3.4 Algorithmic Problems

Validity(F): $\models F$?

Satisfiability(F): F satisfiable?

Entailment(F,G): does F entail G?

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

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

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

Abduce(F): find G with "certain properties" such that $G \models F$.

Theory of an Algebra

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

$$\mathrm{Th}(\mathcal{A}) = \{\, G \in \mathrm{F}_\Sigma(X) \mid \mathcal{A} \models G \,\}$$

Problem of axiomatizability:

Given an algebra $\mathcal{A}$ (or a class of algebras) can one *axiomatize* $\mathrm{Th}(\mathcal{A})$, that is, can one write down a formula F (or a recursively enumerable set F of formulas) such that

$$\mathrm{Th}(\mathcal{A}) = \{\, G \mid F \models G \,\}?$$

Two Interesting Theories

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

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

(Non-)Computability Results

1. For most signatures Σ, validity is undecidable for Σ-formulas.
   (One can easily encode Turing machines in most signatures.)
2. Gödel's completeness theorem:
   For each signature Σ, the set of valid Σ-formulas is recursively enumerable.
   (We will prove this by giving complete deduction systems.)
3. Gödel's incompleteness theorem:
   For $\Sigma = \Sigma_{\mathrm{PA}}$ and $\mathbb{N}_* = (\mathbb{N}, 0, s, +, *, <)$, the theory $\mathrm{Th}(\mathbb{N}_*)$ is not recursively enumerable.

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

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 exponential time and PSPACE-complete.

3.5 Normal Forms and Skolemization

Study of normal forms motivated by

- reduction of logical concepts,
- efficient data structures for theorem proving.

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

$$\mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n \; F,$$

where F is quantifier-free and $\mathsf{Q}_i \in \{\forall, \exists\}$; we call $\mathsf{Q}_1 x_1 \ldots \mathsf{Q}_n x_n$ the *quantifier prefix* and F the *matrix* of the formula.

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

$$\begin{array}{rcl}
H[(F \leftrightarrow G)]_p & \Rightarrow_P & H[(F \to G) \wedge (G \to F)]_p \\
H[\neg \mathsf{Q} x\, F]_p & \Rightarrow_P & H[\overline{\mathsf{Q}} x\, \neg F]_p \\
H[((\mathsf{Q} x\, F) \circ G)]_p & \Rightarrow_P & H[\mathsf{Q} y\, (F\{x \mapsto y\} \circ G)]_p, \\
 & & \quad \circ \in \{\wedge, \vee\} \\
H[((\mathsf{Q} x\, F) \to G)]_p & \Rightarrow_P & H[\overline{\mathsf{Q}} y\, (F\{x \mapsto y\} \to G)]_p, \\
H[(F \circ (\mathsf{Q} x\, G))]_p & \Rightarrow_P & H[\mathsf{Q} y\, (F \circ G\{x \mapsto y\})]_p, \\
 & & \quad \circ \in \{\wedge, \vee, \to\}
\end{array}$$

Here y is always assumed to be some fresh variable and $\overline{\mathsf{Q}}$ 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):

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

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

Together: $F \Rightarrow_P^* \underbrace{G}_{\text{prenex}} \Rightarrow_S^* \underbrace{H}_{\text{prenex, no } \exists}$

Theorem 3.9 *Let F, G, and H as defined above and closed. Then*

(i) F and G are equivalent.

(ii) $H \models G$ but the converse is not true in general.

(iii) G satisfiable (w. r. t. Σ-Alg) $\Leftrightarrow$ H satisfiable (w. r. t. Σ'-Alg) where $\Sigma' = (\Omega \cup SKF, \Pi)$ if $\Sigma = (\Omega, \Pi)$.

The Complete Picture

$$
\begin{array}{lll}
F & \Rightarrow_P^* & \mathsf{Q}_1 y_1 \ldots \mathsf{Q}_n y_n \, G \qquad\qquad (G \text{ quantifier-free}) \\
 & \Rightarrow_S^* & \forall x_1, \ldots, x_m \, H \qquad (m \leq n,\ H \text{ quantifier-free}) \\
 & \Rightarrow_{CNF}^* & \underbrace{\underbrace{\forall x_1, \ldots, x_m}_{\text{leave out}} \bigwedge_{i=1}^{k} \underbrace{\bigvee_{j=1}^{n_i} L_{ij}}_{\text{clauses } C_i}}_{F'}
\end{array}
$$

$N = \{C_1, \ldots, C_k\}$ is called the *clausal (normal) form* (CNF) of F.
Note: The variables in the clauses are implicitly universally quantified.

