

Logic in Computing

Two important dimensions :

1) Modeling. What do we want to reason about?

Example: Reason about computer code (programs) and its execution behavior.
How to model "behavior", e.g.
a sequence of machine configurations?

2) Expressivity. What kinds of claims and distinctions do we want to make about our models?

Example: A program never drives the machine into a "bad" configuration.

Models for logic within computing are often finite.

Finite Model Theory: Logic carried out over classes of finite structures.

Example: Reasoning over finite words, with words modeling abstractions of program execution traces

Example: The input to an algorithm is finite. And descriptive complexity studies computational complexity classes through the lens of the expressive power of logics which define their languages.

Fagin's Theorem: The complexity class NP is exactly captured by $\exists SO$ (existential second-order logic)

$\forall L \in NP. \exists \varphi \in \exists SO. w \models \varphi \text{ iff } w \in L. \text{ And...}$

... $\forall \varphi \in \exists SO. \{w : w \models \varphi\} \in NP.$

Some results for FOL over finite structures

Trakhtenbrodt's Theorem (1950):

The validity problem for FOL over finite structures is undecidable.

Finite Validity Problem: Given sentences Γ and sentence φ , determine whether every finite model of Γ is also a model of φ .

Note: Satisfiability over finite structures is r.e.

Idea: Enumerate finite structures and check if one satisfies φ .

Corollary: FOL validity over finite structures is not even r.e.

Compactness fails for classes of finite models

Claim: There is an infinite set of FOL sentences that is finitely-satisfiable but not satisfiable over a class of finite structures.

Fix an empty signature (just assume equality).
Our class of structures consists of finite sets.

Idea: express, for fixed n , that the set has size at least n .

$$\varphi_n := \exists x_1 \dots x_n \cdot \bigwedge_{i \neq j \in [n]} x_i \neq x_j$$

Then $\Gamma := \{ \varphi_n \}_{n \geq 1}$ is finitely-sat over finite models but is not satisfied by any finite model.

Suppose we want to reason about the expressive power of a symbolic language.

What tools are available?

Within FOL, the compactness theorem (and the Löwenheim-Skolem Theorem) has been used to prove some concepts are not expressible.

You saw an instance of this in the exercise proving that connectivity in graphs is not expressible.

But we just proved that compactness fails for FOL over finite models.

How can we reason about inexpressivity of languages over finite structures?

Ehrenfeucht - Fraïssé Games (EF Games)

Andrzej Ehrenfeucht (founding member of CS dept. at CU-Boulder)

2 player games between "Spoller" and "Duplicator".

Game played over two finite structures A, B .

In round i , Spoller picks an element of either A or B and Duplicator responds by picking an element of the other structure.

Duplicator's goal is to pick elements which "look" identical to the elements picked by A .

Example: Consider two sets A and B
 with $|A| = k$ and $|B| = k$, with constants $a, b \in \mathcal{I}$

d.	1	a, b	$1'$
	2		$2' \quad a \quad s_1$
	3		$3' \quad b \quad s_2$
	$\vdots$		$\vdots$
	k		k'
	A		B

Duplicator is stuck by the second move.

Spoiler has winning strategy in the games
 with at least 2 rounds, but **Duplicator** survives
 for 1 round.

$$B \models \exists x \exists y. x = a \wedge y = b \wedge x \neq y$$

Theorem (Ehrenfeucht-Fraïssé): Let A and B be
 structures in a relational vocabulary. TFAE:

- 1) A and B agree on FOL formulas of quantifier rank k .
- 2) Duplicator has winning strategy in the k round game