3.3 Models, Validity, and Satisfiability

ϕ is valid in $\mathcal{A}$ under assignment β:

$$\mathcal{A}, \beta \models \phi \quad :\Leftrightarrow \quad \mathcal{A}(\beta)(\phi) = 1$$

ϕ is valid in $\mathcal{A}$ ($\mathcal{A}$ is a model of ϕ):

$$\mathcal{A} \models \phi \quad :\Leftrightarrow \quad \mathcal{A}, \beta \models \phi, \text{ for all } \beta \in X \to U_{\mathcal{A}}$$

ϕ is valid (or is a tautology):

$$\models \phi \quad :\Leftrightarrow \quad \mathcal{A} \models \phi, \text{ for all } \mathcal{A} \in \Sigma\text{-Alg}$$

ϕ is called satisfiable iff there exist $\mathcal{A}$ and β such that $\mathcal{A}, \beta \models \phi$. Otherwise ϕ is called unsatisfiable.

Substitution Lemma

The following propositions, to be proved by structural induction, hold for all Σ-algebras $\mathcal{A}$, assignments β, and substitutions σ.

Lemma 3.3:
For any Σ-term t

$$\mathcal{A}(\beta)(t\sigma) = \mathcal{A}(\beta \circ \sigma)(t),$$

where $\beta \circ \sigma : X \to \mathcal{A}$ is the assignment $\beta \circ \sigma(x) = \mathcal{A}(\beta)(x\sigma)$.

Proposition 3.4:
For any Σ-formula ϕ, $\mathcal{A}(\beta)(\phi\sigma) = \mathcal{A}(\beta \circ \sigma)(\phi)$.

Substitution Lemma

Corollary 3.5:

$\mathcal{A}, \beta \models \phi\sigma \quad \Leftrightarrow \quad \mathcal{A}, \beta \circ \sigma \models \phi$

These theorems basically express that the syntactic concept of substitution corresponds to the semantic concept of an assignment.

Entailment and Equivalence

ϕ entails (implies) ψ (or ψ is a consequence of ϕ), written $\phi \models \psi$, if for all $\mathcal{A} \in \Sigma$-Alg and $\beta \in X \to U_{\mathcal{A}}$, whenever $\mathcal{A}, \beta \models \phi$, then $\mathcal{A}, \beta \models \psi$.

ϕ and ψ are called equivalent, written $\phi \models\mid \psi$, if for all $\mathcal{A} \in \Sigma$-Alg and $\beta \in X \to U_{\mathcal{A}}$ we have $\mathcal{A}, \beta \models \phi \quad \Leftrightarrow \quad \mathcal{A}, \beta \models \psi$.

Entailment and Equivalence

Proposition 3.6:
ϕ entails ψ iff $(\phi \rightarrow \psi)$ is valid

Proposition 3.7:
ϕ and ψ are equivalent iff $(\phi \leftrightarrow \psi)$ is valid.

Extension to sets of formulas N in the "natural way", e. g.,
$N \models \phi$

$:\Leftrightarrow$ for all $\mathcal{A} \in \Sigma$-Alg and $\beta \in X \rightarrow U_{\mathcal{A}}$: if $\mathcal{A}, \beta \models \psi$, for all $\psi \in N$, then $\mathcal{A}, \beta \models \phi$.

Validity vs. Unsatisfiability

Validity and unsatisfiability are just two sides of the same medal as explained by the following proposition.

Proposition 3.8:
Let ϕ and ψ be formulas, let N be a set of formulas. Then

(i) ϕ is valid if and only if $\neg\phi$ is unsatisfiable.

(ii) $\phi \models \psi$ if and only if $\phi \wedge \neg\psi$ is unsatisfiable.

(iii) $N \models \psi$ if and only if $N \cup \{\neg\psi\}$ is unsatisfiable.

Hence in order to design a theorem prover (validity checker) it is sufficient to design a checker for unsatisfiability.

Theory of a Structure

Let $\mathcal{A} \in \Sigma$-Alg. The (first-order) theory of $\mathcal{A}$ is defined as

$$Th(\mathcal{A}) = \{ \psi \in \mathsf{F}_\Sigma(X) \mid \mathcal{A} \models \psi \}$$

Problem of axiomatizability:

For which structures $\mathcal{A}$ can one axiomatize $Th(\mathcal{A})$, that is, can one write down a formula ϕ (or a recursively enumerable set ϕ of formulas) such that

$$Th(\mathcal{A}) = \{ \psi \mid \phi \models \psi \}?$$

Analogously for sets of structures.

Two Interesting Theories

Let $\Sigma_{Pres} = (\{0/0, s/1, +/2\},\ \emptyset)$ and $\mathbb{Z}_+ = (\mathbb{Z}, 0, s, +)$ its standard interpretation on the integers. $Th(\mathbb{Z}_+)$ is called Presburger arithmetic (M. Presburger, 1929). (There is no essential difference when one, instead of $\mathbb{Z}$, considers the natural numbers $\mathbb{N}$ as standard interpretation.)

