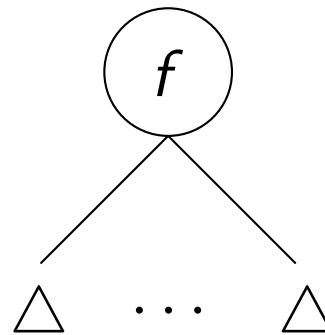# 3.7 Herbrand Interpretations

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

- $U_{\mathcal{A}} = T_{\Sigma}$ (= the set of ground terms over Σ)

- $f_{\mathcal{A}} : (s_1, \ldots, s_n) \mapsto f(s_1, \ldots, s_n), \ f/n \in \Omega$

$$f_{\mathcal{A}}(\triangle, \ldots, \triangle) =$$

Herbrand Interpretations

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 $P_{\mathcal{A}} \subseteq \mathsf{T}_{\Sigma}^m$.

Proposition 3.12:

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

$$(s_1, \ldots, s_n) \in P_{\mathcal{A}} \quad :\Leftrightarrow \quad P(s_1, \ldots, s_n) \in I$$

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

Herbrand Interpretations

Example: $\Sigma_{Pres} = (\{0/0, s/1, +/2\}, \{</2, \leq/2\})$

$\mathbb{N}$ as Herbrand interpretation over Σ_{Pres}:

$$I = \{ \quad 0 \leq 0, \ 0 \leq s(0), \ 0 \leq s(s(0)), \ \dots,$$

$$0 + 0 \leq 0, \ 0 + 0 \leq s(0), \ \dots,$$

$$\dots, \ (s(0) + 0) + s(0) \leq s(0) + (s(0) + s(0))$$

$$\dots$$

$$s(0) + 0 < s(0) + 0 + 0 + s(0)$$

$$\dots\}$$

Existence of Herbrand Models

A Herbrand interpretation I is called a Herbrand model of ϕ, if $I \models \phi$.

Theorem 3.13 (Herbrand):

Let N be a set of Σ-clauses.

$$N \text{ satisfiable} \quad \Leftrightarrow \quad N \text{ has a Herbrand model (over } \Sigma)$$

$$\Leftrightarrow \quad G_\Sigma(N) \text{ has a Herbrand model (over } \Sigma)$$

where $G_\Sigma(N) = \{ C\sigma \text{ ground clause} \mid C \in N, \ \sigma : X \to T_\Sigma \}$ is the set of ground instances of N.

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

Example of a G_Σ

For Σ_{Pres} one obtains for

$$C = (x < y) \vee (y \leq s(x))$$

the following ground instances:

$(0 < 0) \vee (0 \leq s(0))$
$(s(0) < 0) \vee (0 \leq s(s(0)))$
$\ldots$
$(s(0) + s(0) < s(0) + 0) \vee (s(0) + 0 \leq s(s(0) + s(0)))$
$\ldots$

3.8 Inference Systems and Proofs

Inference systems Γ (proof calculi) are sets of tuples

$$(\phi_1, \ldots, \phi_n, \phi_{n+1}), \quad n \geq 0,$$

called inferences, and written

$$\frac{\overbrace{\phi_1 \quad \ldots \quad \phi_n}^{\text{premises}}}{\underbrace{\phi_{n+1}}_{\text{conclusion}}} \ .$$

Clausal inference system: premises and conclusions are clauses.

One also considers inference systems over other data structures.

Inference Systems

Inference systems Γ are short hands for rewrite systems over sets of formulas. If N is a set of formulas, then

$$\overbrace{\phi_1 \ \ldots \ \phi_n}^{\text{premises}} \atop \underbrace{\phi_{n+1}}_{\text{conclusion}} \quad \textit{side condition}$$

is a shorthand for

$$N \cup \{\phi_1 \ \ldots \ \phi_n\} \quad \Rightarrow_\Gamma \quad N \cup \{\phi_1 \ \ldots \ \phi_n\} \cup \{\phi_{n+1}\}$$
$$\text{if } \textit{side condition}$$

Proofs

A proof in Γ of a formula ϕ from a a set of formulas N (called assumptions) is a sequence $\phi_1, \ldots, \phi_k$ of formulas where

(i) $\phi_k = \phi$,

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

$$\frac{\phi_{i_1} \quad \ldots \quad \phi_{i_{n_i}}}{\phi_i}$$

in Γ, such that $0 \leq i_j < i$, for $1 \leq j \leq n_i$.

