Automating deduction in non-classical logics: Signed Resolution for Many-Valued Logics

Luís M. Augusto

Universidade Aberta

MEMC, 2014

Presuppositions

- **Logical systems and classical logic:**
  - Logical systems:
    - truth-functionality, interpretation, propositional logic, FOL, etc.
  - Classical logic (CL):
    - CL syntax, CL semantics, etc.
  - Normal forms and clausal logic:
    - PNF, CNF, DNF, etc.
- **Automated theorem proving (ATP):**
  - Herbrand's theorem (see, e.g., Chang & Lee, 1973):
    - Herbrand universe, skolemization, ground terms, semantic trees, etc.
  - Resolution calculus (see, e.g., Leitsch, 1997) :
    - Binary resolution, factoring, unification, etc.

Outline

Automated theorem proving (ATP)

Given a formula (conclusion) A and a – possibly empty – set of formulae (premises) Γ in a logical system S, one often wishes to find answers for the questions

1. **Deduction problem (DP)**: $\Gamma \vdash_S A$?, i.e., whether A is a *theorem*, or a *logical consequence* of Γ, in S (i.e., $\vdash_S \Gamma \rightarrow A$, or $\vdash_S A$ for $\Gamma = \emptyset$).
2. **Decision problem**: is DP decidible (i.e., is there an algorithm for PD): Yes or No?
   - Answers:
     - S = Classical propositional logic: YES
     - S = Classical FOL: NO (*Church-Turing theorem*) (BUT...)
3. **ATP**: is the algorithm for PD fully automatizable, namely in a computer program?

Many-valued logics: Importance

Many-valued logics

- have many practical applications in pure and applied mathematics, namely in computer science. E.g.,
  - switching theory
  - logic programming
  - hardware verification
  - natural language processing
- generalize CL, reason why they are important tools to investigate into fundamental aspects of classical systems. E.g.,
  - verification of the independence of axioms of CPL

Validity and unsatisfiability

☛

Definition (*validity*) Let Γ be a set of formulae and A a formula entailed from Γ; we say that a formula A is *valid* iff there is no interpretation assigning the value true to all the members of Γ (the premises) and false to A (the conclusion), and we write $\Gamma \models A$ ($\models A$, if $\Gamma = \emptyset$). A formula is said to be *invalid* iff it is not valid.

*

☛

Theorem (*deduction theorem*). $\Gamma \models A$ iff $\Gamma \cup \{\neg A\}$ is unsatisfiable.

DP and (un)satisfiability

☛

Theorem (*deduction theorem*). Given a set of formulae $\Gamma = \{B_1, ..., B_n\}$ and a formula A, A is a logical consequence of Γ iff the formula $((B_1 \land ... \land B_n) \to A)$ is valid. Equivalently, a formula A is a logical consequence of a set of formulae $\Gamma = \{B_1, ..., B_n\}$ iff the formula $(B_1 \land ... \land B_n \land \neg A)$ is unsatisfiable.

- In an adequate logical system, this allows us to test for DP via the semantic notion of (un)satisfiability: A is a logical consequence of Γ iff the negation of $((B_1 \land ... \land B_n) \to A)$ is *refuted*, i.e., iff $\not\models \neg(\Gamma \to A)$, where $\Gamma = \bigwedge_i B_i \in \Gamma$.

SAT

☛

Definition (*the Boolean satisfiability problem, or SAT*). Given a formula $A(x_1, ..., x_n)$, it is asked if A can be evaluated to T by some assignment of the truth values T or F to the x_i, $1 \leq i \leq n$. We say that a (propositional) formula $A(x_1, ..., x_n)$ is *satisfiable* if truth values can be assigned to its variables x_i in such a way as to make A true. Otherwise, A is said to be *unsatisfiable*.

Herbrand's theorem

☛

Theorem (*Herbrand, 1930 - version I*). A set $\mathscr{C}$ of clauses is unsatisfiable iff corresponding to every complete semantic tree of $\mathscr{C}$, there is a finite closed semantic tree.

