=$ is transitive)
  3. $\forall x.\ \forall y.\ x + y = y + x$ ($+$ is commutative)
  4. $\forall x.\ 0 + x = x$ (0 is neutral element for $+$)
- Q: Is φ T-satisfiable?
- A: Yes, S is a model.
- Q: Is φ T-valid?
- A: Yes. (S' does not satisfy the fourth axiom. 3, 4 $\rightarrow \varphi$.)

Examples

- $\Sigma = \{=\}$
- $\varphi = (x = y \wedge y \neq z) \rightarrow x \neq z$ a Σ-formula
- We now define the Σ-theory T by the following axioms:
  1. $\forall x.\ x = x$ (reflexivity)
  2. $\forall x.\ \forall y.\ x = y \rightarrow y = x$ (symmetry)
  3. $\forall x.\ \forall y.\ \forall z.\ x = y \wedge y = z \rightarrow x = z$ (transitivity)
- Q: Is φ T-satisfiable?
- A: Yes.
- Q: Is φ T-valid?
- A: Yes. Every structure that satisfies T also satisfies φ.

Examples

- $\Sigma = \{<\}$
- $\varphi : \forall x.\ \exists y.\ y < x$ a Σ-formula
- Consider the Σ-theory T defined by the axioms:
  1. $\forall x.\ \forall y.\ \forall z.\ (x < y \land y < z) \rightarrow x < z$ (transitivity)
  2. $\forall x.\ \forall y.\ x < y \rightarrow \neg(y < x)$ (anti-symmetry)
- Q: Is φ T-satisfiable?
- A: Yes. We construct a model for it:
  - Domain: $\mathbb{Z}$
  - $<$ means “less than”
- Q: Is φ T-valid?
- A: No. We construct a structure to the contrary:
  - Domain: $\mathbb{N}_0$ 空集
  - $<$ means “less than”

Bonus exercise 10

Consider:

- Signature Σ with variables x and y and a binary predicate $\sim$.
- Σ-formula φ: $\exists x.\ \exists y.\ x \sim y \land y \sim x$

Which of the following structures satisfy φ?
(Multiple choice: please select all models.)

- Domain is $\mathbb{Z}$, $\sim$ means equal
- Domain is $\mathbb{Z}$, $\sim$ means not equal
- Domain is $\mathbb{Z}$, $\sim$ means less than
- Domain is $\mathbb{R}$, $\sim$ means less than
  实数
- Domain is $\mathbb{Q}$, $\sim$ means less than or equal

N：非负整数　　Z：整数　　Q：有理数

R：实数

Let Σ be a signature with variables a and b and a binary predicate $\sim$.
Assume the Σ-formula φ: $\forall a.\ \exists b.\ (a \sim b \to b \sim a)$.
Which of the following structures satisfy φ? (Multiple choice: please select all models.)

Select one or more:

- [x] Domain is $\mathbb{Q}$, $\sim$ means greater than
- [x] Domain is $\mathbb{N}$, $\sim$ means greater than or equal
- [x] Domain is $\mathbb{Z}$, $\sim$ means less than or equal ✔
- [ ] None of the above.

如果题目变成：$\exists a, \forall b. (a \sim b \to b \sim a)$
则错误

$\exists \cdots \forall \cdots$ 和 $\forall \cdots \exists \cdots$
不一样!!!

The correct answers are: Domain is $\mathbb{Q}$, $\sim$ means greater than, Domain is $\mathbb{N}$, $\sim$ means greater than or equal, Domain is $\mathbb{Z}$, $\sim$ means less than or equal

Let Σ be a signature with variables a and b and a binary predicate $\sim$.
Assume the Σ-formula φ: $\exists a.\ \forall b.\ (a \sim b \vee b \sim a)$.
Which of the following structures satisfy φ? (Multiple choice: please select all models.)

Wählen Sie eine oder mehrere Antworten:

- [ ] Domain is $\mathbb{Z}$, $\sim$ means less than or equal
- [ ] Domain is $\mathbb{Q}$, $\sim$ means greater than or equal
- [ ] Domain is $\mathbb{Q}$, $\sim$ means greater than or equal
- [x] None of the above. ✖

Die richtigen Antworten sind: Domain is $\mathbb{Z}$, $\sim$ means less than or equal, Domain is $\mathbb{Q}$, $\sim$ means greater than or equal, Domain is $\mathbb{Q}$, $\sim$ means greater than or equal

Let Σ be a signature with variables u and v and a binary predicate $\sim$.
Assume the Σ-formula φ:
$\exists u.\ \forall v.\ (u \sim v \to v \sim u)$.
Which of the following structures satisfy φ?
(Multiple choice: please select all models.)

Select one or more:

- [ ] Domain is $\mathbb{N}$, $\sim$ means greater than or equal
- [ ] Domain is $\mathbb{N}$, $\sim$ means less than or equal
- [x] Domain is $\mathbb{Z}$, $\sim$ means equal ✔
- [ ] None of the above.

