Decidibilidad y Expresividad

Pedro Cabalar

Lógica
Grado en Inteligencia Artificial
Universidade da Coruña

April 8, 2025

1 Undecidability and Expressiveness

2 Some usual extensions
- Equality
- Arithmetics

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church).

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church). That is, given φ there is no program that decides (i.e., answers 'yes' or 'no') whether $\models \varphi$ in a finite number of steps.

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church). That is, given φ there is no program that decides (i.e., answers 'yes' or 'no') whether $\models \varphi$ in a finite number of steps.
- Consequence 1: satisfiability is also undecidable, since φ satisfiable iff not $\models \neg\varphi$.

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church). That is, given φ there is no program that decides (i.e., answers 'yes' or 'no') whether $\models \varphi$ in a finite number of steps.
- Consequence 1: satisfiability is also undecidable, since φ satisfiable iff not $\models \neg\varphi$.
- Consequence 2: provability $\vdash \varphi$ is also undecidable, since $\vdash \varphi$ iff $\models \varphi$.

Soundness, Completeness and Undecidability

- Predicate calculus is sound and complete (Gödel): $\Gamma \models \alpha$ iff $\Gamma \vdash \alpha$.
- However, validity in predicate calculus is undecidable (Church). That is, given φ there is no program that decides (i.e., answers 'yes' or 'no') whether $\models \varphi$ in a finite number of steps.
- Consequence 1: satisfiability is also undecidable, since φ satisfiable iff not $\models \neg\varphi$.
- Consequence 2: provability $\vdash \varphi$ is also undecidable, since $\vdash \varphi$ iff $\models \varphi$.
- Still, some fragments of Predicate Calculus are known to be decidable.

Some decidable fragments of FOL

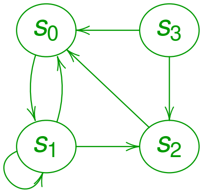
- Monadic predicate calculus (only 1-ary predicates)
- The class with prefix $\exists^*\forall^*$
- The class with prefix $\exists^*\forall\exists^*$
- The class with prefix $\exists^*\forall\forall\exists^*$ (no equality axioms)
- The class with two variables at most (Description Logics)
- Guarded Predicate Calculus:

$$\exists \overline{y} (\ \alpha(\overline{x}, \overline{y}) \wedge \varphi(\overline{x}, \overline{y}\ )$$
$$\forall \overline{y} (\ \alpha(\overline{x}, \overline{y}) \rightarrow \varphi(\overline{x}, \overline{y}\ )$$

where α atomic and including all the free variables of φ.

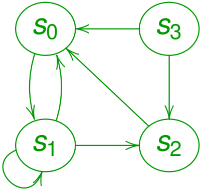
Expressiveness

- Example: let G be a graph with vertices S and edges E. For instance S could represent states $\{s_0, s_1, s_2, s_3\}$ and E transitions among them like in:



Expressiveness

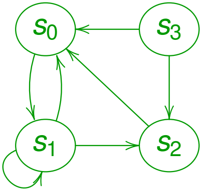
- Example: let G be a graph with vertices S and edges E. For instance S could represent states $\{s_0, s_1, s_2, s_3\}$ and E transitions among them like in:



- Decision problem REACH (Graph reachability): given two vertices $u, v \in V$, can we find a finite path from u to v in G?

Expressiveness

- Example: let G be a graph with vertices S and edges E. For instance S could represent states $\{s_0, s_1, s_2, s_3\}$ and E transitions among them like in:



- Decision problem REACH (Graph reachability): given two vertices $u, v \in V$, can we find a finite path from u to v in G?
- Since REACH is a decision problem, perhaps we can try to represent it as FOL-satisfiability of some formula $\varphi_{REACH}(u, v)$.

Expressiveness

- We use predicate $R(x, y)$ to represent edges and free variables u, v to represent the nodes to check.
- Given any graph G, we have its corresponding model $I(G)$. We look for a formula $\varphi_{REACH}(u, v)$ such that G has a finite path from u to v iff $I(G) \models \varphi_{REACH}(u, v)$.

