First order Theories

Klaus v. Gleissenthall

VRIJE UNIVERSITEIT AMSTERDAM

Where are we?

- Logic as the language of computation
- We’ve seen first-order logic
- *Very* expressive
- But undecidable makes decision procedures unpredictable
  - We don’t know if they will terminate!
- Next first-order theories
- Focus on decidable fragments of FOL that allow encoding interesting questions about programs

First Order Theories: Motivation

- In FOL functions & predicates are uninterpreted

  (the structure can assign *any* meaning)
- But often, we have a particular meaning in mind! (say, >, =, +)
- First-order theories allow us to give meaning to the symbols used in a first-order language

First Order Theories

- A **first-order theory T** consists of:
  1. Signature Σ_T : set of constants, functions, predicate symbols
  2. Axioms A_T: set of FOL sentences over Σ_T
- **Σ_T formula**: Formula constructed from symbols in Σ_T and variables, logical connectives, and quantifiers

Example: The theory of heights $\mathbf{T_H}$ has signature Σ_H : R={taller}, C=F= $\emptyset$ and axiom:

$$\forall x.\forall y.\ (taller\ (x\ ,\ y) \rightarrow \neg taller\ (y,\ x))$$

Quiz:

1. Is $\exists x.\forall z.taller(x,\ z) \wedge taller\ (y,\ w)$ legal Σ_H formula?
2. Is $\exists x\ .\forall z\ .taller\ (x\ ,\ z\ ) \wedge taller\ (joe,\ tom)$?

First Order Theories: Axioms

- The axioms A_T assign meaning to the symbols in Σ_T .
- Specifically, axioms ensure that some legal interpretations in FOL are ruled out in **T**

Example: Consider relation constant *taller*, and universe U = {A, B, C}

- In FOL, a possible implementation is I(*taller*) $\triangleq \{\langle A,B\rangle, \langle B,A\rangle\}$
- In $\mathbf{T_H}$, this interpretation is **not possible**, as it violates our axiom

Models of **T**

- A structure $M \triangleq \langle U,I \rangle$ is a model of theory **T**, or **T**-model, iff $M \models A$ for every $A \in A_T$, i.e., it satisfies all the axioms.

Example: Consider structure consisting of universe $U = \{A, B\}$ and interpretation

$$I(\text{taller}) \triangleq \{\langle A, A\rangle, \langle B, B\rangle\}$$

Quiz: Is this a model of $\mathbf{T}_H$?

Say, we change the interpretation to $I(\text{taller}) \triangleq \{\langle A, B\rangle\}$. Is this a $\mathbf{T}_H$ model?

Say, we add axiom: $\forall x,y,z\ .\ (\text{taller}(x, y) \wedge \text{taller}(y, z) \rightarrow \text{taller}(x, z))$

Consider $I(\text{taller}) \triangleq \{\langle A, B\rangle, \langle B, C\rangle\}$. Is U,I a theory model?

Satisfiability Modulo T (SMT)

- Formula F is **satisfiable modulo T** if there exists a **T**-model M and variable assignment σ, such that M, σ ⊨ F

- Formula F is **valid modulo T** if for all **T**-models M and all variable assignments σ, it holds that M, σ ⊨ F

Quiz:
- Question: How is validity modulo T different from FOL-validity?

- If a formula F is valid modulo theory T, we will write **T** ⊨ **F**

- Theory **T** consists of all sentences that are valid in **T**

Satisfiability Modulo **T**

Quiz:

- Consider some first order theory T:
  - If a formula is valid in FOL, is it also valid modulo T?
  - If a formula is valid modulo T, is it also valid in FOL?