Soundness and Completeness

Provability $\vdash_\Gamma$ of ϕ from N in Γ: $N \vdash_\Gamma \phi$ if there exists a proof Γ of ϕ from N.

Γ is called sound

$$\frac{\phi_1 \quad \ldots \quad \phi_n}{\phi} \in \Gamma \quad \text{implies } \phi_1, \ldots, \phi_n \models \phi$$

Γ is called complete

$$N \models \phi \quad \text{implies } N \vdash_\Gamma \phi$$

Γ is called refutationally complete

$$N \models \bot \quad \text{implies } N \vdash_\Gamma \bot$$

Soundness and Completeness

Proposition 3.14:

(i) Let Γ be sound. Then $N \vdash_\Gamma \phi$ implies $N \models \phi$

(ii) $N \vdash_\Gamma \phi$ implies there exist finitely many clauses $\phi_1, \ldots, \phi_n \in N$ such that $\phi_1, \ldots, \phi_n \vdash_\Gamma \phi$

Proofs as Trees

$$\begin{array}{rcl}
\text{markings} & \widehat{=} & \text{formulas} \\[1em]
\text{leaves} & \widehat{=} & \text{assumptions and axioms} \\[1em]
\text{other nodes} & \widehat{=} & \text{inferences:} \quad \text{conclusion} \quad \widehat{=} \quad \text{ancestor} \\[1em]
& & \text{premises} \quad \widehat{=} \quad \text{direct descendants}
\end{array}$$

$$
\cfrac{
    \cfrac{
        P(f(c)) \vee Q(b)
        \qquad
        \cfrac{
            \cfrac{
                P(f(c)) \vee Q(b) \qquad \neg P(f(c)) \vee \neg P(f(c)) \vee Q(b)
            }{
                \cfrac{\neg P(f(c)) \vee Q(b) \vee Q(b)}{\neg P(f(c)) \vee Q(b)}
            }
        }{}
    }{
        \cfrac{Q(b) \vee Q(b)}{Q(b)}
    }
    \qquad
    \cfrac{}{}
}{}
$$

3.9 Ground Superposition

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 and recall partly the material from Section 2.5 for first-order ground clauses.

The Resolution Calculus *Res*

Resolution inference rule:

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

Terminology: $D \lor C$: resolvent; A: resolved atom For Superposition (*Sup*): A strictly maximal, $\neg A$ maximal

(Positive) factorization inference rule:

$$\frac{C \lor A \lor A}{C \lor A}$$

For Superposition (*Sup*): A maximal

The Resolution Calculus *Res*

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

$$
\begin{array}{lll}
1. & \neg P(f(c)) \vee \neg P(f(c)) \vee Q(b) & \text{(given)} \\
2. & P(f(c)) \vee Q(b) & \text{(given)} \\
3. & \neg P(g(b,c)) \vee \neg Q(b) & \text{(given)} \\
4. & P(g(b,c)) & \text{(given)} \\
5. & \neg P(f(c)) \vee Q(b) \vee Q(b) & \text{(Res. 2. into 1.)} \\
6. & \neg P(f(c)) \vee Q(b) & \text{(Fact. 5.)} \\
7. & Q(b) \vee Q(b) & \text{(Res. 2. into 6.)} \\
8. & Q(b) & \text{(Fact. 7.)} \\
9. & \neg P(g(b,c)) & \text{(Res. 8. into 3.)} \\
10. & \bot & \text{(Res. 4. into 9.)}
\end{array}
$$

Soundness of Resolution

Theorem 3.15:

Propositional resolution is sound.

Proof:

Let $\mathcal{B} \in \Sigma\text{-Alg}$. To be shown:

(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): even simpler. $\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\square$

Soundness of Resolution

Note: In propositional logic (ground clauses) we have:

1. $\mathcal{B} \models L_1 \vee \ldots \vee L_n$ iff 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!