Presburger arithmetic is decidable in 3EXPTIME (D. Oppen, JCSS, 16(3):323–332, 1978), and in 2EXPSPACE, using automata-theoretic methods (and there is a constant $c \geq 0$ such that $Th(\mathbb{Z}_+) \notin \text{NTIME}(2^{2^{cn}})$).

Two Interesting Theories

However, $\mathbb{N}_* = (\mathbb{N}, 0, s, +, *)$, the standard interpretation of $\Sigma_{PA} = (\{0/0, s/1, +/2, */2\}, \emptyset)$, has as theory the so-called Peano arithmetic which is undecidable, not even recursively enumerable.

Note: The choice of signature can make a big difference with regard to the computational complexity of theories.

3.4 Algorithmic Problems

Validity(ϕ): $\models \phi$?

Satisfiability(ϕ): ϕ satisfiable?

Entailment(ϕ,ψ): does ϕ entail ψ?

Model($\mathcal{A}$,ϕ): $\mathcal{A} \models \phi$?

Solve($\mathcal{A}$,ϕ): find an assignment β such that $\mathcal{A}, \beta \models \phi$.

Solve(ϕ): find a substitution σ such that $\models \phi\sigma$.

Abduce(ϕ): find ψ with "certain properties" such that $\psi \models \phi$.

Gödel's Famous Theorems

1. For most signatures Σ, validity is undecidable for Σ-formulas. (Later by Turing: Encode Turing machines as Σ-formulas.)
2. For each signature Σ, the set of valid Σ-formulas is recursively enumerable. (We will prove this by giving complete deduction systems.)
3. For $\Sigma = \Sigma_{PA}$ and $\mathbb{N}_* = (\mathbb{N}, 0, s, +, *)$, the theory $Th(\mathbb{N}_*)$ is not recursively enumerable.

These complexity results motivate the study of subclasses of formulas (fragments) of first-order logic

Q: Can you think of any fragments of first-order logic for which validity is decidable?

Some Decidable Fragments

Some decidable fragments:

- Monadic class: no function symbols, all predicates unary; validity is NEXPTIME-complete.
- Variable-free formulas without equality: satisfiability is NP-complete. (why?)
- Variable-free Horn clauses (clauses with at most one positive atom): entailment is decidable in linear time.
- Finite model checking is decidable in time polynomial in the size of the structure and the formula.

Plan

Lift superposition from propositional logic to first-order logic.

3.5 Normal Forms and Skolemization

Study of normal forms motivated by

- reduction of logical concepts,
- efficient data structures for theorem proving,
- satisfiability preserving transformations (renaming),
- Skolem's and Herbrand's theorem.

The main problem in first-order logic is the treatment of quantifiers. The subsequent normal form transformations are intended to eliminate many of them.

Prenex Normal Form (Traditional)

Prenex formulas have the form

$$Q_1 x_1 \dots Q_n x_n \ \phi,$$

where ϕ is quantifier-free and $Q_i \in \{\forall, \exists\}$; we call $Q_1 x_1 \dots Q_n x_n$ the quantifier prefix and ϕ the matrix of the formula.

Prenex Normal Form (Traditional)

Computing prenex normal form by the rewrite system $\Rightarrow_P$:

$$
\begin{array}{rcll}
(\phi \leftrightarrow \psi) & \Rightarrow_P & (\phi \rightarrow \psi) \land (\psi \rightarrow \phi) & \\
\neg Qx\phi & \Rightarrow_P & \overline{Q}x\neg\phi & (\neg Q) \\
((Qx\phi)\ \rho\ \psi) & \Rightarrow_P & Qy(\phi\{x \mapsto y\}\ \rho\ \psi),\ \ \rho \in \{\land, \lor\} & \\
((Qx\phi) \rightarrow \psi) & \Rightarrow_P & \overline{Q}y(\phi\{x \mapsto y\} \rightarrow \psi), & \\
(\phi\ \rho\ (Qx\psi)) & \Rightarrow_P & Qy(\phi\ \rho\ \psi\{x \mapsto y\}),\ \ \rho \in \{\land, \lor, \rightarrow\} &
\end{array}
$$

Here y is always assumed to be some fresh variable and $\overline{Q}$ denotes the quantifier dual to Q, i. e., $\overline{\forall} = \exists$ and $\overline{\exists} = \forall$.

Skolemization

Intuition: replacement of $\exists y$ by a concrete choice function computing y from all the arguments y depends on.

Transformation $\Rightarrow_S$ (to be applied outermost, *not* in subformulas):

$$\forall x_1, \ldots, x_n \exists y\, \phi \quad \Rightarrow_S \quad \forall x_1, \ldots, x_n\, \phi\{y \mapsto f(x_1, \ldots, x_n)\}$$

where f/n is a new function symbol (Skolem function).

Skolemization

Together: $\phi \Rightarrow_P^* \underbrace{\psi}_{\text{prenex}} \Rightarrow_S^* \underbrace{\chi}_{\text{prenex, no } \exists}$

Theorem 3.9:

Let ϕ, ψ, and χ as defined above and closed. Then

(i) ϕ and ψ are equivalent.

