SEMANTICS OF PREDICATE LOGIC

MODELS

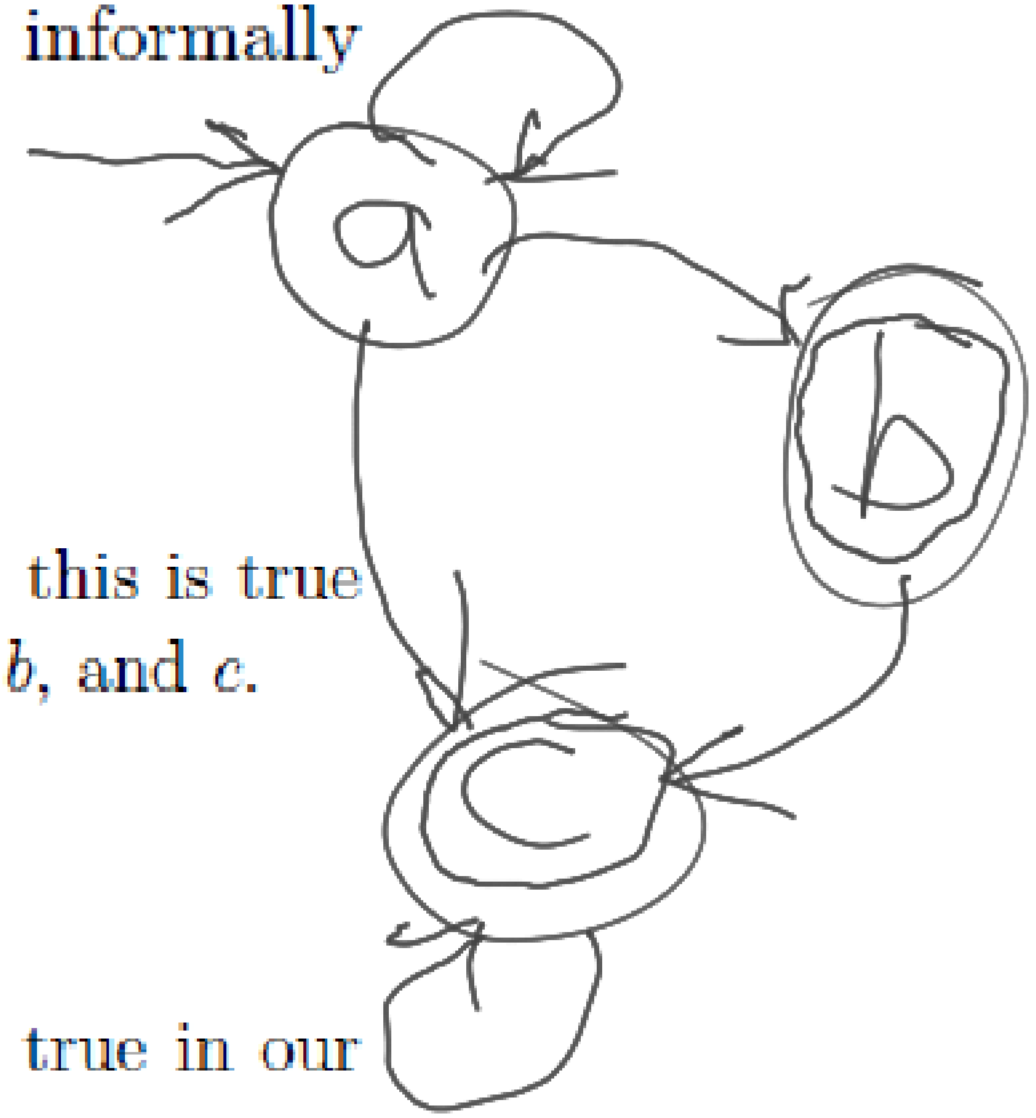
Definition 2.14 Let $\mathcal{F}$ be a set of function symbols and $\mathcal{P}$ a set of predicate symbols, each symbol with a fixed number of required arguments. A model $\mathcal{M}$ of the pair $(\mathcal{F}, \mathcal{P})$ consists of the following set of data:

1. A non-empty set A, the universe of concrete values;
2. for each nullary function symbol $f \in \mathcal{F}$, a concrete element $f^{\mathcal{M}}$ of A
3. for each $f \in \mathcal{F}$ with arity $n > 0$, a concrete function $f^{\mathcal{M}} : A^n \to A$ from A^n, the set of n-tuples over A, to A; and
4. for each $P \in \mathcal{P}$ with arity $n > 0$, a subset $P^{\mathcal{M}} \subseteq A^n$ of n-tuples over A.