Closure of Clause Sets under *Res*

$$Res(N) = \{\, C \mid C \text{ is conclusion of an inference in } Res$$
$$\text{with premises in } N \,\}$$

$$Res^0(N) = N$$

$$Res^{n+1}(N) = Res(Res^n(N)) \cup Res^n(N), \text{ for } n \geq 0$$

$$Res^*(N) = \bigcup_{n \geq 0} Res^n(N)$$

N is called saturated (w. r. t. resolution), if $Res(N) \subseteq N$.

Closure of Clause Sets under *Res*

Proposition 3.16:

(i) $Res^*(N)$ is saturated.

(ii) Res is refutationally complete, iff for each set N of ground clauses:

$$N \models \perp \text{ iff } \perp \in Res^*(N)$$

Construction of Interpretations

Done the same way as for propositional logic: ground atoms play the rôle of propositional variables.

Model Existence Theorem

Theorem 3.17 (Bachmair & Ganzinger 1990):

Let $\succ$ be a clause ordering, let N be saturated w. r. t. Res (or Sup), and suppose that $\perp \notin N$. Then $N_{\mathcal{I}}^{\succ} \models N$.

Corollary 3.18:

Let N be saturated w. r. t. Res. Then $N \models \perp \Leftrightarrow \perp \in N$.

Compactness of Propositional Logic

Theorem 3.19 (Compactness):

Let N be a set of propositional (or first-order ground) formulas. Then N is unsatisfiable, if and only if some finite subset $M \subseteq N$ is unsatisfiable.

Proof:

"$\Leftarrow$": trivial. "$\Rightarrow$": Let N be unsatisfiable.

$\Rightarrow Res^*(N)$ unsatisfiable

$\Rightarrow \perp \in Res^*(N)$ by refutational completeness of resolution

$\Rightarrow \exists n \geq 0 : \perp \in Res^n(N)$

$\Rightarrow \perp$ has a finite resolution proof P;

choose M as the set of assumptions in P. □

3.10 General Resolution

Propositional (ground) resolution:
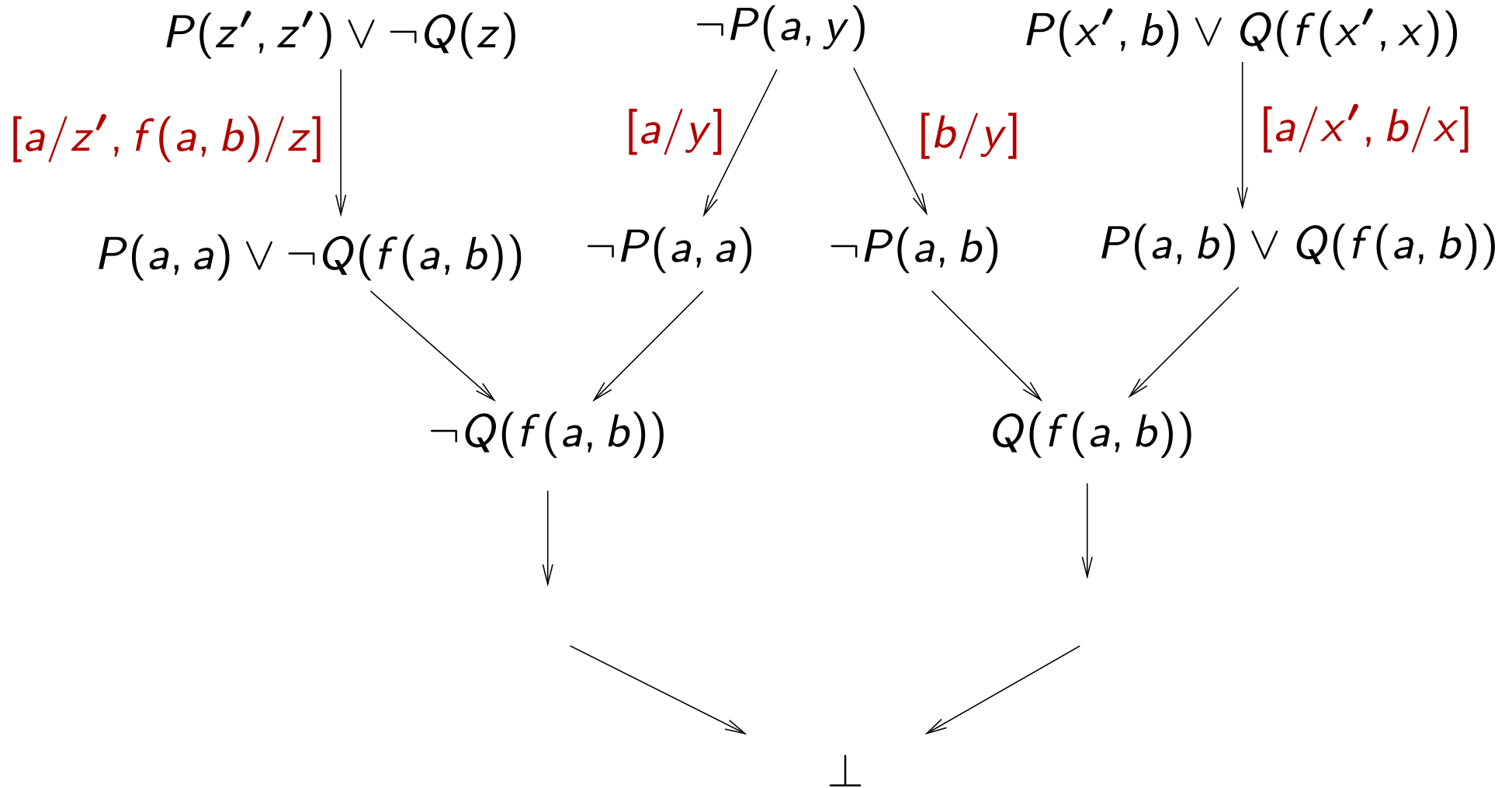
refutationally complete,

