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 3.10 *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 3.11 (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 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 3.12

(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

$$\begin{array}{rcl} \text{markings} & \widehat{=} & \text{formulas} \\ \text{leaves} & \widehat{=} & \text{assumptions and axioms} \\ \text{other nodes} & \widehat{=} & \text{inferences: conclusion } \widehat{=} \text{ parent node} \\ & & \text{premises } \widehat{=} \text{ child nodes} \end{array}$$

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

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.

Note that 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:

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

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

(Positive) factorization inference rule:

$$\dfrac{C \vee A \vee A}{C \vee A}$$

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 3.13 *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!

3.10 Refutational Completeness of Resolution

How to show refutational completeness of ground resolution:

- We have to show: $N \models \bot \;\Rightarrow\; N \vdash_{Res} \bot$, or equivalently: If $N \not\vdash_{Res} \bot$, then N has a model.
- Idea: Suppose that we have computed sufficiently many inferences (and not derived $\bot$).
- Now order the clauses in N according to some appropriate ordering, inspect the clauses in ascending order, and construct a series of Herbrand interpretations.
- The limit interpretation can be shown to be a model of N.

Clause Orderings

1. We assume that $\succ$ is any fixed ordering on ground atoms that is *total* and *well-founded*. (There exist many such orderings, e. g., the lenght-based ordering on atoms when these are viewed as words over a suitable alphabet.)
2. Extend $\succ$ to an *ordering $\succ_L$ on ground literals:*

$$\begin{array}{lll} [\neg]A & \succ_L & [\neg]B \quad \text{, if } A \succ B \\ \neg A & \succ_L & A \end{array}$$

3. Extend $\succ_L$ to an *ordering $\succ_C$ on ground clauses:*
   $\succ_C = (\succ_L)_{\text{mul}}$, the multiset extension of $\succ_L$.

   Notation: $\succ$ also for $\succ_L$ and $\succ_C$.

Example

Suppose $A_5 \succ A_4 \succ A_3 \succ A_2 \succ A_1 \succ A_0$. Then:

$$\begin{array}{ll} & A_1 \vee \neg A_5 \\ \succ & A_3 \vee \neg A_4 \\ \succ & \neg A_1 \vee A_3 \vee A_4 \\ \succ & A_1 \vee \neg A_2 \\ \succ & \neg A_1 \vee A_2 \\ \succ & A_1 \vee A_1 \vee A_2 \\ \succ & A_0 \vee A_1 \end{array}$$