First Order Predicate Logic
4. Resolution

June 23, 2020

Resolution in FOL: Basic Idea

Method for showing inconsistency of a set of clauses: infer $\mathbb{F}$ by resolution

Propositional resolution: $\left.\begin{array}{l} A \vee B \\ \overline{A} \vee C \end{array}\right\} \vdash B \vee C$

Predicate logic: $\left.\begin{array}{l} P[a,b] \vee Q[a] \\ \overline{P}[a,b] \vee R[b] \end{array}\right\} \vdash Q[a] \vee R[b]$

$\left.\begin{array}{l} P[x,b] \vee Q[x] \longrightarrow P[a,b] \vee Q[a] \\ \overline{P}[a,b] \vee R[b] \end{array}\right\} \vdash Q[a] \vee R[b]$ substitution $\{x \to a\}$ "unifier"

$\left.\begin{array}{l} P[x,f[b]] \vee Q[x] \longrightarrow P[a,f[b]] \vee Q[a] \\ \overline{P}[a,f[y]] \vee R[y] \longrightarrow \overline{P}[a,f[b]] \vee R[b] \end{array}\right\} \vdash Q[a] \vee R[b]$

$\{x \to a,\ y \to b\}$

In general:

$\mathcal{A}_1, \mathcal{A}_2$: atoms, $\mathcal{C}_1, \mathcal{C}_2$: clauses, σ: most general unifier of $\mathcal{A}_1, \mathcal{A}_2$

$$\left.\begin{array}{l} \mathcal{A}_1 \vee \mathcal{C}_1 \\ \overline{\mathcal{A}}_2 \vee \mathcal{C}_2 \end{array}\right\} \vdash (\mathcal{C}_2 \vee \mathcal{C}_2)\sigma \quad \text{resolvent}$$

$\mathcal{A}_1$ and $\overline{\mathcal{A}}_2$ are the literals resolved upon.

Substitution and Resolution: Examples

$$\begin{array}{l}\mathcal{C}_1 : P[x] \vee Q[x] \\ \mathcal{C}_2 : \overline{P}[f[y]] \vee R[y]\end{array} \qquad \sigma' : \{x \to f[a],\ y \to a\} \qquad \begin{array}{l}\mathcal{C}'_1 : P[f[a]] \vee Q[f[a]] \\ \mathcal{C}'_2 : \overline{P}[f[a]] \vee R[a]\end{array}$$

$\mathcal{C}'_1, \mathcal{C}'_2$ are instances of $\mathcal{C}_1, \mathcal{C}_2$ (ground, because they contain no variables).
σ' is an unifier of $\{P[x], \overline{P}[f[y]]\}$.

The resolvent of $\mathcal{C}'_1$ and $\mathcal{C}'_2$ is

$$\mathcal{C}'_3 : Q[f[a]] \vee R[a]$$

$$\begin{array}{l}\mathcal{C}_1 : P[x] \vee Q[x] \\ \mathcal{C}_2 : \neg P[f[y]] \vee R[y]\end{array} \qquad \sigma^* : \{x \to f[y]\} \qquad \begin{array}{l}\mathcal{C}^*_1 : P[f[y]] \vee Q[f[y]] \\ \mathcal{C}^*_2 : \neg P[f[y]] \vee R[y]\end{array}$$

$\mathcal{C}^*_1, \mathcal{C}^*_2$ are instances of $\mathcal{C}_1, \mathcal{C}_2$. $\mathcal{C}'_1, \mathcal{C}'_2$ are instances of $\mathcal{C}^*_1, \mathcal{C}^*_2$ ($\{y \to a\}$).
σ^* is an unifier of $\{P[x], \overline{P}[f[y]]\}$. σ^* is more general then σ', because σ^* can be "specialized" to σ' by composition with another substitution:

$$\{x \to f[a],\ y \to a\} \ = \ \{x \to f[y]\} \circ \{y \to a\}$$

σ^* is the most general unifier of $\{P[x], \overline{P}[f[y]]\}$.

The resolvent of $\mathcal{C}^*_1$ and $\mathcal{C}^*_2$ is $\mathcal{C}^*_3 : Q[f[y]] \vee R[y]$.
$\mathcal{C}'_3$ is and instance of $\mathcal{C}^*_3$. $\mathcal{C}^*_3$ is also the resolvent of $\mathcal{C}_1$ and $\mathcal{C}_2$, because is uses the most general unifier of the literals resolved upon.