Expressiveness

- We use predicate $R(x,y)$ to represent edges and free variables u, v to represent the nodes to check.
- Given any graph G, we have its corresponding model $I(G)$. We look for a formula $\varphi_{REACH}(u,v)$ such that G has a finite path from u to v iff $I(G) \models \varphi_{REACH}(u,v)$.
- Trying to encode reachability as a formula ...

$$\begin{aligned}\varphi_{REACH}(u,v) \stackrel{def}{=} u = v \quad & \vee \quad \exists x (R(u,x) \wedge R(x,v)) \\ & \vee \quad \exists x_1 \exists x_2 (R(u,x_1) \wedge R(x_1,x_2) \wedge R(x_2,v)) \\ & \vee \quad \ldots\end{aligned}$$

But this is not a well-formed formula! (infinite disjunction)

Expressiveness

- We use predicate $R(x, y)$ to represent edges and free variables u, v to represent the nodes to check.
- Given any graph G, we have its corresponding model $I(G)$. We look for a formula $\varphi_{REACH}(u, v)$ such that G has a finite path from u to v iff $I(G) \models \varphi_{REACH}(u, v)$.
- Trying to encode reachability as a formula ...

$$\begin{aligned} \varphi_{REACH}(u, v) \stackrel{def}{=} u = v \quad & \vee \quad \exists x (R(u, x) \wedge R(x, v)) \\ & \vee \quad \exists x_1 \exists x_2 (R(u, x_1) \wedge R(x_1, x_2) \wedge R(x_2, v)) \\ & \vee \quad \ldots \end{aligned}$$

But this is not a well-formed formula! (infinite disjunction)
Can we find an equivalent well-founded formula?

Expressiveness

- We use predicate $R(x, y)$ to represent edges and free variables u, v to represent the nodes to check.
- Given any graph G, we have its corresponding model $I(G)$. We look for a formula $\varphi_{REACH}(u, v)$ such that G has a finite path from u to v iff $I(G) \models \varphi_{REACH}(u, v)$.
- Trying to encode reachability as a formula ...

$$\begin{aligned}\varphi_{REACH}(u,v) \stackrel{def}{=} u = v \quad & \vee \quad \exists x(R(u,x) \wedge R(x,v)) \\ & \vee \quad \exists x_1 \exists x_2 (R(u,x_1) \wedge R(x_1,x_2) \wedge R(x_2,v)) \\ & \vee \quad \ldots\end{aligned}$$

But this is not a well-formed formula! (infinite disjunction)
Can we find an equivalent well-founded formula? NO

Expressiveness

Theorem

There is no FOL-formula $\varphi_{REACH}(u, v)$ *depending on* R, u, v *such that there is a finite path from* u *to* v *in* G *iff* $I(G) \models \varphi_{REACH}(u, v)$.

Expressiveness

Theorem

There is no FOL-formula $\varphi_{REACH}(u, v)$ depending on R, u, v such that there is a finite path from u to v in G iff $I(G) \models \varphi_{REACH}(u, v)$.

- Two important properties:

Theorem (Compactness Theorem)

Let Γ be a set of sentences. If all finite subsets of Γ are satisfiable, then Γ is satisfiable.

Theorem (Löwenheim-Skolem Theorem)

If Γ has a model then it has a model with a countable domain.

Countable domain means: $|D| = |S|$ for some subset S of natural numbers (including the whole set too).

1. Undecidability and Expressiveness

2. Some usual extensions
   - Equality
   - Arithmetics

Outline

1 Undecidability and Expressiveness

2 Some usual extensions
- Equality
- Arithmetics

FOL with equality

- *$FOL_=$* : We have an (infix) binary predicate '$=$' whose meaning is fixed by the axiom schemata:

$$
\begin{aligned}
x = x & \\
x = y & \rightarrow f(\overline{z}, x, \overline{z'}) = f(\overline{z}, y, \overline{z'}) \\
x = y \wedge \varphi(x) & \rightarrow \varphi(y)
\end{aligned}
$$

for any variables x, y, tuples of variables $\overline{z}, \overline{z'}$, function symbol f and any formula φ.

FOL with equality

- *$FOL_=$* : We have an (infix) binary predicate '$=$' whose meaning is fixed by the axiom schemata:

$$
\begin{aligned}
x &= x \\
x = y &\rightarrow f(\overline{z}, x, \overline{z'}) = f(\overline{z}, y, \overline{z'}) \\
x = y \wedge \varphi(x) &\rightarrow \varphi(y)
\end{aligned}
$$

  for any variables x, y, tuples of variables $\overline{z}, \overline{z'}$, function symbol f and any formula φ.