Satisfiability Modulo **T**

- **Plan**: we'll look at theories we need for reasoning about programs
  - Equality
  - Arithmetic
  - Data-structures: Arrays

- Remember: we want to find theories that are <u>decidable</u> as we want verification to be predictable (i.e., not loop forever on some inputs)

Theory of Equality $T_=$

Signature:

- Extend first-order logic with a “built-in” equality predicate =
- Signature $\Sigma_=$: $R \triangleq \{=, p, q, r, \ldots\}$, $C \triangleq \{a, b, c, \ldots\}$ $F \triangleq \{f, g, \ldots\}$
- Only = is “interpreted"

Axioms: Define the meaning (interpretation) of =

1. $\forall x . x = x$ (reflexivity)
2. $\forall x. \forall y. (x = y \rightarrow y = x)$ (symmetry)
3. $\forall x. \forall y. \forall z. (x = y \land y = z \rightarrow x = z)$ (transitivity)

Theory of Equality $T_=$

Example: Consider universe U = {□,◍}

Quiz: Which interpretations of = are allowed by the axioms?

- I(=) ≜ {⟨□,◍⟩,⟨◍,□⟩} ?
- I(=) ≜ {⟨□,□⟩,⟨◍,◍⟩} ?
- I(=) ≜ {⟨□,□⟩,⟨□,◍⟩,⟨◍,□⟩,⟨◍,◍⟩} ?

Congruence

Function Congruence:

For function $f(x_1, \ldots, x_k)$, we add an axiom:

$$\forall x_1,\ldots, x_n, y_1,\ldots, y_n.\ \ x_1 = y_1 \wedge \ldots \wedge x_k = y_k \rightarrow f(x_1,\ldots , x_n) = f(y_1,\ldots , y_n)$$

Predicate Congruence:

For function $p(x_1, \ldots, x_k)$, we add an axiom:

$$\forall x_1,\ldots, x_n, y_1,\ldots, y_n.\ \ x_1 = y_1 \wedge \ldots \wedge x_k = y_k \rightarrow p(x_1,\ldots , x_n) \leftrightarrow p(y_1,\ldots , y_n)$$

- Function and predicate congruence "axioms" are really sets of axioms, one for each function or predicate

Congruence

Example: Consider universe U = {□,◍,★} and

- I(=) ≜ {⟨□,□⟩,⟨□,◍⟩,⟨◍,□⟩,⟨◍,◍⟩,⟨★,★⟩} ?

Quiz: Which interpretations for a function *f* is valid?

- I(f) ≜ {◍ → □, □ →★, ★→★ } ?
- I(f) ≜ {◍ → ◍, □ → ◍, ★→ ◍ } ?
- I(f) ≜ {◍ → □, □ →◍, ★→★ } ?