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

Theorem 3.11 *Let F be closed. Then F is satisfiable iff F' 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.

- the 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 Skolem Functions with Small Arity

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

- eliminate trivial subformulas
- replace beneficial subformulas
- produce a negation normal form (NNF)
- apply miniscoping
- rename all variables
- Skolemize
- push quantifiers upward
- apply distributivity

We start with a closed formula.

Elimination of Trivial Subformulas

Eliminate subformulas $\top$ and $\bot$ essentially as in the propositional case modulo associativity/commutativity of $\wedge$, $\vee$:

$$
\begin{aligned}
H[(F \wedge \top)]_p &\Rightarrow_{\mathrm{OCNF}} H[F]_p \\
H[(F \vee \bot)]_p &\Rightarrow_{\mathrm{OCNF}} H[F]_p \\
H[(F \leftrightarrow \bot)]_p &\Rightarrow_{\mathrm{OCNF}} H[\neg F]_p \\
H[(F \leftrightarrow \top)]_p &\Rightarrow_{\mathrm{OCNF}} H[F]_p \\
H[(F \vee \top)]_p &\Rightarrow_{\mathrm{OCNF}} H[\top]_p \\
H[(F \wedge \bot)]_p &\Rightarrow_{\mathrm{OCNF}} H[\bot]_p \\
H[\neg\top]_p &\Rightarrow_{\mathrm{OCNF}} H[\bot]_p \\
H[\neg\bot]_p &\Rightarrow_{\mathrm{OCNF}} H[\top]_p \\
H[(F \rightarrow \bot)]_p &\Rightarrow_{\mathrm{OCNF}} H[\neg F]_p \\
H[(F \rightarrow \top)]_p &\Rightarrow_{\mathrm{OCNF}} H[\top]_p \\
H[(\bot \rightarrow F)]_p &\Rightarrow_{\mathrm{OCNF}} H[\top]_p \\
H[(\top \rightarrow F)]_p &\Rightarrow_{\mathrm{OCNF}} H[F]_p \\
H[\mathsf{Q}x\,\top]_p &\Rightarrow_{\mathrm{OCNF}} H[\top]_p \\
H[\mathsf{Q}x\,\bot]_p &\Rightarrow_{\mathrm{OCNF}} H[\bot]_p
\end{aligned}
$$

Replacement of Beneficial Subformulas

The functions ν and $\bar{\nu}$ that give us an overapproximation for the number of clauses generated by a formula are extended to quantified formulas by

$$\nu(\forall x\, F) = \nu(\exists x\, F) = \nu(F),$$
$$\bar{\nu}(\forall x\, F) = \bar{\nu}(\exists x\, F) = \bar{\nu}(F).$$

The other cases are defined as for propositional formulas.

Introduce top-down fresh predicates for beneficial subformulas:

$$H[F]_p \;\Rightarrow_{\mathrm{OCNF}}\; H[P(x_1, \ldots, x_n)]_p \wedge \mathrm{def}(H, p, P, F)$$

if $\nu(H[F]_p) > \nu(H[P(\ldots)]_p \wedge \mathrm{def}(H, p, P, F))$,

where $\{x_1, \ldots, x_n\}$ are the free variables in F, P/n is a predicate new to $H[F]_p$, and $\mathrm{def}(H, p, P, F)$ is defined by

$$\begin{array}{l} \forall x_1, \ldots, x_n\ (P(x_1, \ldots, x_n) \to F), \text{ if } \mathrm{pol}(H, p) = 1, \\ \forall x_1, \ldots, x_n\ (F \to P(x_1, \ldots, x_n)), \text{ if } \mathrm{pol}(H, p) = -1, \\ \forall x_1, \ldots, x_n\ (P(x_1, \ldots, x_n) \leftrightarrow F), \text{ if } \mathrm{pol}(H, p) = 0. \end{array}$$

As in the propositional case, one can test $\nu(H[F]_p) > \nu(H[P]_p \wedge \mathrm{def}(H, p, P, F))$ in constant time without actually computing ν.