Substitution: Definition

A substitution is a finite set of pairs (replacements) $v \to t$ between a variable and a different term:

$$\{v_1 \to t_1, ..., v_n \to t_n\},$$

where for each $i \neq j$: $v_i \neq v_j$.

Application of a substitution σ to an expression $\mathcal{E}$:
$\mathcal{E}\sigma$ is the expression obtained by replacing simultaneously in $\mathcal{E}$ each occurrence of every variable v_i present in the substitution by its corresponding term t_i.

Example: $f[z, a, g[x], y]\{x \to z,\ z \to h[a, y]\} = f[h[a, y], a, g[z], y]$.

Remark. For formulae ϕ where all free variables are implicitely universally quantified, we have:

$$\phi \models \phi\sigma$$

Substitution: Composition

$$(\mathcal{E}\sigma)\theta = \mathcal{E}(\sigma \circ \theta)$$

$\sigma \circ \theta$ is the composition of σ and θ.

Composition rule:

$$\sigma \circ \theta$$
$$\{x_1 \to t_1, ..., x_n \to t_n\} \circ \{y_1 \to u_1, ..., y_n \to u_n\}$$
is the union of the two sets:

$\{x_1 \to t_1\theta, ..., x_n \to t_n\theta\}$ without the elements for which $x_j = t_j\theta$,
$\{y_1 \to u_1, ..., y_n \to u_n\}$ without the elements for which $y_i \in \{x_1, ..., x_n\}$.

Composition of substitutions ...

- ...is associative: $(\sigma \circ \theta) \circ \lambda = \sigma \circ (\theta \circ \lambda)$
- ...is not commutative: $\sigma \circ \theta \neq \theta \circ \sigma$
- ...has neutral element: $\{\}$ (the empty substitution)

Substitution: Composition Examples

Example 1:

$$\theta = \{x \to f[y], y \to z\}$$
$$\lambda = \{x \to a, y \to b, z \to y\}$$

$$\{x \to f[b], y \to y, x \to a, y \to b, z \to y\}$$
$$\theta \circ \lambda = \{x \to f[b], z \to y\}$$

Example 2:

$$\theta_1 = \{x \to a, y \to f[z], z \to y\}$$
$$\theta_2 = \{x \to b, y \to z, z \to g[x]\}$$

$$\{x \to a, y \to f[g[x]], z \to z, x \to b, y \to z, z \to g[x]\}$$
$$\theta_1 \circ \theta_2 = \{x \to a, y \to f[g[x]]\}$$

Unification

The substitution σ is a unifier of $\{\mathcal{E}_1, ..., \mathcal{E}_n\}$ iff $\mathcal{E}_1\sigma = ... = \mathcal{E}_n\sigma$.
The set $\{\mathcal{E}_1, ..., \mathcal{E}_n\}$ is unifiable iff there exists an unifier of it.

A unifier σ of a set $\mathcal{S}$ of expressions is a most general unifier (MGU) of it iff
for each unifier θ of the set $\mathcal{S}$ there exists a substitution λ such that

$$\theta = \sigma \circ \lambda.$$

Examples:

- $\{Q[a, y],\ Q[x, f[b]]\}$, MGU: $\{x \to a, y \to f[b]\}$
- $\{P[x],\ P[y]\}$, MGUs: $\{x \to y\},\ \{y \to x\}$
- $\{P[a],\ P[b]\}$, not unifiable
- $\{P[f[x]],\ P[g[y]]\}$, not unifiable
- $\{P[x],\ P[f[x]]\}$, not unifiable ($\{x \to f[x]\}$ not correct)
  $\{P[x],\ P[f[x]]\}\{x \to f[x]\} = \{P[f[x]],\ P[f[f[x]]]\}$
- $\{P[x],\ Q[y]\}$ not unifiable
- $\{Q[x, f[x]],\ Q[y, y]\}$, not unifiable
  $\{x \to y\} : \{Q[y, f[y]],\ Q[y, y]\}$
  $\{x \to f[y],\ y \to f[y]\} : \{Q[f[y], f[f[y]]],\ Q[f[y], f[y]]\}$

Unification Algorithm: Disagreement Set

Unification algorithm for finding a most general unifier, or its nonexistence.