The correct answers are: Domain is $\mathbb{N}$, $\sim$ means greater than or equal, Domain is $\mathbb{Z}$, $\sim$ means equal

Let Σ be a signature with variables a and b and a binary predicate $\sim$.
Assume the Σ-formula φ:
$\forall a.\ \forall b.\ (a \sim b \to b \sim a)$.
Which of the following structures satisfy φ?
(Multiple choice: please select all models.)

Wählen Sie eine oder mehrere Antworten:

- [x] Domain is $\mathbb{Q}$, $\sim$ means equal ✔
- [ ] Domain is $\mathbb{N}$, $\sim$ means less than
- [x] Domain is $\mathbb{N}$, $\sim$ means equal ✔
- [ ] None of the above.

Die richtigen Antworten sind: Domain is $\mathbb{Q}$, $\sim$ means equal, Domain is $\mathbb{N}$, $\sim$ means equal

(VU) Satisfiability Checking / Bonus test 10

Bonus test 10

Started on	Friday, 3 November 2023, 9:48 AM
State	Finished
Completed on	Friday, 3 November 2023, 9:54 AM
Time taken	5 mins 2 secs
Grade	**0.33** out of 0.33 (**100**%)

Question 1

Correct

Mark 0.33 out of 0.33

Flag question

Let Σ be a signature with variables u and v and a binary predicate $\sim$.
Assume the Σ-formula φ: $\forall u.\ \forall v.\ (u \sim v \rightarrow v \sim u)$.
Which of the following structures satisfy φ? (Multiple choice: please select all models.)

Select one or more:

- [ ] Domain is $\mathbb{R}$, $\sim$ means less than
- [ ] Domain is $\mathbb{Q}$, $\sim$ means less than or equal
- [ ] Domain is $\mathbb{Q}$, $\sim$ means less than or equal
- [x] None of the above. ✔

The correct answer is: None of the above.

Some famous theories

We assume in the following that the interpretation of symbols is fixed to their common use.

- Thus $+$ is plus, ...

Some famous theories:

- Presburger arithmetic: $\Sigma = \{0, 1, +, >\}$ over integers
- Peano arithmetic: $\Sigma = \{0, 1, +, *, >\}$ over integers
- Linear real arithmetic: $\Sigma = \{0, 1, +, >\}$ over reals
- Real arithmetic: $\Sigma = \{0, 1, +, *, >\}$ over reals
- Theory of arrays
- Theory of pointers
- ...

Logic fragments

- So far we only restricted the non-logical symbols by signatures and their interpretation by theories.
- Sometimes we want to restrict the grammar and the logical symbols that we can use as well.
- These are called logic fragments.
- Examples:
  - The quantifier-free fragment over $\Sigma = \{0, 1, +, =\}$
  - The conjunctive fragment over $\Sigma = \{0, 1, +, =\}$

Logic fragments

- Q: Which FO theory is propositional logic?
- A: The quantifier-free fragment of the FO theory with signature $\Sigma = \{x_1, x_2, \ldots, \mathit{identity}\}$ with variables $x_1, x_2, \ldots$, the unary *identity* predicate (which we skip in the syntax), and without axioms.

  Example: $x_1 \rightarrow (x_2 \vee x_3)$

  Thus, propositional logic is also a first-order theory.
  (A very degenerate one.)

- Q: What if we allow quantifiers?
- A: We get the theory of quantified boolean formulas (QBF).
  Example:
  - $\forall x_1.\ \exists x_2.\ \forall x_3.\ x_1 \rightarrow (x_2 \vee x_3)$

Logic fragments

It is common to present logic fragments via an abstract grammar rather than restrictions on the generic first-order grammar.

Example: Equality logic

- Grammar:

$$
\begin{array}{ll}
\mathit{formula} ::= & \mathit{atom} \mid \mathit{formula} \land \mathit{formula} \mid \neg \mathit{formula} \\[1em]
\mathit{atom} ::= & \mathit{Boolean\text{-}variable} \mid \\
& \mathit{variable} = \mathit{variable} \mid \\
& \mathit{variable} = \mathit{constant} \mid \\
& \mathit{constant} = \mathit{constant}
\end{array}
$$

- Interpretation: $=$ is equality.

Logic fragments

Exampe: 2-CNF

- Grammar:

$$\begin{array}{lcl} \mathit{formula} & ::= & (\mathit{literal} \vee \mathit{literal}) \mid \mathit{formula} \wedge \mathit{formula} \\ \mathit{literal} & ::= & \mathit{Boolean\text{-}variable} \mid \neg \mathit{Boolean\text{-}variable} \end{array}$$

- Example formula:

$$(x_1 \vee \neg x_2) \wedge (\neg x_3 \vee x_2)$$

Expressivity

- Consider the propositional logic formula $\varphi = (x_1 \vee x_2 \vee x_3)$.
- Q: Can we express this in 2-CNF, i.e., can we define a 2-CNF formula that is satisfied by the same assignments?
- A: No.
- Proof:
  - The language accepted by φ has 7 words: all assignments other than $x_1 = x_2 = x_3 = 0$ (*false*).
  - A 2-CNF clause removes 2 assignments, which leaves us with 6 accepted words.
    E.g., $(x_1 \vee x_2)$ removes the assignments $x_1 = x_2 = x_3 = 0$ and $x_1 = x_2 = 0$, $x_3 = 1$.
  - Additional clauses only remove more assignments.
- We say that propositional logic is more expressive than 2-CNF.
- Notation: $\mathcal{L}_1 \prec \mathcal{L}_2$ means that $\mathcal{L}_2$ is more expressive than $\mathcal{L}_1$.
- Generally there is only a partial order between theories.