Decidability of $T_=$

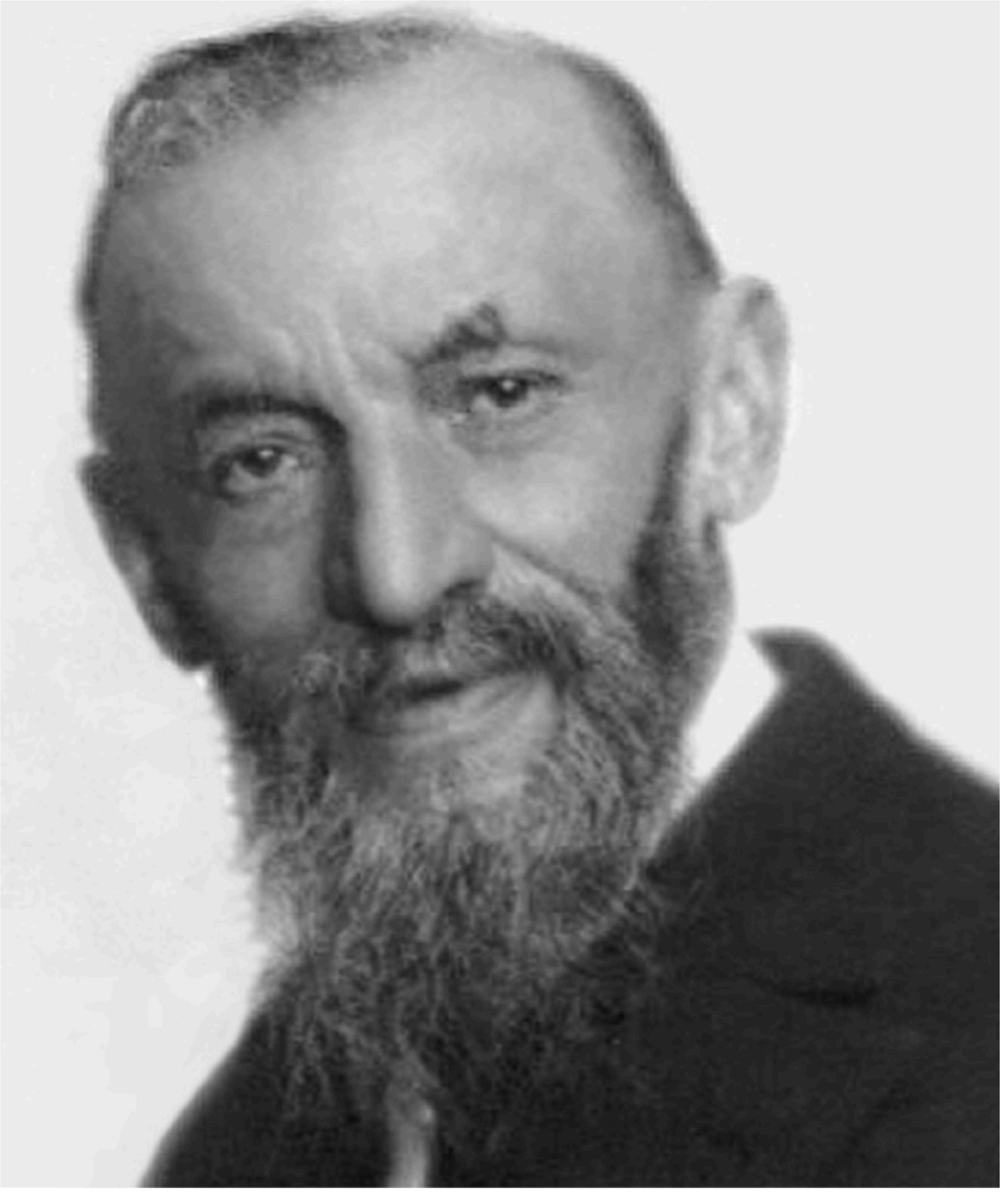
- Is the full theory of equality <u>decidable</u>?
- The quantifier-free fragment of $T_=$ is decidable but NP-complete
- Conjunctive quantifier-free formulas can be efficiently solved via congruence closure
- To solve disjunctive formulas, pair up with SAT solver

Peano Arithmetic

- Allows multiplication and addition over natural numbers
- The theory of Peano arithmetic T_{PA} has signature

$$\Sigma_{PA} \triangleq \{0,\ 1,\ +,\ \cdot,\ =\}$$

- 0,1 are constants
- $+, \cdot$ are binary functions
- $=$ is a binary predicate



Giuseppe Peano

Peano Arithmetic

Quiz:

- Is the following formula a well-formed T_{PA} formula?

$$x + y = 1 \lor f(x) = 1 + 1$$

- What about $\forall x. \exists y. \exists z \,.\, x + y = 1 \lor z \cdot x = 1 + 1$

- What about $2x = y$?

Peano Arithmetic: Axioms

- Includes equality axioms, reflexivity, symmetry, and transitivity
- In addition, axioms to give meaning to remaining symbols:

1. $\forall x . \neg(x + 1 = 0)$: 0 is the minimal element of $\mathbb{N}$ *(zero)*
2. $\forall x . x + 0 = x$: 0 is identity for + *(plus zero)*
3. $\forall x.\forall y.\ x + 1 = y + 1 \rightarrow x = y$ *(successor)*
4. $\forall x.\forall y.\ x + (y + 1) = (x + y) + 1$ *(plus successor)*
5. $\forall x . x \cdot 0 = 0$ *(times zero)*
6. $\forall x.\forall y.\ x \cdot (y + 1) = x \cdot y + x$ *(times successor)*