*

☛

Theorem (*Herbrand, 1930 - version II*). A set $\mathscr{C}$ of clauses is unsatisfiable iff there is a finite unsatisfiable set $\mathscr{C}'$ of ground instances of $\mathscr{C}$.

H-unsatisfiability

☛

Theorem A set $\mathscr{C}$ of clauses is unsatisfiable iff $\mathscr{C}$ is false under all the H-interpretations, i.e., iff it is *H-unsatisfiable*.

A semantic tree allows us to check H-unsatisfiability (cf. Herbrand's theorem, version I).

The empty clause

☛

Theorem A formula F is unsatisfiable iff it is possible to derive a contradiction from F, i.e., $F \models G \wedge \neg G$.

Let $G \wedge \neg G = \square$, where $\square$ denotes the empty clause. Then $\square \equiv \bot$, because the empty clause has no literal that can be satisfied by any interpretation. Therefore, if we can obtain $\square$ from a set of clauses $\mathscr{C}$, then $\mathscr{C}$ is unsatisfiable.

The resolution principle

☛

Theorem A resolvent $C = \left(C_1' \vee C_2'\right)\sigma$ of two clauses $C_1 = C_1' \vee L_1$ and $C_2 = C_2' \vee \neg L_2$ is a logical consequence of $C_1 \wedge C_2$, i.e.,

$$\frac{C_1' \vee L_1 \quad C_2' \vee \neg L_2}{\left(C_1' \vee C_2'\right)\sigma}, \qquad \sigma = mgu\left(L_1, L_2\right)^*.$$

* For FOL; in the propositional case, a resolvent is obtained iff $L_1 = L_2$.

☛

Definition A *(resolution) deduction* of C from a set of clauses $\mathscr{C}$ is a finite sequence $C_1, C_2, ..., C_k$ of clauses such that each C_i either is a clause in $\mathscr{C}$ or a resolvent of clauses preceding C_i, and $C_k = C$. We call the deduction of the empty set $\square$ from $\mathscr{C}$ a *refutation*, or *proof* of $\mathscr{C}$.

Example 1

Let $\mathscr{C} = \{\neg P(x) \vee Q(x), P(f(a)), \neg Q(z)\}$. We apply binary resolution to this set of clauses:

1.	$\neg P(x) \vee Q(x)$	
2.	$P(f(a))$	
3.	$\neg Q(z)$	
4.	$Q(f(a))$	res. 1, 2; $\sigma = \{x \mapsto f(a)\}$
5.	$\square$	res. 3, 4; $\theta = \{z \mapsto f(a)\}$

Note that $H_{\mathscr{C}} = \{a, f(a), f(f(a)), \ldots\}$ and $H(\mathscr{C}) = \{P(a), Q(a), P(f(a)), Q(f(a)), \ldots\}$, $H_{\mathscr{C}}$ and $H(\mathscr{C})$ denote the Herbrand universe and the Herbrand base of $\mathscr{C}$, respectively.

Example 1 (cont.)

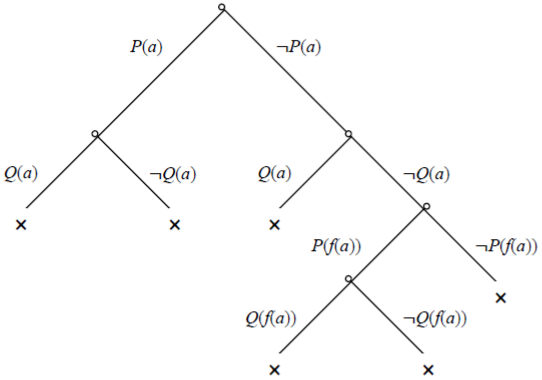

Figure : Closed semantic tree for $\mathscr{C} = \{\neg P(x) \vee Q(x), P(f(a)), \neg Q(z)\}$.Note that $A(\mathscr{C}) = \{P(a), Q(a), P(f(a)), Q(f(a))\}$, $A(\mathscr{C}) \subseteq H(\mathscr{C})$.