$P^{\mathcal{M}} : A^n \longrightarrow \{T, F\}$

The distinction between f and $f^{\mathcal{M}}$ and between P and $P^{\mathcal{M}}$ is most important. The symbols f and P are just that: symbols, whereas $f^{\mathcal{M}}$ and $P^{\mathcal{M}}$ denote a concrete function (or element) and relation in a model $\mathcal{M}$, respectively.

Example 2.15 Let $\mathcal{F} \stackrel{\text{def}}{=} \{i\}$ and $\mathcal{P} \stackrel{\text{def}}{=} \{R, F\}$; where i is a constant, F a predicate symbol with one argument and R a predicate symbol with two arguments. A model $\mathcal{M}$ contains a set of concrete elements A – which may be a set of states of a computer program. The interpretations $i^{\mathcal{M}}$, $R^{\mathcal{M}}$, and $F^{\mathcal{M}}$ may then be a designated initial state, a state transition relation, and a set of final (accepting) states, respectively. For example, let $A \stackrel{\text{def}}{=} \{a, b, c\}$, $i^{\mathcal{M}} \stackrel{\text{def}}{=} a$, $R^{\mathcal{M}} \stackrel{\text{def}}{=} \{(a,a), (a,b), (a,c), (b,c), (c,c)\}$, and $F^{\mathcal{M}} \stackrel{\text{def}}{=} \{b, c\}$. We informally check some formulas of predicate logic for this model:



1. The formula

$$\exists y\, R(i, y)$$

   says that there is a transition from the initial state to some state; this is true in our model, as there are transitions from the initial state a to a, b, and c.
2. The formula

$$\neg F(i)$$

   states that the initial state is not a final, accepting state. This is true in our model as b and c are the only final states and a is the intitial one.
3. The formula

3. The formula

$$\forall x \forall y \forall z\, (R(x,y) \land R(x,z) \to y = z)$$

makes use of the equality predicate and states that the transition relation is deterministic: all transitions from any state can go to at most one state (there may be no transitions from a state as well). This is false in our model since state a has transitions to b and c.

4. The formula

$$\forall x \exists y\, R(x,y)$$

states that the model is free of states that deadlock: all states have a transition to some state. This is true in our model: a can move to a, b or c; and b and c can move to c.

Definition 2.17 A look-up table or environment for a universe A of concrete values is a function $l\colon \mathsf{var} \to A$ from the set of variables var to A. For such an l, we denote by $l[x \mapsto a]$ the look-up table which maps x to a and any other variable y to $l(y)$.

Definition 2.18 Given a model $\mathcal{M}$ for a pair $(\mathcal{F}, \mathcal{P})$ and given an environment l, we define the satisfaction relation $\mathcal{M} \vDash_l \phi$ for each logical formula ϕ over the pair $(\mathcal{F}, \mathcal{P})$ and look-up table l by structural induction on ϕ. If $\mathcal{M} \vDash_l \phi$ holds, we say that ϕ computes to T in the model $\mathcal{M}$ with respect to the environment l.

- P: If ϕ is of the form $P(t_1, t_2, \ldots, t_n)$, then we interpret the terms $t_1, t_2, \ldots, t_n$ in our set A by replacing all variables with their values according to l. In this way we compute concrete values $a_1, a_2, \ldots, a_n$ of A for each of these terms, where we interpret any function symbol $f \in \mathcal{F}$ by $f^{\mathcal{M}}$. Now $\mathcal{M} \vDash_l P(t_1, t_2, \ldots, t_n)$ holds iff $(a_1, a_2, \ldots, a_n)$ is in the set $P^{\mathcal{M}}$.
- $\forall x$: The relation $\mathcal{M} \vDash_l \forall x\, \psi$ holds iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for all $a \in A$.
- $\exists x$: Dually, $\mathcal{M} \vDash_l \exists x\, \psi$ holds iff $\mathcal{M} \vDash_{l[x \mapsto a]} \psi$ holds for some $a \in A$.
- $\neg$: The relation $\mathcal{M} \vDash_l \neg\psi$ holds iff it is not the case that $\mathcal{M} \vDash_l \psi$ holds.
- $\vee$: The relation $\mathcal{M} \vDash_l \psi_1 \vee \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_1$ or $\mathcal{M} \vDash_l \psi_2$ holds.
- $\wedge$: The relation $\mathcal{M} \vDash_l \psi_1 \wedge \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_1$ and $\mathcal{M} \vDash_l \psi_2$ hold.
- $\rightarrow$: The relation $\mathcal{M} \vDash_l \psi_1 \rightarrow \psi_2$ holds iff $\mathcal{M} \vDash_l \psi_2$ holds whenever $\mathcal{M} \vDash_l \psi_1$ holds.