Peano Arithmetic: Axioms

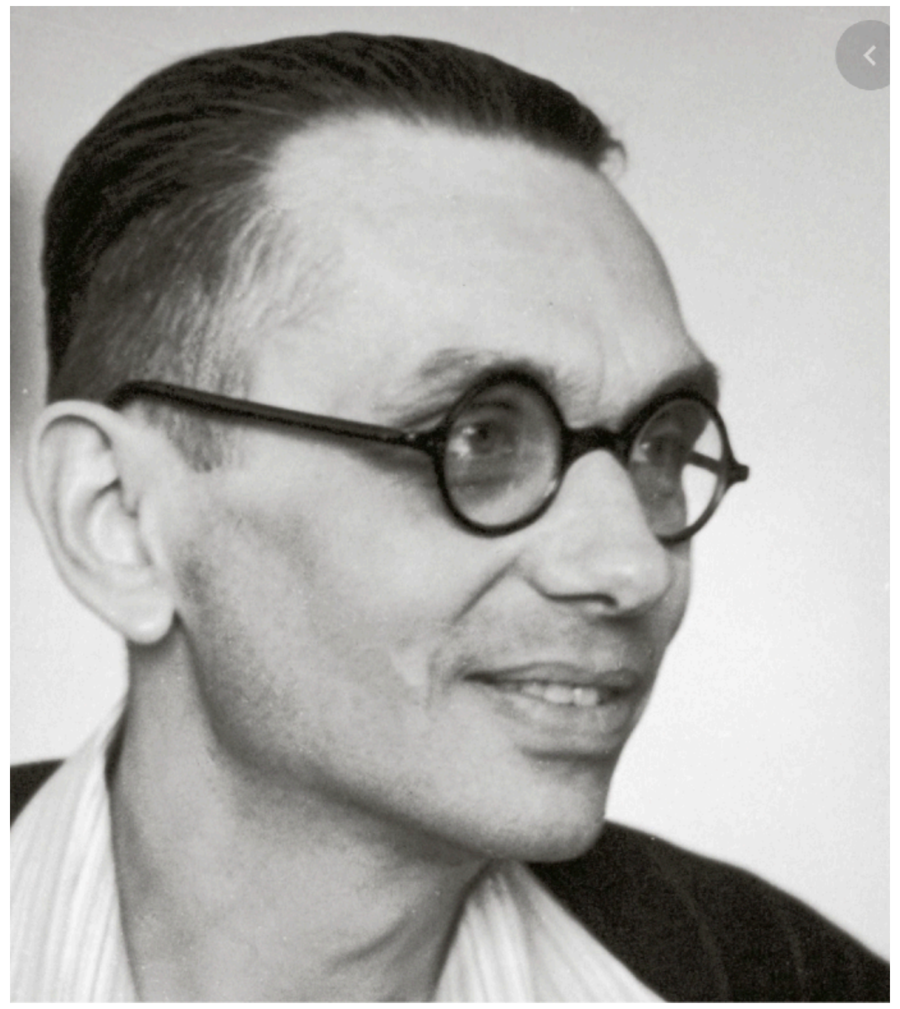
- Final Axiom: Axiom schema for induction

$$(F[0] \wedge (\forall x \,.\, F[x] \rightarrow F[x+1])) \rightarrow \forall x . F[x]$$

- Substitute for F each T_{PA} formula with exactly one free variable

- Any valid interpretation must obey the principle of induction

Decidability of T_{PA}



Kurt Gödel

- Validity in full T_{PA} is undecidable. (Gödel)
- Validity in even the quantifier-free fragment of T_{PA} is undecidable. (Matiyasevitch, 1970)
- The kicker: T_{PA} is also incomplete. (Gödel)
- Why are we even discussing T_{PA}? Shouldn't there be a better set of axioms?
- No: Any suitable axiomatization of arithmetic is either inconsistent or incomplete (Gödel)
- Clearly too expressive! We need less expressive encodings of arithmetic
- Idea: drop multiplication!