Negation Normal Form (NNF)

Apply the reduction system $\Rightarrow_{\mathrm{NNF}}$:

$$H[F \leftrightarrow G]_p \;\Rightarrow_{\mathrm{NNF}}\; H[(F \to G) \wedge (G \to F)]_p$$

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

$$H[F \leftrightarrow G]_p \;\Rightarrow_{\mathrm{NNF}}\; H[(F \wedge G) \vee (\neg G \wedge \neg F)]_p$$

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

$$\begin{array}{rcl} H[F \to G]_p & \Rightarrow_{\mathrm{NNF}} & H[\neg F \vee G]_p \\ H[\neg\neg F]_p & \Rightarrow_{\mathrm{NNF}} & H[F]_p \\ H[\neg(F \vee G)]_p & \Rightarrow_{\mathrm{NNF}} & H[\neg F \wedge \neg G]_p \\ H[\neg(F \wedge G)]_p & \Rightarrow_{\mathrm{NNF}} & H[\neg F \vee \neg G]_p \\ H[\neg \mathsf{Q} x\ F]_p & \Rightarrow_{\mathrm{NNF}} & H[\overline{\mathsf{Q}} x\ \neg F]_p \end{array}$$

Miniscoping

Apply the reduction system $\Rightarrow_{\mathrm{MS}}$ modulo associativity and commutativity of $\wedge$, $\vee$. For the rules below we assume that x occurs freely in F, F', but x does not occur freely in G:

$$\begin{array}{lll}
H[\mathsf{Q}x\,(F \wedge G)]_p & \Rightarrow_{\mathrm{MS}} & H[(\mathsf{Q}x\,F) \wedge G]_p \\
H[\mathsf{Q}x\,(F \vee G)]_p & \Rightarrow_{\mathrm{MS}} & H[(\mathsf{Q}x\,F) \vee G]_p \\
H[\forall x\,(F \wedge F')]_p & \Rightarrow_{\mathrm{MS}} & H[(\forall x\,F) \wedge (\forall x\,F')]_p \\
H[\exists x\,(F \vee F')]_p & \Rightarrow_{\mathrm{MS}} & H[(\exists x\,F) \vee (\exists x\,F')]_p \\
H[\mathsf{Q}x\,G]_p & \Rightarrow_{\mathrm{MS}} & H[G]_p
\end{array}$$

Variable Renaming

Rename all variables in H such that there are no two different positions p, q with $H|_p = \mathsf{Q}x\,F$ and $H|_q = \mathsf{Q}'x\,G$.

Standard Skolemization

Apply the reduction system:

$$H[\exists x\,F]_p \Rightarrow_{\mathrm{SK}} H[F\{x \mapsto f(y_1, \ldots, y_n)\}]_p$$

where p has minimal length,
$\{y_1, \ldots, y_n\}$ are the free variables in $\exists x\,F$,
and f/n is a new function symbol to H.

Final Steps

Apply the reduction system modulo commutativity of $\wedge$, $\vee$ to push $\forall$ upward:

$$\begin{array}{lll}
H[(\forall x\,F) \wedge G]_p & \Rightarrow_{\mathrm{OCNF}} & H[\forall x\,(F \wedge G)]_p \\
H[(\forall x\,F) \vee G]_p & \Rightarrow_{\mathrm{OCNF}} & H[\forall x\,(F \vee G)]_p
\end{array}$$

Note that variable renaming ensures that x does not occur in G.

Apply the reduction system modulo commutativity of $\wedge$, $\vee$ to push disjunctions downward:

$$H[(F \wedge F') \vee G]_p \Rightarrow_{\mathrm{CNF}} H[(F \vee G) \wedge (F' \vee G)]_p$$

3.7 Herbrand Interpretations

From now on we shall consider FOL without equality. We assume that Ω contains at least one constant symbol.

