CS 512, Spring 2017, Handout 30

Limits of Formal Modeling in Propositional Logic and First-Order Logic

Assaf Kfoury

April 18, 2017

TWO KINDS OF LIMITS

- **Complexity limits** :

- **Expressiveness limits** :

TWO KINDS OF LIMITS

- **Complexity limits**:
  - Many properties we want to verify can be formally expressed in propositional logic and/or first-order logic and/or in some fragments thereof.
  - In most real-world cases, however, they cannot be analyzed or verified by hand and we need to rely on automated or semi-automated tools.
  - But before we do this, we need to know their complexity which, in some cases, may be beyond the most powerful tools currently available.
  - Our formalization of a property may turn out to be such that its verification is **feasible** (*e.g.*, *linear* or *low-degree polynomial* time) or **unfeasible** (*e.g.*, *exponential* or *double-exponential* time or worse) – or even **undecidable**.
- **Expressiveness limits**:

TWO KINDS OF LIMITS

- **Complexity limits**:
  - Many properties we want to verify can be formally expressed in propositional logic and/or first-order logic and/or in some fragments thereof.
  - In most real-world cases, however, they cannot be analyzed or verified by hand and we need to rely on automated or semi-automated tools.
  - But before we do this, we need to know their complexity which, in some cases, may be beyond the most powerful tools currently available.
  - Our formalization of a property may turn out to be such that its verification is **feasible** (*e.g.*, *linear* or *low-degree polynomial* time) or **unfeasible** (*e.g.*, *exponential* or *double-exponential* time or worse) – or even **undecidable**.
- **Expressiveness limits**:
  - What is the "weakest" or "least expressive" logic (*e.g.*, propositional logic, or a fragment of it, in preference to first-order logic) in which we can formally express a given property?
  - What are (realistic) examples of properties that are not expressible in first-order logic, let alone propositional logic?
  - Are there **tradeoffs** between expressiveness and complexity? A "strong" or "more expressive" logic generally gives rise to formalizations of properties that are more difficult to verify, but not always.

PIGEON-HOLE PRINCIPLE (PHP) once more

- We already studied PHP in propositional logic – in the handout *Formal Modeling with Propositional Logic* (click here).[1] We defined propositional WFF's:

  $$\varphi_2, \varphi_3, \ldots, \varphi_n, \ldots,$$

  where φ_n expresses PHP_n, *i.e.*, PHP for the case of n pigeons.

[1] Reminder of what the PHP says: "If n pigeons sit in $(n-1)$ holes, then some hole contains more than one pigeon" .

PIGEON-HOLE PRINCIPLE (PHP) once more

- We already studied PHP in propositional logic – in the handout *Formal Modeling with Propositional Logic* (click here).[1] We defined propositional WFF's:

  $$\varphi_2, \varphi_3, \ldots, \varphi_n, \ldots,$$

  where φ_n expresses PHP_n, *i.e.*, PHP for the case of n pigeons.

- We now want a first-order sentence Ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Every structure $\mathcal{M}_n$ of the form $(\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is a model of Ψ and the interpretation of Ψ in $\mathcal{M}_n$ expresses PHP_n.

[1] Reminder of what the PHP says: "If n pigeons sit in $(n-1)$ holes, then some hole contains more than one pigeon" .

PIGEON-HOLE PRINCIPLE (PHP) once more

- We already studied PHP in propositional logic – in the handout *Formal Modeling with Propositional Logic* (click here).[1] We defined propositional WFF's:

  $$\varphi_2, \varphi_3, \ldots, \varphi_n, \ldots,$$

  where φ_n expresses PHP_n, *i.e.*, PHP for the case of n pigeons.

- We now want a first-order sentence Ψ in the signature $\Sigma = \{R, c\}$ where R is a binary relation symbol and c is a constant symbol, such that:

  Every structure $\mathcal{M}_n$ of the form $(\{1, 2, \ldots, n\}, R^{\mathcal{M}_n}, c^{\mathcal{M}_n})$ is a model of Ψ and the interpretation of Ψ in $\mathcal{M}_n$ expresses PHP_n.

- Here is a possible first-order formulation of ψ:

$$\Psi \triangleq \big(\forall x\, \exists y\, R(x,y)\big) \wedge \big(\forall x\, \neg R(x,c)\big) \rightarrow \exists v\, \exists w\, \exists y\big(\neg(v \doteq w) \wedge R(v,y) \wedge R(w,y)\big)$$

  Note: If $\big(\forall x\, \neg R(x,c)\big)$ is omitted to obtain a new sentence Ψ', there is a structure $\mathcal{M}_n$ satisfying $\big(\forall x\, \exists y\, R(x,y)\big)$ but not $\exists v\, \exists w\, \exists y\big(\neg(v \doteq w) \wedge R(v,y) \wedge R(w,y)\big)$, in which case $\mathcal{M}_n \not\models \Psi'$ and PHP_n is not enforced in $\mathcal{M}_n$.

- Advantage of a first-order formulation over a propositional formulation : one first-order WFF Ψ instead of infinitely many propositional WFF's $\{\varphi_2, \varphi_3, \ldots\}$

[1] Reminder of what the PHP says: "If n pigeons sit in $(n-1)$ holes, then some hole contains more than one pigeon" .