We sometimes write $\mathcal{M} \nvDash_l \phi$ to denote that $\mathcal{M} \vDash_l \phi$ does not hold.

Definition 2.20 Let Γ be a (possibly infinite) set of formulas in predicate logic and ψ a formula of predicate logic.

1. Semantic entailment $\Gamma \vDash \psi$ holds iff for all models $\mathcal{M}$ and look-up tables l, whenever $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$, then $\mathcal{M} \vDash_l \psi$ holds as well.
2. Formula ψ is satisfiable iff there is some model $\mathcal{M}$ and some environment l such that $\mathcal{M} \vDash_l \psi$ holds.
3. Formula ψ is valid iff $\mathcal{M} \vDash_l \psi$ holds for all models $\mathcal{M}$ and environments l in which we can check ψ.
4. The set Γ is consistent or satisfiable iff there is a model $\mathcal{M}$ and a look-up table l such that $\mathcal{M} \vDash_l \phi$ holds for all $\phi \in \Gamma$.

Example 2.21 The justification of the semantic entailment

$$\forall x\,(P(x) \to Q(x)) \vDash \forall x\,P(x) \to \forall x\,Q(x)$$

is as follows. Let $\mathcal{M}$ be a model satisfying $\forall x\,(P(x) \to Q(x))$. We need to show that $\mathcal{M}$ satisfies $\forall x\,P(x) \to \forall x\,Q(x)$ as well. On inspecting the definition of $\mathcal{M} \vDash \psi_1 \to \psi_2$, we see that we are done if not every element of our model satisfies P. Otherwise, every element does satisfy P. But since $\mathcal{M}$ satisfies $\forall x\,(P(x) \to Q(x))$, the latter fact forces every element of our model to satisfy Q as well. By combining these two cases (i.e. either all elements of $\mathcal{M}$ satisfy P, or not) we have shown that $\mathcal{M}$ satisfies $\forall x\,P(x) \to \forall x\,Q(x)$.

What about the converse of the above? Is

Consider a model $\mathcal{M}$
$A = \{a, b\}$ $P^{\mathcal{M}} = \{a\}$
$Q^{\mathcal{M}} = \{\}$

$$\underbrace{\forall x\,P(x) \to \forall x\,Q(x)}_{T} \vDash \underbrace{\forall x\,(P(x) \to Q(x))}_{F}$$

because $P(a) \to Q(a)$ is false

~~construct a counter-example model. Let $A' \stackrel{\text{def}}{=} \{a, b\}$, $P^{\mathcal{M}'} \stackrel{\text{def}}{=} \{a\}$ and $Q^{\mathcal{M}'} \stackrel{\text{def}}{=} \{b\}$. Then $\mathcal{M}' \vDash \forall x\,P(x) \to \forall x\,Q(x)$ holds, but $\mathcal{M}' \vDash \forall x\,(P(x) \to Q(x))$ does not.~~

UNDECIDABILITY OF PREDICATE LOGIC

Validity in predicate logic. Given a logical formula ϕ in predicate logic, does $\vDash \phi$ hold, yes or no?

Turing Machine - Alan Turing

Lambda Calculus - Alonzo Church

→ LISP, Scheme — Functional programming languages

C, Python - Imperative language

Diagonalization argument

REDUCTION:

$A \leq B$

Fast algorithm

Yes

Yes

Decision Problems

Known Undecidable problem: Post's Correspondence problem

Theorem 2.22 The decision problem of validity in predicate logic is undecidable: no program exists which, given any ϕ, decides whether $\vDash \phi$.

Post's Correspondence Problem $\leq$ validity in predicate logic

Is satisfiability checking decidable? No!

validity checking $\leq$ Unsatisfiability checking

ϕ is unsatisfiable if, and only if, $\neg\phi$ is valid

Soundness & completeness

$\vDash \phi$ iff $\vdash \phi$

Provability is undecidable

EXPRESSIVENESS OF PREDICATE LOGIC OR FIRST-ORDER LOGIC

Existential Second-Order Logic

Example 2.23 Given a set of states $A = \{s_0, s_1, s_2, s_3\}$, let $R^{\mathcal{M}}$ be the set $\{(s_0, s_1), (s_1, s_0), (s_1, s_1), (s_1, s_2), (s_2, s_0), (s_3, s_0), (s_3, s_2)\}$. We may depict this model as a directed graph in Figure 2.5, where an edge (a transition) leads from a node s to a node s' iff $(s, s') \in R^{\mathcal{M}}$. In that case, we often denote this as $s \to s'$.