in its most naive version: not guaranteed to terminate for satisfiable sets of clauses, (improved versions do terminate, however)

inferior to the DPLL procedure.

But: in contrast to the DPLL procedure, resolution can be easily extended to non-ground clauses.

General Resolution through Instantiation

Idea: instantiate clauses appropriately:

$$P(z', z') \vee \neg Q(z) \qquad \neg P(a, y) \qquad P(x', b) \vee Q(f(x', x))$$

$[a/z', f(a, b)/z]$ $[a/y]$ $[b/y]$ $[a/x', b/x]$

$$P(a, a) \vee \neg Q(f(a, b)) \qquad \neg P(a, a) \qquad \neg P(a, b) \qquad P(a, b) \vee Q(f(a, b))$$

$$\neg Q(f(a, b)) \qquad\qquad\qquad Q(f(a, b))$$

$$\bot$$

General Resolution through Instantiation

Problems:

More than one instance of a clause can participate in a proof.

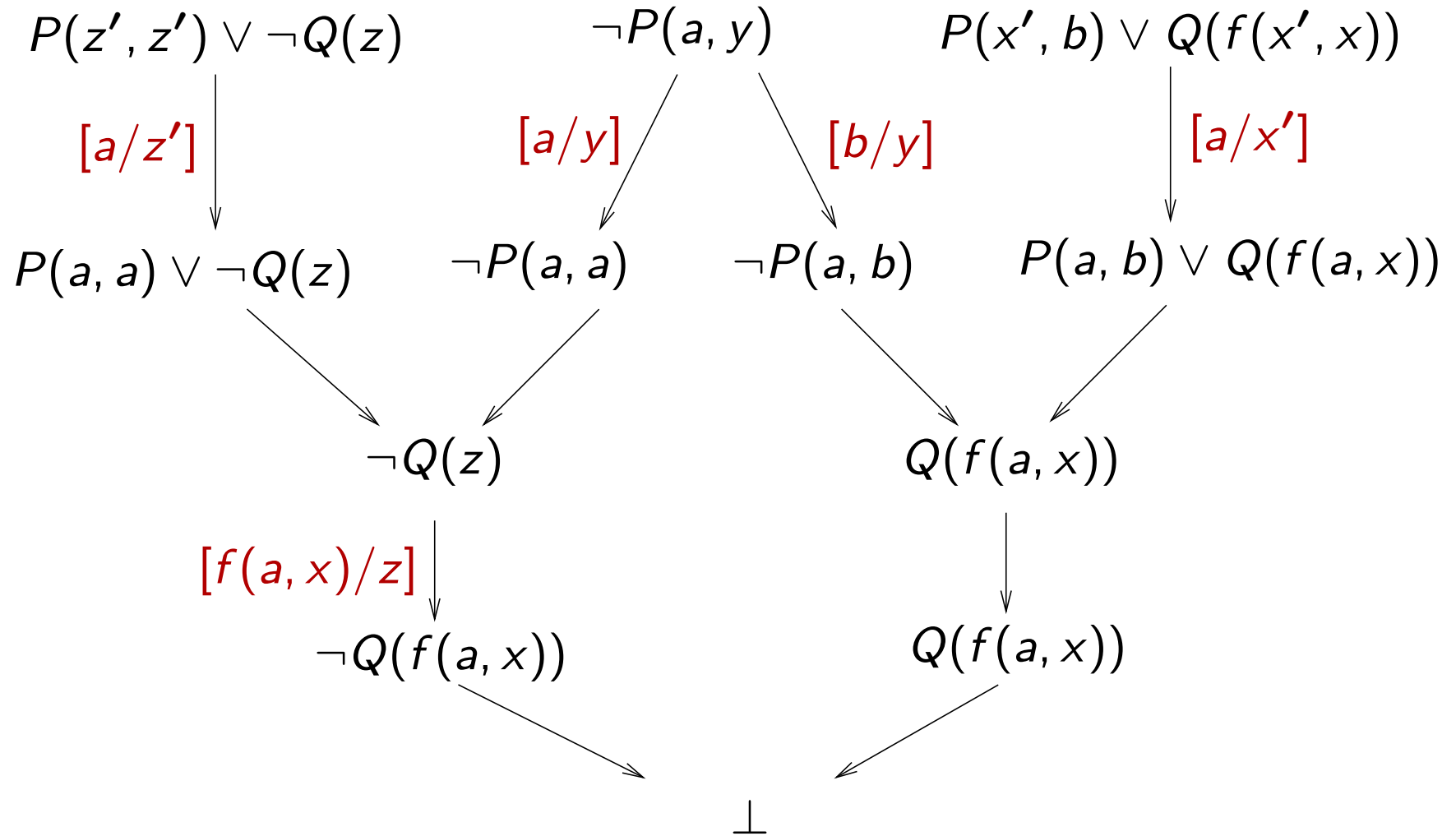Even worse: There are infinitely many possible instances.

Observation:

Instantiation must produce complementary literals (so that inferences become possible).

Idea:

Do not instantiate more than necessary to get complementary literals.

General Resolution through Instantiation

$$P(z', z') \vee \neg Q(z) \qquad \neg P(a, y) \qquad P(x', b) \vee Q(f(x', x))$$

$[a/z']$ $\qquad$ $[a/y]$ $\qquad$ $[b/y]$ $\qquad$ $[a/x']$

$$P(a, a) \vee \neg Q(z) \qquad \neg P(a, a) \qquad \neg P(a, b) \qquad P(a, b) \vee Q(f(a, x))$$

$$\neg Q(z) \qquad\qquad\qquad Q(f(a, x))$$

$[f(a, x)/z]$

$$\neg Q(f(a, x)) \qquad\qquad\qquad Q(f(a, x))$$

$$\bot$$

Lifting Principle