PIGEON-HOLE PRINCIPLE (PHP) once more

- **Exercise**:
  1. Translate Ψ into a propositional WFF ψ_n which depends on an additional parameter $n \geqslant 2$. (Ψ represents an infinite family of propositional WFF's, one ψ_n for every $n \geqslant 2$.)
     Hint: Consider replacing every "$\forall$" by a "$\bigwedge$" and every "$\exists$" by a "$\bigvee$".
  2. Compare φ_n and ψ_n.
     Hint: They are very close to each other.
- **Exercise**:
  1. Use an automated proof-assistant (*e.g.*, Isabelle, Coq, etc.) to establish that Ψ is valid.
  2. Use a SAT solver to establish that each of φ_2, φ_3, and φ_4 is valid.
  3. Compare the performances in part 1 and part 2.

PIGEON-HOLE PRINCIPLE (PHP) once more

- **Exercise**:
  1. Translate Ψ into a propositional WFF ψ_n which depends on an additional parameter $n \geqslant 2$. (Ψ represents an infinite family of propositional WFF's, one ψ_n for every $n \geqslant 2$.)
     Hint: Consider replacing every "$\forall$" by a "$\bigwedge$" and every "$\exists$" by a "$\bigvee$".
  2. Compare φ_n and ψ_n.
     Hint: They are very close to each other.
- **Exercise**:
  1. Use an automated proof-assistant (*e.g.*, Isabelle, Coq, etc.) to establish that Ψ is valid.
  2. Use a SAT solver to establish that each of φ_2, φ_3, and φ_4 is valid.
  3. Compare the performances in part 1 and part 2.
- **Fact**: A resolution proof of φ_n or ψ_n is possible but does not help (bad news!). More precisely, any resolution proof of φ_n or ψ_n has size at least $\Omega(2^n)$.
- **Fact**: There are proofs of φ_n and ψ_n using what is called extended resolution (not covered this semester) which have size $\mathcal{O}(n^4)$.
- **Fact**: There are Hilbert-style proofs (not covered this semester) of φ_n and ψ_n which have size at most $\mathcal{O}(n^{20})$ (not really good news!).

how strong is first-order logic?

- Two very similar first-order sentences:

$$\theta_1 \triangleq \forall x \exists y \left(x < y \ \wedge \ prime(y) \ \wedge \ prime(y+2) \right)$$
$$\theta_2 \triangleq \forall x \exists y \left(\neg(x \doteq 0) \ \rightarrow \ (x < y) \ \wedge \ (y \leqslant 2 \times x) \ \wedge \ prime(y) \right)$$

  both to be interpreted in the structure $\mathcal{N} \triangleq (\mathbb{N}; \times, +, 0, 1)$ and where $prime(\ )$ is a unary predicate that tests whether its argument is a prime number. $prime(\ )$ is first-order definable in $\mathcal{N}$.

- θ_1 formally expresses the *Twin-Prime Conjecture*, a long-standing open problem.
- θ_2 formally expresses the *Bertrand-Chebyshev Conjecture*, which was shown to be true – by hand, before digital computers were invented![2]
- In recent years, formal proofs of θ_2 have been produced in several automated proof assistants (Isabelle, Coq, Metamath, Mizar, and perhaps others of which I am not aware), though all beyond the scope of this semester.

[2] A nice history of the *Bertrand-Chebyshev Conjecture* and its generalizations, and their increasingly simpler proofs, are presented in a Wikipedia article (click here).

how strong is first-order logic?

Theorem

1. *The validity problem of first-order logic*[3] *is semi-decidable but not decidable.*
2. *The unsatisfiability problem of first-order logic is semi-decidable but not decidable.*

Proof.

1. Different ways of proving the semi-decidability of the validity problem. One way: Gilmore's algorithm in Handout 25 is a semi-decision procedure (details left to you).

   One proof of the undecidability is in [LCS, page 133] which consists in reducing the (undecidable) PCP to the validity problem of FOL. A more direct proof of the undecidability reduces the Halting Problem for Turing machines to the validity problem of FOL (posted on the course website – click here).

2. This follows from part 1 because, for any first-order WFF φ,
   φ is valid iff $\neg\varphi$ is unsatisfiable.

   Exercise: Give another proof (not based on Gilmore's algorithm) for the semi-decidability of the validity problem of FOL.

□

[3] This is the decision problem that asks whether an arbitrary first-order WFF, or its universal closure as a sentence, is valid.

how strong is first-order logic?

Theorem (Skolem-Lowenheim)

1. *If φ is a first-order sentence such that, for every $n \geqslant 1$, there is a model of φ with at least n elements, then φ has an infinite model.*

   "First-order logic cannot enforce finiteness of models."

2. *If φ is a first-order sentence which has a model (i.e., φ is satisfiable), then φ has a model with a countable universe.*

   "First-order logic cannot enforce uncountable models."

Proof.

1. A proof is given in [LCS, page 138].
2. A proof is a simple variation on the proof of Lemma 24 in the handout *Compactness and Completeness of Propositional Logic and First-Order Logic* – click here. We omit the details.

□

how strong is first-order logic?

Theorem

There is no first-order WFF $\psi(x, y)$ with two free variables x and y, over the signature $\{R, \doteq\}$ where R is a binary predicate symbol, such that for every graph model $\mathcal{M} = (M, R^{\mathcal{M}})$ and every $a, b \in M$, it holds that:

$$\mathcal{M}, a, b \models \psi \quad \text{iff there is a path from } a \text{ to } b$$

"Reachibility in graphs is not first-order definable."

Proof.

One possible proof is in [LCS, page 138]. □