A *Herbrand interpretation* (over Σ) is a Σ-algebra $\mathcal{A}$ such that

- $U_{\mathcal{A}} = \mathrm{T}_\Sigma$ (= the set of ground terms over Σ)
- $f_{\mathcal{A}} : (s_1, \ldots, s_n) \mapsto f(s_1, \ldots, s_n),\ f/n \in \Omega$

In other words, *values are fixed* to be ground terms and *functions are fixed* to be the *term constructors*. Only predicate symbols $P/m \in \Pi$ may be freely interpreted as relations $P_{\mathcal{A}} \subseteq \mathrm{T}_\Sigma^m$.

Proposition 3.12 *Every set of ground atoms I uniquely determines a Herbrand interpretation $\mathcal{A}$ via*

$$(s_1, \ldots, s_n) \in P_{\mathcal{A}} \ \text{ iff } \ P(s_1, \ldots, s_n) \in I$$

Thus we shall identify Herbrand interpretations (over Σ) with sets of Σ-ground atoms.

Existence of Herbrand Models

A Herbrand interpretation I is called a *Herbrand model* of F, if $I \models F$.

Theorem 3.13 (Herbrand) *Let N be a set of (universally quantified) Σ-clauses.*

$$\begin{array}{rcl} N \text{ satisfiable} & \Leftrightarrow & N \text{ has a Herbrand model (over } \Sigma) \\ & \Leftrightarrow & G_\Sigma(N) \text{ has a Herbrand model (over } \Sigma) \end{array}$$

where $G_\Sigma(N) = \{\, C\sigma \text{ ground clause} \mid (\forall \vec{x}\, C) \in N,\ \sigma : X \to \mathrm{T}_\Sigma \,\}$ is the set of ground instances *of N.*

[The proof will be given below in the context of the completeness proof for general resolution.]

3.8 Inference Systems and Proofs

Inference systems Γ (proof calculi) are sets of tuples

$$(F_1, \ldots, F_n, F_{n+1}),\ n \geq 0,$$

called *inferences*, and written

$$\frac{\overbrace{F_1 \ \ldots \ F_n}^{\text{premises}}}{\underbrace{F_{n+1}}_{\text{conclusion}}}\ .$$

Clausal inference system: premises and conclusions are clauses. One also considers inference systems over other data structures.

Inference Systems

Inference systems Γ are shorthands for reduction systems over sets of formulas. If N is a set of formulas, then

$$\frac{\overbrace{F_1 \ \ldots \ F_n}^{\text{premises}}}{\underbrace{F_{n+1}}_{\text{conclusion}}}\quad \textit{side condition}$$

is a shorthand for

$$N \cup \{F_1, \ldots, F_n\} \quad \Rightarrow_\Gamma \quad N \cup \{F_1, \ldots, F_n\} \cup \{F_{n+1}\} \quad \text{if } \textit{side condition}$$

Proofs

A *proof* in Γ of a formula F from a set of formulas N (called *assumptions*) is a sequence $F_1, \ldots, F_k$ of formulas where

(i) $F_k = F$,

(ii) for all $1 \leq i \leq k$: $F_i \in N$ or there exists an inference

$$\frac{F_{m_1} \ \ldots \ F_{m_n}}{F_i}$$

in Γ, such that $0 \leq m_j < i$, for $1 \leq j \leq n$.

Soundness and Completeness

Provability $\vdash_\Gamma$ of F from N in Γ:

$N \vdash_\Gamma F$ if there exists a proof in Γ of F from N.

Γ is called *sound*, if

$$\frac{F_1 \ \ldots \ F_n}{F} \in \Gamma \text{ implies } F_1, \ldots, F_n \models F$$

Γ is called *complete*, if

$N \models F$ implies $N \vdash_\Gamma F$

Γ is called *refutationally complete*, if

$N \models \bot$ implies $N \vdash_\Gamma \bot$

Proposition 3.14

(i) Let Γ be sound. Then $N \vdash_\Gamma F \Rightarrow N \models F$

(ii) If $N \vdash_\Gamma F$ then there exist finitely many $F_1, \ldots, F_n \in N$ such that $F_1, \ldots, F_n \vdash_\Gamma F$

Reduced Proofs

The definition of a proof of F given above admits sequences $F_1, \ldots, F_k$ of formulas where some F_i are not ancestors of $F_k = F$ (i. e., some F_i are not actually used to derive F).

