Some Turing-Complete Extensions of First-Order Logic

Antti Kuusisto

Technical University of Denmark
and
Unversity of Wrocław

D^*

Extend FO as follows.

- Add dependence, independence, inclusion and exclusion atoms to the language.
- Add the formula formation rule $\varphi \mapsto \mathrm{I}y\,\varphi$.

$\mathfrak{A}, X \models \mathrm{I}y\varphi$ iff there is a finite nonempty set S of fresh elements such that

$$\mathfrak{A} + S,\, X[S/y] \models \varphi.$$

D^*

Theorem
D^* *captures* RE.

Proof D^* is contained in RE: given a sentence φ of D^*, construct a nondeterministic Turing machine that first guesses for each subformula $\mathrm{I}y\,\psi$ a finite cardinality to be added to the input model, and then checks if φ is satisfied when the guessed cardinalities are used.

Define a predicate logic that extends ESO and captures RE. Show that the predicate logic translates into D^*.

The language of $\mathcal{L}_{\mathrm{RE}}$ consists of formulae $\mathrm{I}Y\,\psi$, where ψ is a formula of ESO.

$\mathfrak{A} \models \mathrm{I}Y\,\psi$ iff there exists a finite nonempty set S such that

- $S \cap A = \emptyset$
- $\mathfrak{A} + S, Y \mapsto S \models \psi$.

Theorem

$\mathcal{L}_{\mathrm{RE}}$ *captures* RE.

Proof.

Let TM be a Turing machine. It is routine to write a formula $\mathrm{I}Y\overline{\exists Z}\,\beta$ such that $\mathfrak{A} \models \mathrm{I}Y\overline{\exists Z}\,\beta$ iff there exists a model $\mathfrak{A} + \mathfrak{C}$, where $\mathfrak{C}$ encodes the computation table of an accepting computation of TM on the input $enc(\mathfrak{A})$.

For the converse, given a sentence $\mathrm{I}Y\,\delta$ of $\mathcal{L}_{\mathrm{RE}}$, we can write a Turing machine that first non-deterministically provides a number of fresh points n to be added to an input model $\mathfrak{A}$, and then checks if δ holds in the extended model. □

Let D^+ denote D^* without operators I. Assume we have a translation $\mathbb{T}_Y^y$ from dependence logic into D^+ such that

$$(\mathfrak{M}, Y \mapsto S), \{\emptyset\} \models \varphi \text{ iff } \mathfrak{M}, \{\emptyset\}[S/y] \models \mathbb{T}_Y^y(\varphi).$$

Then we are done. Let $(\cdot)^\#$ denote the translation from ESO into dependence logic. We have

$$\begin{aligned}
\mathfrak{A} \models \mathrm{I}Y\overline{\exists X}\psi &\Leftrightarrow \big(\mathfrak{A} + S, Y \mapsto S\big) \models \overline{\exists X}\psi \text{ for some } S \\
&\Leftrightarrow \big(\mathfrak{A} + S, Y \mapsto S\big), \{\emptyset\} \models \big(\overline{\exists X}\psi\big)^\# \text{ for some } S \\
&\Leftrightarrow \mathfrak{A} + S, \{\emptyset\}[S/y] \models \mathbb{T}_Y^y\big(\big(\overline{\exists X}\psi\big)^\#\big) \text{ for some } S \\
&\Leftrightarrow \mathfrak{A}, \{\emptyset\} \models \mathrm{I}y\, \mathbb{T}_Y^y\big(\big(\overline{\exists X}\psi\big)^\#\big)
\end{aligned}$$

1. $(Y(x))^* := x \subseteq y$
2. $(\neg Y(x))^* := x|y$
3. $\varphi^* := \varphi$ for other literals φ.
4. $(\varphi \wedge \psi)^* := \varphi^* \wedge \psi^*$
5. $(\varphi \vee \psi)^*$

$$:= \exists v(\; v \perp_{\overline{z}} y \wedge \big((\varphi^* \wedge v = u) \vee (\psi^* \wedge v = u') \big) \;),$$

6. $(\exists x\, \varphi)^* := \exists x \big(x \perp_{\overline{z}} yv \wedge \varphi^* \big),$
7. $(\forall x\, \varphi)^* := \forall x\, (\varphi^*)$

$$\mathbb{T}^y_Y(\varphi) := \exists u \exists u' \big(u \neq u' \wedge =\!(u) \wedge =\!(u') \wedge \varphi^* \big).$$

□

Extend FO by operators that

1. allow addition of fresh points to the domain,
2. enable recusive looping when playing the semantic game.

Leads to a Turing-complete logic $\mathcal{L}$ with a game-theoretic semantics.

Logic $\mathcal{L}$

Syntax: extend FO by the following constructs:

1. $\mathrm{I}x\,\varphi$
2. $\mathrm{I}Rx_1, ..., x_k\,\varphi$
3. $\mathrm{D}Rx_1, ..., x_k\,\varphi$
4. $k\,\varphi$, where $k \in \mathbb{N}$.
5. If k is (a symbol representing) a natural number, then k is an atomic formula.