The disagreement set of a nonempty set $\mathcal{S}$ of expressions is obtained by

- locating the first position (starting from the left) at which not all the expressions in $\mathcal{S}$ have exactly the same symbol and then
- extracting from each expression in $\mathcal{S}$ the subexpression that begins with the symbol occupying that position.

Examples:

$\mathcal{S}$	$\mathcal{D}$	σ
$\{P[\,g\,[x], x, f[g[y]]],\ P[\,z\,, a, f[z]]\}$	$\{g[x], z\}$	$\{z \to g[x]\}$
$\{P[g[x],\,x\,, f[g[y]]],\ P[g[x],\,a\,, f[g[x]]]\}$	$\{x, a\}$	$\{z \to g[a], x \to a\}$
$\{P[g[a], a, f[g[\,y\,]]],\ P[g[a], a, f[g[\,a\,]]]\}$	$\{y, a\}$	$\{z \to g[a], x \to a, y \to a\}$

Unification Algorithm: Imperative Version

Input: set $\mathcal{S}$ of expressions.

1. $k := 0$, $\mathcal{S}_k := \mathcal{S}$, $\sigma_k := \{\}$
2. If $\mathcal{S}_k$ is singleton then stop; σ_k is MGU of $\mathcal{S}$. Otherwise find the disagreement set $\mathcal{D}_k$ of $\mathcal{S}_k$.
3. If there exists v_k, $t_k \in \mathcal{D}_k$ s.t. v_k is a variable which does not occur in t_k, go to **4**. Otherwise, stop; $\mathcal{S}$ is not unifiable.
4. Let $\sigma_{k+1} = \sigma_k \circ \{v_k \to t_k\}$ and $\mathcal{S}_{k+1} = \mathcal{S}_k\{v_k \to t_k\}$.
5. $k = k + 1$ and go to **2**.

Resolution: the Inference Rule

If two or more literals (with the same sign) of a clause $\mathcal{C}$ have MGU σ, then $\mathcal{C}\sigma$ is called a factor of $\mathcal{C}$.

Example: Let $\mathcal{C}: \ P[x] \ \vee \ P[a] \ \vee \ Q[f[x]] \ \vee Q[f[a]]$ be a clause. Then the MGU is $\sigma = \{x \to a\}$ and $\mathcal{C}\sigma: \ P[a] \ \vee Q[f[a]]$ is a factor of $\mathcal{C}$.

Let $\mathcal{C}_1$ and $\mathcal{C}_2$ be two clauses with *no variables in common*. Let $\mathcal{A}_1$ and $\overline{\mathcal{A}}_2$ be two literals in $\mathcal{C}_1$ and $\mathcal{C}_2$, respectively. If the atoms $\mathcal{A}_1$ and $\mathcal{A}_2$ have a MGU σ, then the clause $\mathcal{C}_1\sigma \ \vee \ \mathcal{C}_2\sigma$ is a binary resolvent of $\mathcal{C}_1$ and $\mathcal{C}_2$.

Example: $\mathcal{C}_1: \ P[x] \vee Q[x], \ \ \mathcal{C}_2: \ \overline{P}[a] \vee R[x]$.
By renaming x with y in $\mathcal{C}_2$, we have $\mathcal{C}_2: \overline{P}[a] \vee R[y]$
$\sigma = \{x \to a\}$ is the MGU of the literals $P[x]$ and $\overline{P}[a]$.
The binary resolvent of $\mathcal{C}_1$ and $\mathcal{C}_2$ is $Q[a] \vee R[y]$.

From two clauses we obtain a set of resolvents:

- From each clause $\mathcal{C}, \mathcal{C}'$ one obtains a set of factors $\mathcal{F}, \mathcal{F}'$;
- From each pair from $(\{\mathcal{C}\} \cup \mathcal{F}) \times (\{\mathcal{C}'\} \cup \mathcal{F}')$, a possible binary resolvent may be generated. (If there are several MGUs, considering only one of them is sufficient).

Resolution: Proof Method

Resolution: (Robinson, 1965)

- is an inference rule which generates resolvents from a set of clauses
- is a refutation proof procedure: empty clause is tried to be derived from a set of clauses
- is refutationally complete: a set of clauses is unsatisfiable iff the empty clause can be derived

Resolution together with refutation and normal form:

Given: formulae φ_1, ..., φ_n

Prove: ψ by resolution.