A proof is called *reduced*, if every F_i with $i < k$ is an ancestor of F_k.

We obtain a reduced proof from a proof by marking first F_k and then recursively all the premises used to derive a marked conclusion, and by deleting all non-marked formulas in the end.

Reduced Proofs as Trees

markings	$\widehat{=}$	formulas			
leaves	$\widehat{=}$	assumptions and axioms			
other nodes	$\widehat{=}$	inferences:	conclusion	$\widehat{=}$	parent node
			premises	$\widehat{=}$	child nodes

$$\dfrac{\dfrac{P(f(c)) \lor Q(b) \quad \neg P(f(c)) \lor \neg P(f(c)) \lor Q(b)}{\neg P(f(c)) \lor Q(b) \lor Q(b)}}{\dfrac{P(f(c)) \lor Q(b) \qquad \neg P(f(c)) \lor Q(b)}{Q(b) \lor Q(b)}} \qquad \neg P(f(c)) \lor \neg Q(b)$$
$$\dfrac{\dfrac{Q(b) \lor Q(b)}{Q(b)} \qquad \neg P(f(c)) \lor \neg Q(b)}{P(f(c)) \qquad \neg P(f(c))}$$
$$\bot$$

3.9 Ground (or propositional) Resolution

We observe that propositional clauses and ground clauses are essentially the same, as long as we do not consider equational atoms.

In this section we only deal with ground clauses.

Unlike in Section 2 we admit duplicated literals in clauses, i. e., we treat clauses like multisets of literals, not like sets.

The Resolution Calculus *Res*

Resolution inference rule:

$$\dfrac{D \lor A \qquad C \lor \neg A}{D \lor C}$$

Terminology: $D \lor C$: *resolvent*; A: *resolved atom*

(Positive) factorization inference rule:

$$\dfrac{C \lor A \lor A}{C \lor A}$$

These are *schematic inference rules*; for each substitution of the *schematic variables* C, D, and A, by ground clauses and ground atoms, respectively, we obtain an inference.

We treat "$\vee$" as associative and commutative, hence A and $\neg A$ can occur anywhere in the clauses; moreover, when we write $C \vee A$, etc., this includes unit clauses, that is, $C = \bot$.

Sample Refutation

1.	$\neg P(f(c)) \vee \neg P(f(c)) \vee Q(b)$	(given)
2.	$P(f(c)) \vee Q(b)$	(given)
3.	$\neg P(g(b,c)) \vee \neg Q(b)$	(given)
4.	$P(g(b,c))$	(given)
5.	$\neg P(f(c)) \vee Q(b) \vee Q(b)$	(Res. 2. into 1.)
6.	$\neg P(f(c)) \vee Q(b)$	(Fact. 5.)
7.	$Q(b) \vee Q(b)$	(Res. 2. into 6.)
8.	$Q(b)$	(Fact. 7.)
9.	$\neg P(g(b,c))$	(Res. 8. into 3.)
10.	$\bot$	(Res. 4. into 9.)

Soundness of Resolution

Theorem 3.15 *Propositional resolution is sound.*

Proof. Let $\mathcal{B} \in \Sigma$-Alg. We have to show:

(i) for resolution: $\mathcal{B} \models D \vee A,\ \mathcal{B} \models C \vee \neg A \ \Rightarrow\ \mathcal{B} \models D \vee C$

(ii) for factorization: $\mathcal{B} \models C \vee A \vee A \ \Rightarrow\ \mathcal{B} \models C \vee A$

(i): Assume premises are valid in $\mathcal{B}$. Two cases need to be considered:
If $\mathcal{B} \models A$, then $\mathcal{B} \models C$, hence $\mathcal{B} \models D \vee C$.
Otherwise, $\mathcal{B} \models \neg A$, then $\mathcal{B} \models D$, and again $\mathcal{B} \models D \vee C$.
(ii): Obvious. □

Note: In ground first-order logic we have (like in propositional logic):

1. $\mathcal{B} \models L_1 \vee \ldots \vee L_n$ if and only if there exists i: $\mathcal{B} \models L_i$.
2. $\mathcal{B} \models A$ or $\mathcal{B} \models \neg A$.

This does not hold for formulas with variables!