- Symmetry and transitivity can be proved from the axioms above:

$$
\begin{aligned}
x = y &\rightarrow y = x \\
x = y \wedge y = z &\rightarrow x = z
\end{aligned}
$$

Sequent Calculus with equality

$$\frac{}{\Gamma \vdash \ t = t} \quad (= R)$$

$$\frac{\Gamma, s = t \ \vdash \ A[x/s]}{\Gamma, s = t \ \vdash \ A[x/t]} \quad (= L1)$$

$$\frac{\Gamma, s = t \ \vdash \ A[x/t]}{\Gamma, s = t \ \vdash \ A[x/s]} \quad (= L2)$$

Outline

1 Undecidability and Expressiveness

2 Some usual extensions
- Equality
- Arithmetics

Dedekind/Peano axioms

- We use $FOL_=$ and we have one constant 0, a unary function s (successor) and two (infix) binary functions $+$ and $\cdot$.

Dedekind/Peano axioms

- We use $FOL_=$ and we have one constant 0, a unary function s (successor) and two (infix) binary functions $+$ and $\cdot$.
- Each natural number n is represented by n nested applications of s to 0. Example: 5 is written $s(s(s(s(s(0)))))$ or just $s^5(0)$.

Dedekind/Peano axioms

- We use $FOL_=$ and we have one constant 0, a unary function s (successor) and two (infix) binary functions $+$ and $\cdot$.
- Each natural number n is represented by n nested applications of s to 0. Example: 5 is written $s(s(s(s(s(0)))))$ or just $s^5(0)$.
- Peano Arithmetics (PA) axioms: universal closure of

$$\begin{aligned}
\neg(0 &= s(x)) \\
s(x) = s(y) \rightarrow x &= y \\
x + 0 &= x \\
x + s(y) &= s(x+y) \\
x \cdot 0 &= 0 \\
x \cdot s(y) &= x \cdot y + x
\end{aligned}$$

plus the induction schema ...

Dedekind/Peano axioms

- Induction schema: contains a countably infinite set of axioms:

$$\begin{aligned} \forall \overline{y} \; (\quad & \varphi(0, \overline{y}) \wedge \\ & \forall x \; (\varphi(x, \overline{y}) \rightarrow \varphi(s(x), \overline{y}) \;) \\ & \rightarrow \forall x \; \varphi(x, \overline{y}) \;) \end{aligned}$$

for any formula $\varphi(x, \overline{y})$ with free variables x and (tuple) $\overline{y}$.

Dedekind/Peano axioms

- Induction schema: contains a countably infinite set of axioms:

$$\forall \overline{y} \; (\quad \varphi(0, \overline{y}) \wedge \\ \forall x \; (\varphi(x, \overline{y}) \rightarrow \varphi(s(x), \overline{y}) \;) \\ \rightarrow \forall x \; \varphi(x, \overline{y}) \;)$$

  for any formula $\varphi(x, \overline{y})$ with free variables x and (tuple) $\overline{y}$.

- Induction has a simpler encoding in second order logic:

$$\forall P \; (\; P(0) \; \wedge \; \forall x \; (P(x) \rightarrow P(s(x)) \;) \; \rightarrow \forall x \; P(x) \;)$$

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.
- By *recursive* we mean that it can be infinite, but effectively generated (for instance, by a computer program).

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.
- By *recursive* we mean that it can be infinite, but effectively generated (for instance, by a computer program). Otherwise, we could take the trivial axiomatisation = all the valid formulas!

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.
- By *recursive* we mean that it can be infinite, but effectively generated (for instance, by a computer program). Otherwise, we could take the trivial axiomatisation = all the valid formulas!
- It follows that there are valid formulas that are unprovable (in fact, there are infinitely many of them).

Gödel's first incompleteness theorem

- First incompleteness theorem: there is no *recursive* set of axioms for arithmetics that is both consistent and complete.
- By *recursive* we mean that it can be infinite, but effectively generated (for instance, by a computer program). Otherwise, we could take the trivial axiomatisation = all the valid formulas!
- It follows that there are valid formulas that are unprovable (in fact, there are infinitely many of them).
- The theorem can also be stated as: for a recursive, consistent set of axioms for arithmetics there are sentences such that neither φ nor $\neg\varphi$ has a proof.