Problem: Make saturation of infinite sets of clauses as they arise from taking the (ground) instances of finitely many general clauses (with variables) effective and efficient.

Idea (Robinson 1965):

- Resolution for general clauses:

- *Equality* of ground atoms is generalized to *unifiability* of general atoms;

- Only compute *most general* (minimal) unifiers (mgu).

Lifting Principle

Significance: The advantage of the method in (Robinson 1965) compared with (Gilmore 1960) is that unification enumerates only those instances of clauses that participate in an inference. Moreover, clauses are not right away instantiated into ground clauses. Rather they are instantiated only as far as required for an inference. Inferences with non-ground clauses in general represent infinite sets of ground inferences which are computed simultaneously in a single step.

Resolution for General Clauses

General binary resolution *Res*:

$$\frac{D \vee B \qquad C \vee \neg A}{(D \vee C)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \qquad [\text{resolution}]$$

$$\frac{C \vee A \vee B}{(C \vee A)\sigma} \quad \text{if } \sigma = \text{mgu}(A, B) \quad [\text{factorization}]$$

Resolution for General Clauses

For inferences with more than one premise, we assume that the variables in the premises are (bijectively) renamed such that they become different to any variable in the other premises. We do not formalize this. Which names one uses for variables is otherwise irrelevant.