Presburger Arithmetic: $T_{\mathbb{N}}$

- The theory of Presburger arithmetic $T_{\mathbb{N}}$ has signature: $\Sigma_{\mathbb{N}} \triangleq \{0, 1, +, =\}$
- Axioms:

1. $\forall x . \neg(x + 1 = 0)$ *(zero)*
2. $\forall x . x + 0 = x$ *(plus zero)*
3. $\forall x . \forall y .\ x + 1 = y + 1 \rightarrow x = y$ *(successor)*
4. $\forall x . \forall y .\ x + (y + 1) = (x + y) + 1$ *(plus successor)*
5. $F[0] \wedge (\forall x . F[x] \rightarrow F[x+1]) \rightarrow \forall x . F[x]$ *(induction)*

Decidability of $T_{\mathbb{N}}$

- Validity in quantifier-free fragment of Presburger arithmetic is decidable (coNP-complete).
- Validity in full Presburger arithmetic is also decidable (Presburger, 1929)
- But: super exponential complexity: $O(2^{2n})$
- Presburger arithmetic is also complete: For any sentence F, $T_{\mathbb{N}} \models F$ or $T_{\mathbb{N}} \models \neg F$
- Admits quantifier elimination: For any formula F in $T_{\mathbb{N}}$, there exists an equivalent quantifier-free formula F'
- Nice properties, still too slow to be used in practice!

Integer Arithmetic $T_{\mathbb{Z}}$

- Signature: $\Sigma_{\mathbb{Z}}$: {. . . , −2, −1, 0, 1, 2, . . . , -3*,-2*,2*,3*, +, −, =, >}
- Also referred to as the theory of linear arithmetic over integers (LIA)
- Equivalent in expressiveness to Presburger arithmetic

  (i.e., every $T_{\mathbb{Z}}$ can be encoded as a formula in Presburger arithmetic)

Theory of Rationals $T_{\mathbb{Q}}$

- So far, arithmetic over integers
- Next: the theory of rationals $T_{\mathbb{Q}}$, which is much more efficiently decidable
- Signature:

$$\Sigma_{\mathbb{Q}} \triangleq \{0, 1, +, -, =, \geq\}$$

- Too many axioms to cover!
- (Almost) the same signature as $T_{\mathbb{Z}}$. What's the difference?

Theory of Rationals $T_{\mathbb{Q}}$

Example: $\exists x \,.\, (1 + 1) * x = 1 + 1 + 1$

Quiz:

- Is this formula valid in $T_{\mathbb{Q}}$?
- What about $T_{\mathbb{Z}}$?

Decidability of $T_{\mathbb{Q}}$

- Full theory of rationals is decidable, but doubly exponential
- Conjunctive quantifier-free fragment efficiently decidable (polynomial time 😍)
- $T_{\mathbb{Q}}$ is the basis for arithmetic reasoning in SMT solvers (like Z3)!
- In practice: use the simplex algorithm (Dantzig)
- Really nice algorithm & deep theory, but we likely won't have time to cover
- Also serves as basis for $\Sigma_{\mathbb{Z}}$ using some clever tricks:

  https://theory.stanford.edu/~aiken/publications/papers/fmsd11.pdf

Theories about Data Structures

- So far, we only considered first-order theories involving numbers and arithmetic
- There are also theories that formalize data structures used in programming
- We'll look at one example: theory of arrays

Theory of Arrays

Signature

$$\Sigma_A \triangleq \{\cdot[\cdot],\ \langle\cdot \lhd \cdot\rangle,\ =\}$$

- *a[i]* binary function: read array *a* at index *i*, (“read(a,i)“)
- $a\langle i \lhd v\rangle$ ternary function: write value of *v* to array *a* at index *i*, (“write(a,i,v)“)
- $a\langle i \lhd v\rangle$ represents the resulting array after writing value v at index *i* in *a*

Theory of Arrays

Example:

- $a[3]=2$ “the value of array *a* at position *3* is *2*“
- $a\langle 3 \lhd 5\rangle[3]=5$ “if we set position *3* of a to *5* and then read *3*, the result is *5*“
- $a\langle 3 \lhd 5\rangle[3]=3$ “if we set position *3* of a to *5* and then read *3*, the result is *3*“
- $a[3]=2 \land a\langle 3 \lhd 5\rangle[3]=5$

Quiz:

- According to the usual semantics of array read and write, which formula is valid/satisfiable/unsatisfiable?