Game-theoretic semantics

Extend the game-theoretic semantics of first-order logic.

In a position $(\mathfrak{A}, f, \#, \mathrm{I}x\,\varphi)$, the domain is extended by one new isolated point u. The play continues from the position $(\mathfrak{A} \cup \{u\}, f, \#, \varphi)$.

Game-theoretic semantics

- In a position $(\mathfrak{A}, f, +, \mathrm{I}Rx_1, ..., x_k\, \varphi)$, the player $\exists$ chooses a k-tuple $(u_1, ..., u_k)$. The play continues from the position $(\mathfrak{A}^*, f^*, +, \varphi)$, where
  - $f^* = f[x_1 \mapsto u_1, ..., x_k \mapsto u_k]$,
  - $\mathfrak{A}^*$ is $\mathfrak{A}$ with the tuple $(u_1, ..., us_k)$ added to R.
- In a position $(\mathfrak{A}, f, -, \mathrm{I}Rx_1, ..., x_k\, \varphi)$, the player $\forall$ chooses a k-tuple $(u_1, ..., u_k)$. The play continues from te position $(\mathfrak{A}^*, f^*, -, \varphi)$.
- The operator $\mathrm{D}Rx_1, ..., x_k$ is similar to $\mathrm{I}Rx_1, ..., x_k$, but a tuple is deleted rather than added.

Game-theoretic semantics

- If a position $(\mathfrak{A}, f, +, k)$ is reached, where $k \in \mathbb{N}$, then the player $\exists$ chooses a subformula $k\psi$ of the original formula the game begun with. The play continues from the position $(\mathfrak{A}, f, +, \psi)$.
- If a position $(\mathfrak{A}, f, -, k)$ is reached, then the play continues as above, but the player $\forall$ makes the choice.
- If a position $(\mathfrak{A}, f, \#, k\varphi)$ is reached, the game continues from the position $(\mathfrak{B}, f, \#, \varphi)$.

Game-theoretic semantics

- The game is played for at most ω rounds.
- A play can be won only by reaching a first-order atom.
- The winning conditions are exactly as in FO.

We write $\mathfrak{A}, f \models^{+} \varphi$ iff $\exists$ has a winning strategy in the game $G(\mathfrak{A}, f, +, \varphi)$.

$\mathfrak{A}, f \models^{-} \varphi$ iff $\forall$ has a winning strategy in the game $G(\mathfrak{A}, f, +, \varphi)$.

Turing-completeness

Theorem

Let τ be a nonempty vocabulary. Let TM be a Turing machine that operates on encodings of finite τ-models. Then there exists a sentence φ of $\mathcal{L}$ such that the following conditions hold for every finite τ-model $\mathfrak{A}$.

1. TM *accepts* *enc*$(\mathfrak{A})$ *iff* $\mathfrak{A} \models^{+} \varphi$.
2. TM *rejects* *enc*$(\mathfrak{A})$ *iff* $\mathfrak{A} \models^{-} \varphi$.

Proof sketch.

The formula φ is essentially of the type

$$1\Big(\bigwedge_{instr \in I} \psi_{instr} \Big),$$

where

- I is the set of instructions of TM.
- The computation of TM is encoded using word models that encode the machine tape contents.
- The word models are built by adding new points and adding new tuples to relations.
- The state and head position of TM are encoded by using variable symbols x, whose interpretation can be dynamically altered using quantification.
- Let *instr* lead to a non-final state. The ψ_{instr} is of the type

$$(\psi_{state} \wedge \psi_{tape\_position}) \rightarrow (\psi_{new\_state} \wedge \psi_{new\_tape\_position} \wedge 1)$$

- Let *instr* lead to an accepting final state. The ψ_{instr} is of the type
$$\left(\psi_{state} \wedge \psi_{tape\_position}\right) \ \rightarrow \ \top.$$
- Let *instr* lead to a rejecting final state. The ψ_{instr} is of the type
$$\left(\psi_{state} \wedge \psi_{tape\_position}\right) \ \rightarrow \ \bot.$$

□

Turing-completeness

Theorem

Let τ be a nonempty vocabulary. Let φ be a sentence of $\mathcal{L}$. Then there exists a Turing machine TM such that the following conditions hold for every finite τ-model $\mathfrak{A}$.

1. TM *accepts* *enc*$(\mathfrak{A})$ *iff* $\mathfrak{A} \models^{+} \varphi$.
2. TM *rejects* *enc*$(\mathfrak{A})$ *iff* $\mathfrak{A} \models^{-} \varphi$.

Proof. TM non-deterministically provides a number $n \in \mathbb{N}$.

TM enumerates all plays of at most n moves.

TM accepts iff the player $\exists$ has a strategy that leads to a win in every play with up to n moves.

Importantly, $\exists$ cannot have a winning strategy that results in arbitrarily long plays. Assume the contrary.

Each position can have only finitely many successor positions. Thus by König's lemma, the game tree restricted to the strategy of $\exists$ has an infinite path. Thus the strategy of $\exists$ is not a winning strategy.

$\square$