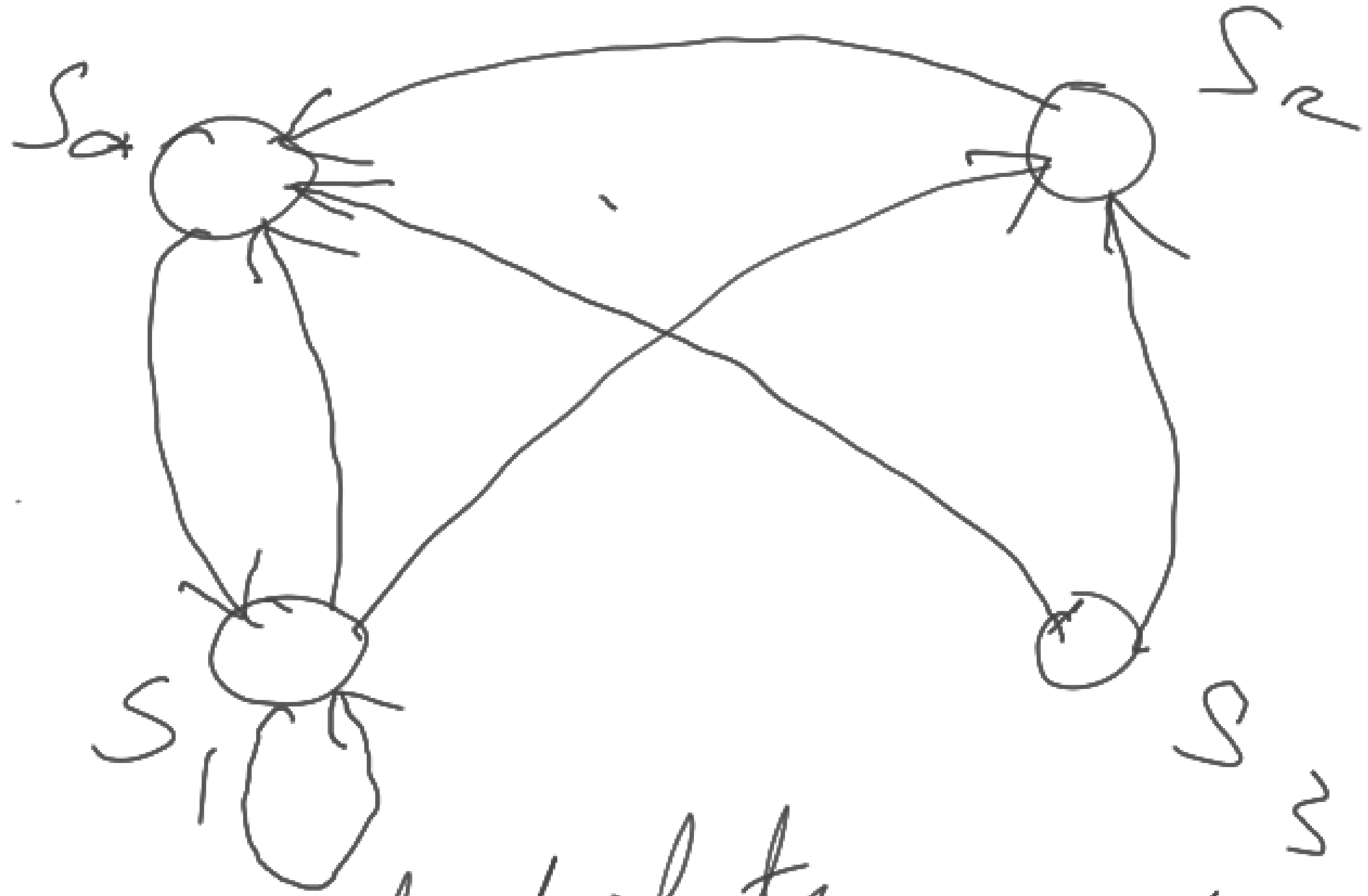
Directed Graph

$G(V, E)$

↑ vertex set

$E \subseteq V \times V$

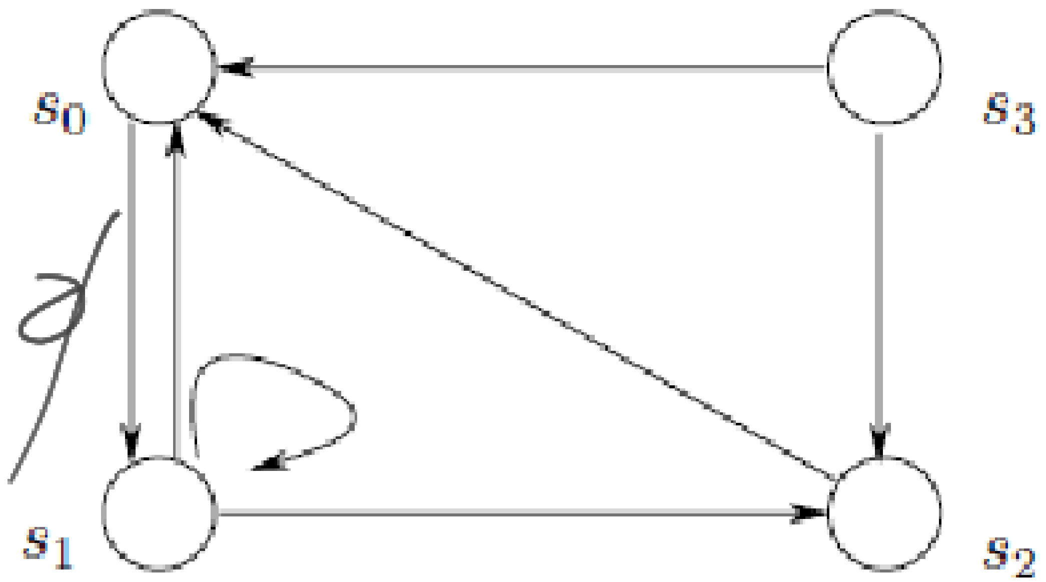
$E = \{(V_1, V_2), (V_1, V_3), (V_3, V_2)\}$



Reachability

e.g. S_2 is reachable from S_0

Is S_3 reachable from S_0? No!



Reachability: Given nodes n and n' in a directed graph, is there a finite path of transitions from n to n'?

Can we find a predicate logic formula ϕ

$$u = v \lor \exists x(R(u,x) \land R(x,v)) \lor \exists x_1 \exists x_2 (R(u,x_1) \land R(x_1,x_2) \land R(x_2,v)) \lor \ldots$$

with n & n' as its only free variables & R as its only predicate symbol, such that ϕ holds in directed graphs iff there is a path from n to n'?

Theorem 2.24 (Compactness Theorem) Let Γ be a set of sentences of predicate logic. If all finite subsets of Γ are satisfiable, then so is Γ.

PROOF: We use proof by contradiction: Assume that Γ is not satisfiable. Then the semantic entailment $\Gamma \vDash \bot$ holds as there is no model in which all $\phi \in \Gamma$ are true. By completeness, this means that the sequent $\Gamma \vdash \bot$ is valid. (Note that this uses a slightly more general notion of sequent in which we may have infinitely many premises at our disposal. Soundness and completeness remain true for that reading.) Thus, this sequent has a proof in natural