1. Bring φ_1, ..., φ_n, ..., $\neg\psi$ into standard form: set of clauses.
2. Use the resolution inference rule to derive new clauses in all possible ways, including by using the new produced clauses.
3. If the empty clause appears, stop: contradiction found, ψ is proved.
4. If no new clause can be produced (and the empty clause is not found), then the proof fails.

Only for unsatisfiable sets of clauses the procedure is guaranteed to stop, if the set is satisfiable then it may continue forever.

"Semi–decision" procedure

Resolution: Correctness

Φ is unsatisfiable

$\models \neg \wedge \Phi$

Completeness $\Downarrow$ **iff** $\Uparrow$ Correctness

$\wedge\Phi \vdash^* \mathbb{F}$

there is a deduction of
the empty clause from Φ

Correctness of the inference rule. Take $\mathcal{A} = \mathcal{A}_1\sigma = \mathcal{A}_2\sigma$.

$$\left.\begin{array}{l} \mathcal{A}_1 \vee \mathcal{C}_1 \\ \overline{\mathcal{A}_2} \vee \mathcal{C}_2 \end{array}\right. \begin{array}{c} \longrightarrow \\ \text{instantiation} \\ \longrightarrow \end{array} \left.\begin{array}{l} \mathcal{A} \vee \mathcal{C}_1\sigma \\ \overline{\mathcal{A}} \vee \mathcal{C}_2\sigma \end{array}\right\} \underset{\text{resolution}}{\longrightarrow} \mathcal{C}_1\sigma \vee \mathcal{C}_2\sigma$$

From $\varphi \models \varphi\sigma$ and $\left.\begin{array}{l} \psi \vee \varphi_1 \\ \overline{\psi} \vee \varphi_2 \end{array}\right\} \models \varphi_1 \vee \varphi_2$ we obtain:

$$\left.\begin{array}{l} \mathcal{A}_1 \vee \mathcal{C}_1 \\ \overline{\mathcal{A}_2} \vee \mathcal{C}_2 \end{array}\right\} \models \mathcal{C}_1\sigma \vee \mathcal{C}_2\sigma$$

Correctness of method. Due to correctness of the inference rule:

If $\wedge\Phi \vdash^* \mathbb{F}$, then $\wedge\Phi \models \mathbb{F}$.

Thus if some interpretation $I \models \wedge\Phi$, then $I \models \mathbb{F}$ (impossible).

Completeness of Resolution: Herbrand Interpretations

Running example:

$$P[a],\ \underset{x}{\forall}(\overline{P}[x] \vee P[f[x]]) \models P[f[f[a]]]$$

(1) $P[a]$ (2) $\overline{P}[x] \vee P[f[x]]$ (3) $\overline{P}[f[f[a]]]$

The Herbrand Universe: an "universal" domain.

$H = \{a,\ f[a],\ f[f[a]],\ \ldots,\ f[f[\ldots[a]\ldots]],\ \ldots\}$ (enumerable!)

Herbrand interpretations and their "universality".

$$I = \begin{cases} D_I = H \\ a_I = a \\ f_I : H \to H,\ f_I[a] = f[a],\ f_I[f[a]] = f[f[a]],\ \ldots \\ P_I : H \to \{\mathbb{T}, \mathbb{F}\},\ P_I[a] = \mathbb{T}\ (P[a]),\ P_I[f[a]] = \mathbb{F}\ (\overline{P}[f[a]]), \ldots \end{cases}$$

Universality: From $J \models \varphi$ construct H-interpretation $I \models \varphi$:

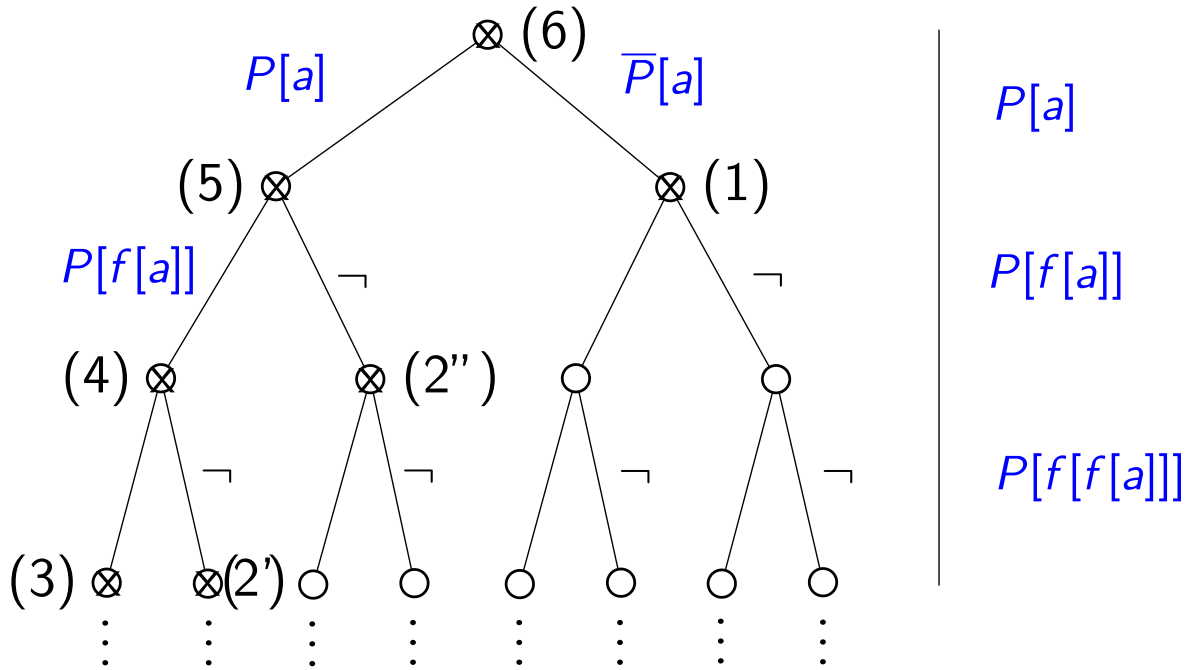
$$P_I[f[f[\ldots[a]\ldots]]] = P_J[f_J[f_J[\ldots[a_J]\ldots]]]$$

It is sufficient to check satifiability on Herbrand interpretations !

(But this holds only for formulae in Skolem standard form.)

Completeness of Resolution: Semantic Tree

Running example: (1) $P[a]$ (2) $\overline{P}[x] \vee P[f[x]]$ (3) $\overline{P}[f[f[a]]]$
Herbrand interpretation: $\{P[a],\ \overline{P}[f[a]],\ \ldots, \overline{P}[f[f[\ldots[a]\ldots]]],\ \ldots\}$
Representation of all interpretations: binary tree ("semantic tree")



(2') $\overline{P}[f[a]] \vee P[f[f[a]]]$ $x \to f[a]$ (2'') $\overline{P}[a] \vee P[f[a]]$ $x \to a$ (4) $\overline{P}[f[a]]$ (5) $\overline{P}[a]$ (6) $\mathbb{F}$

Unsatisfiable set $\longrightarrow$ closed semantic tree

Resolution $\longrightarrow$ Unsatisfiable set of ground instances

Completeness of Resolution: Herbrand Theorem

For every unsatisfiable set of clauses Φ there exists an unsatisfiable finite set of instances of Φ.

Herbrand proving procedure: Instantiate incrementally using the Herbrand universe, and check consistency at every step.

If the set is unsatisfiable, the theorem insures that this will finish.

If the set is satisfiable, then the procedure runs for ever: semidecision procedure.

Resolution: more efficient, because unification finds the instantiations which can lead to contradiction.

H-Theorem insures that a ground proof by resolution exists.

A resolution step involving variables simulates several ground resolutions: Lifting Lemma shows that ground proof can be lifted to a general resolution proof.

Same problem as H-procedure: resolution for a satisfiable set may not finish (however possible termination is easier to detect: no new clauses).

Satisfiability in first order predicate logic is undecidable.

Gödel Completeness Theorem

In classical logic:

$$\Phi \text{ inconsistent (contradictory)} \quad \textbf{iff} \quad \Phi \models \mathbb{F}$$

We proved:

If Φ unsatisfiable, **then** Φ inconsistent (contradictory)

Thus, by contraposition,

If Φ consistent, **then** Φ satisfiable*

in a constructive manner: the satisfying interpretation (model) of the set is constructed from the syntactic material available in the formula.

This has a strong computational significance: when we implement algorithms relevant to a certain logic theory, we can always use as elements of the respective domains the ground atoms arising from the Skolem normal form of that theory.

* This is Gödel completeness theorem, a milestone in the development of classic logic, which gets an important additional significance in the current technological situation.