(ii) $\chi \models \psi$ but the converse is not true in general.

(iii) ψ satisfiable (Σ-Alg) $\Leftrightarrow$ χ satisfiable (Σ'-Alg) where $\Sigma' = (\Omega \cup \mathit{SKF}, \Pi)$, if $\Sigma = (\Omega, \Pi)$.

The Complete Picture

$$\phi \quad \Rightarrow^*_P \quad Q_1 y_1 \ldots Q_n y_n \, \psi \qquad (\psi \text{ quantifier-free})$$

$$\Rightarrow^*_S \quad \forall x_1, \ldots, x_m \, \chi \qquad (m \leq n, \ \chi \text{ quantifier-free})$$

$$\Rightarrow^*_{OCNF} \quad \underbrace{\underbrace{\forall x_1, \ldots, x_m}_{\text{leave out}} \; \bigwedge_{i=1}^{k} \; \underbrace{\bigvee_{j=1}^{n_i} L_{ij}}_{\text{clauses } C_i}}_{\phi'}$$

$N = \{C_1, \ldots, C_k\}$ is called the clausal (normal) form (CNF) of ϕ.

Note: the variables in the clauses are implicitly universally quantified.

The Complete Picture

Theorem 3.10:
Let ϕ be closed. Then $\phi' \models \phi$. (The converse is not true in general.)

Theorem 3.11:
Let ϕ be closed. Then ϕ is satisfiable iff ϕ' is satisfiable iff N is satisfiable

Optimization

The normal form algorithm described so far leaves lots of room for optimization. Note that we only can preserve satisfiability anyway due to Skolemization.

- size of the CNF is exponential when done naively; the transformations we introduced already for propositional logic avoid this exponential growth;
- we want to preserve the original formula structure;
- we want small arity of Skolem functions (see next section).

3.6 Getting Small Skolem Functions

A clause set that is better suited for automated theorem proving can be obtained using the following steps:

- produce a negation normal form (NNF)
- apply miniscoping
- rename all variables
- skolemize

Negation Normal Form (NNF)

Apply the rewrite system $\Rightarrow_{\mathsf{NNF}}$:

$$\phi[\psi_1 \leftrightarrow \psi_2]_p \;\Rightarrow_{\mathsf{NNF}}\; \phi[(\psi_1 \rightarrow \psi_2) \wedge (\psi_2 \rightarrow \psi_1)]_p$$

if $\mathrm{pol}(\phi, p) = 1$ or $\mathrm{pol}(\phi, p) = 0$

$$\phi[\psi_1 \leftrightarrow \psi_2]_p \;\Rightarrow_{\mathsf{NNF}}\; \phi[(\psi_1 \wedge \psi_2) \vee (\neg\psi_2 \wedge \neg\psi_1)]_p$$

if $\mathrm{pol}(\phi, p) = -1$

Negation Normal Form (NNF)

$$
\begin{aligned}
\neg Qx\,\phi &\quad \Rightarrow_{\mathsf{NNF}} \quad \overline{Q}x\,\neg\phi \\
\neg(\phi \vee \psi) &\quad \Rightarrow_{\mathsf{NNF}} \quad \neg\phi \wedge \neg\psi \\
\neg(\phi \wedge \psi) &\quad \Rightarrow_{\mathsf{NNF}} \quad \neg\phi \vee \neg\psi \\
\phi \rightarrow \psi &\quad \Rightarrow_{\mathsf{NNF}} \quad \neg\phi \vee \psi \\
\neg\neg\phi &\quad \Rightarrow_{\mathsf{NNF}} \quad \phi
\end{aligned}
$$

Miniscoping

Apply the rewrite relation $\Rightarrow_{\mathsf{MS}}$. For the rules below we assume that x occurs freely in ψ, χ, but x does not occur freely in ϕ:

$$\begin{aligned}
Qx\,(\psi \wedge \phi) &\quad \Rightarrow_{\mathsf{MS}} \quad (Qx\,\psi) \wedge \phi \\
Qx\,(\psi \vee \phi) &\quad \Rightarrow_{\mathsf{MS}} \quad (Qx\,\psi) \vee \phi \\
\forall x\,(\psi \wedge \chi) &\quad \Rightarrow_{\mathsf{MS}} \quad (\forall x\,\psi) \wedge (\forall x\,\chi) \\
\exists x\,(\psi \vee \chi) &\quad \Rightarrow_{\mathsf{MS}} \quad (\exists x\,\psi) \vee (\exists x\,\chi)
\end{aligned}$$

Variable Renaming

Rename all variables in ϕ such that there are no two different positions p, q with $\phi|_p = Qx\,\psi$ and $\phi|_q = Q'x\,\chi$.

Standard Skolemization

Apply the rewrite rule:

$$\phi[\exists x\,\psi]_p \quad \Rightarrow_{\mathsf{SK}} \quad \phi[\psi\{x \mapsto f(y_1, \ldots, y_n)\}]_p$$

where p has minimal length,

$\{y_1, \ldots, y_n\}$ are the free variables in $\exists x\,\psi$,

f/n is a new function symbol to ϕ