deduction; this proof – being a finite piece of text – can use only finitely many premises Δ from Γ. But then $\Delta \vdash \bot$ is valid, too, and so $\Delta \vDash \bot$ follows by soundness. But the latter contradicts the fact that all finite subsets of Γ are consistent. $\square$

Theorem 2.25 (Löwenheim-Skolem Theorem) Let ψ be a sentence of predicate logic such for any natural number $n \geq 1$ there is a model of ψ with at least n elements. Then ψ has a model with infinitely many elements.

PROOF: The formula $\phi_n \stackrel{\text{def}}{=} \exists x_1 \exists x_2 \ldots \exists x_n \bigwedge_{1 \leq i < j \leq n} \neg(x_i = x_j)$ specifies that there are at least n elements. Consider the set of sentences $\Gamma \stackrel{\text{def}}{=} \{\psi\} \cup \{\phi_n \mid n \geq 1\}$ and let Δ be any if its finite subsets. Let $k \geq 1$ be such that $n \leq k$ for all n with $\phi_n \in \Delta$. Since the latter set is finite, such a k has to exist. By assumption, $\{\psi, \phi_k\}$ is satisfiable; but $\phi_k \to \phi_n$ is valid for all $n \leq k$ (why?). Therefore, Δ is satisfiable as well. The compactness theorem then implies that Γ is satisfiable by some model $\mathcal{M}$; in particular, $\mathcal{M} \vDash \psi$ holds. Since $\mathcal{M}$ satisfies ϕ_n for all $n \geq 1$, it cannot have finitely many elements. $\square$