The tradeoff

- So we see that theories can have different expressive power.
- The more expressive the logic the harder it might be to decide the satisfiability/validity of formulas; thus sometimes we aim at less expressiveness that is decidable at lower costs.
- Perhaps it is a bit counterintuitive, but adding restrictions to a theory in form of further axioms may make the theory harder to decide or even turn the satisfiability problem to be undecidable.

Example: First-order Peano arithmetic

- $\Sigma = \{0, 1, +, *, =\}$
- Domain: Natural numbers
- Axioms ("semantics"):
  1. $\forall x.\ (x \neq x + 1)$
  2. $\forall x.\ \forall y.\ (x \neq y) \rightarrow (x + 1 \neq y + 1)$
  3. Induction
  4. $\forall x.\ x + 0 = x$
  5. $\forall x.\ \forall y : (x + y) + 1 = x + (y + 1)$
  6. $\forall x.\ x * 0 = 0$
  7. $\forall x.\ \forall y.\ x * (y + 1) = x * y + x$

UNDECIDABLE!

Reduction: Peano arithmetic to Presburger arithmetic

- $\Sigma = \{0, 1, +, \cancel{*}, =\}$
- Domain: Natural numbers
- Axioms ("semantics"):
  1. $\forall x.\ (x \neq x + 1)$
  2. $\forall x.\ \forall y.\ (x \neq y) \rightarrow (x + 1 \neq y + 1)$
  3. Induction
  4. $\forall x.\ x + 0 = x$
  5. $\forall x.\ \forall y.\ (x + y) + 1 = x + (y + 1)$
  6. $\cancel{\forall x.\ x * 0 = 0}$
  7. $\cancel{\forall x.\ \forall y.\ x * (y + 1) = x * y + x}$

DECIDABLE!

- Q1: Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two theories whose satisfiability problem is decidable and in the same complexity class. Is the satisfiability problem of an $\mathcal{L}_1$ formula reducible to a satisfiability problem of an $\mathcal{L}_2$ formula?

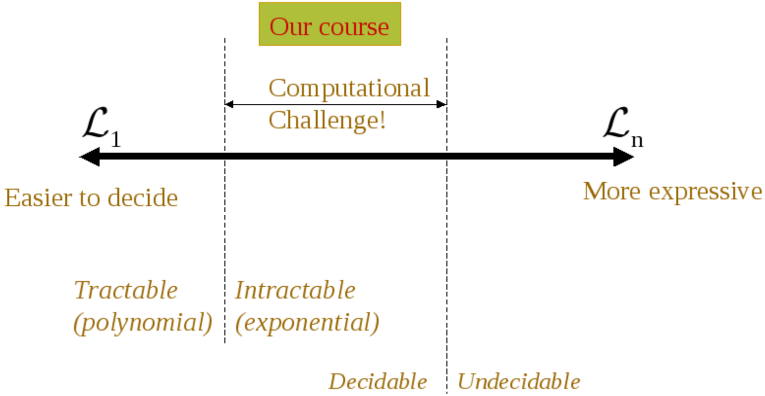
  A1: Yes, reduction with the given complexity is possible.

- Q2: Let $\mathcal{L}_1$ and $\mathcal{L}_2$ be two theories whose satisfiability problems are reducible to each other. Are $\mathcal{L}_1$ and $\mathcal{L}_2$ in the same complexity class?

  A2: It depends on the complexity of the reduction.

Tradeoff: Expressivity vs. computational hardness

- Expressible enough to state something interesting.
- Decidable (or semi-decidable) and more efficiently solvable than richer theories.



In this lecture we assume $P \neq NP$.

Exercise B

Which of the following statements are true?
Multiple choice: Please select all true statements.

- Propositional logic is a fragment of 2-CNF.
- 2-CNF is less expressive than propositional logic.
- There exists a signature Σ and a Σ-theory T such that no Σ-formulas are T-satisfiable.
- There exist undecidable FO theories.

Learning target

- What is first-order (FO) logic?
- How is the semantics of FO logic formulas defined by structures?
- When is a Σ-formula satisfiable resp. valid?
- What is a Σ-theory T?

  When are Σ-formulas T-satisfiable resp. T-valid?
- What is a logic fragment?
- What does it mean that one theory or logic fragment is more expressive than another one?