Unification

Let $E = \{s_1 \doteq t_1, \ldots, s_n \doteq t_n\}$ (s_i, t_i terms or atoms) a multiset of equality problems. A substitution σ is called a unifier of E if $s_i\sigma = t_i\sigma$ for all $1 \leq i \leq n$.

If a unifier of E exists, then E is called unifiable.

Unification

A substitution σ is called more general than a substitution τ, denoted by $\sigma \leq \tau$, if there exists a substitution ρ such that $\rho \circ \sigma = \tau$, where $(\rho \circ \sigma)(x) := (x\sigma)\rho$ is the composition of σ and ρ as mappings. (Note that $\rho \circ \sigma$ has a finite domain as required for a substitution.)

If a unifier of E is more general than any other unifier of E, then we speak of a most general unifier of E, denoted by $\mathrm{mgu}(E)$.

Unification

Proposition 3.20:

(i) $\leq$ is a quasi-ordering on substitutions, and $\circ$ is associative.

(ii) If $\sigma \leq \tau$ and $\tau \leq \sigma$ (we write $\sigma \sim \tau$ in this case), then $x\sigma$ and $x\tau$ are equal up to (bijective) variable renaming, for any x in X.

A substitution σ is called idempotent, if $\sigma \circ \sigma = \sigma$.

Proposition 3.21:

σ is idempotent iff $dom(\sigma) \cap codom(\sigma) = \emptyset$.

Rule-Based Naive Standard Unification

$$t \doteq t, E \quad \Rightarrow_{SU} \quad E$$

$$f(s_1, \ldots, s_n) \doteq f(t_1, \ldots, t_n), E \quad \Rightarrow_{SU} \quad s_1 \doteq t_1, \ldots, s_n \doteq t_n, E$$

$$f(\ldots) \doteq g(\ldots), E \quad \Rightarrow_{SU} \quad \bot$$

$$x \doteq t, E \quad \Rightarrow_{SU} \quad x \doteq t, E\{t \mapsto x\}$$
$$\text{if } x \in var(E), x \notin var(t)$$

$$x \doteq t, E \quad \Rightarrow_{SU} \quad \bot$$
$$\text{if } x \neq t, x \in var(t)$$

$$t \doteq x, E \quad \Rightarrow_{SU} \quad x \doteq t, E$$
$$\text{if } t \notin X$$

SU: Main Properties

If $E = x_1 \doteq u_1, \ldots, x_k \doteq u_k$, with x_i pairwise distinct, $x_i \notin var(u_j)$, then E is called an (equational problem in) solved form representing the solution $\sigma_E = \{x_1 \mapsto u_1, \ldots, x_k \mapsto u_k\}$.

Proposition 3.22:

If E is a solved form then σ_E is an mgu of E.

SU: Main Properties

Theorem 3.23:

  1. If $E \Rightarrow_{SU} E'$ then σ is a unifier of E iff σ is a unifier of E'

  2. If $E \Rightarrow_{SU}^{*} \bot$ then E is not unifiable.

  3. If $E \Rightarrow_{SU}^{*} E'$ with E' in solved form, then $\sigma_{E'}$ is an mgu of E.

Proof:
(1) We have to show this for each of the rules. Let's treat the case for the 4th rule here. Suppose σ is a unifier of $x \doteq t$, that is, $x\sigma = t\sigma$. Thus, $\sigma \circ \{x \mapsto t\} = \sigma[x \mapsto t\sigma] = \sigma[x \mapsto x\sigma] = \sigma$. Therefore, for any equation $u \doteq v$ in E: $u\sigma = v\sigma$, iff $u\{x \mapsto t\}\sigma = v\{x \mapsto t\}\sigma$. (2) and (3) follow by induction from (1) using Proposition 3.22. $\square$

Main Unification Theorem

Theorem 3.24:

E is unifiable if and only if there is a most general unifier σ of E, such that σ is idempotent and $dom(\sigma) \cup codom(\sigma) \subseteq var(E)$.