Theory of Arrays: Axioms

- To get the intended semantics of array reads and writes, we need to provide axioms
- Axioms include reflexivity, symmetry, transitivity of =
- In addition, we get the following axioms:

1. $\forall a.\forall i.\forall j.\ i=j \rightarrow a[i]=a[j]$ *(array congruence)*
2. $\forall a.\forall v.\forall i.\forall j.\ i=j \rightarrow a\langle i \lhd v\rangle[j] = v$ *(array update 1)*
3. $\forall a.\forall v.\forall i.\forall j.\ i\neq j \rightarrow a\langle i \lhd v\rangle[j] = a[j]$ *(array update 2)*

Theory of Arrays: Axioms

Quiz:

- Is the following T_A formula valid?
- $F \triangleq a[i] = e \rightarrow (\forall j.\ a\langle i \triangleleft e\rangle[j] = a[j])$

Theory of Arrays: Axioms

Quiz:

- Is the following T_A formula valid?
- $F \triangleq a[i] = e \rightarrow (\forall j.\ a\langle i \triangleleft e\rangle[j] = a[j])$
- Yes! We overwrite *i* with its old value, so *a* doesn't change
- Let's prove this via the semantic argument method
- We are allowed to use theory axioms
- As before, we start by assuming that $M ,\sigma \not\models F$

Theory of Arrays: Axioms

Example: $F \triangleq a[i] = e \rightarrow (\forall j.\ a\langle i \lhd e\rangle[j] = a[j])$

- Start: assume there exist M,σ such that $M,\sigma \not\models F$

1. $\forall a. \forall i. \forall j.\ i=j \rightarrow a[i]=a[j]$ *(array congruence)*
2. $\forall a. \forall v. \forall i. \forall j.\ i=j \rightarrow a\langle i \lhd v\rangle[j] = v$ *(array update 1)*
3. $\forall a. \forall v. \forall i. \forall j.\ i\neq j \rightarrow a\langle i \lhd v\rangle[j] = a[j]$ *(array update 2)*

Theory of Arrays: Decidability

- The full theory of arrays if not decidable.
- The quantifier-free fragment of TA is decidable.
- But, the quantifier-free fragment not sufficiently expressive in many contexts
- Thus, people have studied other richer fragments that are still decidable.
- Example: array property fragment (disallows nested arrays, restrictions on where quantified variables can occur)
- See also: http://theory.stanford.edu/∼arbrad/papers/arrays.ps

Combinations of Theories

- So far, we only talked about individual first-order theories
- Examples: $T_=$, T_{PA}, T_Z, T_A, . . .
- But in many applications, we need combined reasoning about several of these theories

Example:

- The formula $f(x) + 3 = y$ isn't a well-formed formula in any individual theory, but belongs to combined theory $T_Z \cup T_=$.

Deciding Combined Theories

- Given decision procedures for individual theories T_1 and T_2, can we decide satisfiability of formulas in $T_1 \cup T_2$?
- In the early 80s, Nelson and Oppen showed this is possible
- Specifically, if
  - the quantifier-free fragment of T_1 is decidable
  - the quantifier-free fragment of T_2 is decidable
  - T_1 and T_2 meet certain technical requirements
- the quantifier-free fragment of $T_1 \cup T_2$ is also decidable
- Nelson and Oppen's technique also shows how to combine decision procedures for T_1 and T_2 into a procedure for deciding $T_1 \cup T_2$

Where are we?

- Logic as the language of computation
- Decidable fragments of FOL that allow encoding interesting questions about programs (SMT-solvers!)
- From now, assume we have an SMT solver like Z3 (https://github.com/Z3Prover/z3) to satisfiability/validity of SMT formulas in relevant theories
- Examples QF_LIA, QF_LIA_UF, QF_Array
- Next, we show how to use SMT to encode proofs about programs using Floyd/Hoare logic