Interpretation and logical matrix

- An *interpretation* for some $\mathscr{L}^{Prop} = (F, O_1, ..., O_m)$, where F is a set of formulae and $O_1, ..., O_m$ are finitary operations over F, can be provided by an interpretation structure $\mathfrak{A} = (\mathscr{A}, f_1, ..., f_m)$ where $\mathscr{A}$ is the range of semantic correlates of $\mathscr{L}^{Prop}$.
- A *logical matrix* $\mathfrak{M}$ is a pair $(\mathfrak{A}, D)$ where $\mathfrak{A}$ is an algebra similar to a propositional language $\mathscr{L}^{Prop}$ and $D \subseteq \mathscr{A}$ is a non-empty subset of the universe of $\mathfrak{A}$ with D the *designated values* of $\mathfrak{M}$.

Validity, tautologousness and contradictoriness in many-valued logics

The set D of designated values allows for a natural generalization of the classical notions of validity, tautologousness, and contradictoriness to the many-valued logics. E.g.,

☛

Definition (*validity in many-valued logics*). Given a designated set $D \subset W, D \neq \emptyset$, we say that an inference is valid iff it preserves designated values, i.e.,

$$\Gamma \models_D A \quad \text{iff for every interpretation } \mathscr{I}, \text{ whenever } val_{\mathscr{I}}(B) \in D,$$

$$\text{for all } B \in \Gamma, val_{\mathscr{I}}(A) \in D.$$

Content of a logical matrix

- With each matrix $\mathfrak{M}$ there is associated a set of formulae

$$E(\mathfrak{M}) = \left\{\phi \in F : h\phi \in D \text{ for any } h \in Hom\left(\mathscr{L}^{Prop}, \mathfrak{A}\right)\right\}$$

called the *content* of $\mathfrak{M}$, and for any such matrix $\mathfrak{M}$ we define the relation $\models_{\mathfrak{M}}$ for any $X \subseteq F, \phi \in F$,

$$X \models_{\mathfrak{M}} \phi \text{ iff for every } h \in Hom\left(\mathscr{L}^{Prop}, \mathfrak{A}\right), h\phi \in D$$

$$\text{whenever } hX \subseteq D.$$

- In fact, for any logical system S,

$$E(\mathfrak{M}_S) = \{\phi \mid \models_S \phi\} = TAUT(\mathrm{S})$$

A criterion for many-valuedness

☛

Proposition (*Malinowski, 1993*) A logical matrix $\mathfrak{M}_{n>2}$ determines a many-valued logic iff for *no* matrix $\mathfrak{M}_2$ for $\mathscr{L}^{Prop}$ it is the case that

1. $E(\mathfrak{M}_{n>2}) = E(\mathfrak{M}_2)$;
2. $\models_{\mathfrak{M}_{n>2}} = \models_{\mathfrak{M}_2}$.

The finitely many-valued logic $Ł_3$

- Logical matrix: $\mathfrak{L}_3 = (\{T, I, F\}, \neg, \rightarrow .\wedge, \vee, \leftrightarrow, \{T\})$
- Truth tables:

A	$\neg A$
T	F
I	I
F	T

$\rightarrow$	T	I	F
T	T	I	F
I	T	T	I
F	T	T	T

$\vee$	T	I	F
T	T	T	T
I	T	I	I
F	T	I	F

$\wedge$	T	I	F
T	T	I	F
I	I	I	F
F	F	F	F

$\leftrightarrow$	T	I	F
T	T	I	F
I	I	T	I
F	F	I	T

- $E(\mathfrak{L}_3) \subsetneq E(\mathfrak{M}_2)$

The fuzzy (i.e. infintely many-valued) logic Ł$_\aleph$

- $\mathfrak{L}_\aleph = ([0,1], \neg, \rightarrow .\wedge, \vee, \leftrightarrow, 1 \text{ or } \varepsilon \in (0,1])$
- Truth functions: for all $x, y \in [0,1]$,

$$x \rightarrow y = \begin{cases} 1 & \text{if} \quad x \leq y \\ 1 - x + y & \text{if} \quad x > y \end{cases}$$

$$\neg x = 1 - x$$

Also:

$$x \vee y = max(x,y)$$

$$x \wedge y = min(x,y)$$

$$x \leftrightarrow y = 1 - |x - y|$$

- $E(\mathfrak{L}_\aleph) \subsetneq E(\mathfrak{M}_2)$

Other relevant many-valued logics

- **Finitely many-valued:** B_3^I, B_3^E, K_3^S, K_3^W, P_n (n finite) (cf. Bolc & Borowik, 1992; Rescher, 1969)
- **Infinitely many-valued:**
  - Fuzzy logics: LG (Gödel logic), LΠ (product logic)
  - Also: P_n (n infinite) (cf. e.g., Rescher, 1969)
- These logics have quantified calculi: ex.: $\mathrm{q}Ł_3$, qLG, etc.
- With some exceptions (e.g., qLΠ), they have adequate axiom systems.

MVSAT

- Satisfiability for a many-valued formula ϕ (MVSAT) can be expressed as

☛

Is it ever the case that ϕ takes a truth value $x \in D$?

- The classical duality between validity and satisfiability is extended to many-valued logics in the following way: A formula ϕ is D-valid iff it is not $\overline{D}$-satisfiable, or, by defining sets D^+ abd D^-, $W = D^+ \cup D^-$, ϕ is D^+-valid iff it is not D^--satisfiable.

Signed logic

- By always "marking" a many-valued formula with the truth value(s) that it takes or can take – i.e., its *signal* – we obtain *signed logic*.
- This formalism allows us to generalize the important classical notions of (in)validity and (un)satisfiability to the many-valued logics. As is well-known, a valuation in CL is indicated by P and $\neg P$; given $W_2 = \{\mathrm{T}, \mathrm{F}\}$, we can *sign* (i.e., give a sign to) P and $\neg P$ as $\{\mathrm{T}\}[P]$ and $\{\mathrm{F}\}[P]$, respectively.
- This strategy allows the extension of classical bivalent reasoning to many-valued logics by signing many-valued formulae as $S[\phi]$ or $(W \setminus S)[\phi]$ (i.e., $\overline{S}[\phi]$), for a given $S \subseteq W$.

Signed clausal logic (SCL)

- By allowing the building of CNFs, SCL allows the direct application of the resolution principle to many-valued logics. Just as in CL, in SCL
  - every signed formula ϕ is equivalent to a signed formula (expression) ϕ_1 in DNF and to a signed formula (expression) ϕ_2 in CNF;
  - $\neg\phi_1 \equiv \phi_2$ and $\neg\phi_2 \equiv \phi_1$;
  - $\bigwedge_{i=1}^{n} \overline{S}\,[A_i]$ is a refutation of $\bigvee_{i=1}^{n} S\,[A_i]$;
  - a set of signed clauses $\mathscr{C}$ is unsatisfiable iff it is H-unsatisfiable.
- Thus, all that is required is a set of transformation rules for the translation of any signed formula into a signed formula in clausal form, i.e., a signed formula expression (SFE).

Transformation rules (Baaz et al., 2001)

☛

Definition Given a pair (ϕ, Φ), where ϕ is a *signed formula* and Φ is a *signed formula expression*, $\phi \Longrightarrow \Phi$ is a *transformation rule* (TR). The rule is *correct* iff $\phi \equiv \Phi$ is valid.

- A propositional TR is an expression of the form

$$S\left[O\left(A_1,...,A_n\right)\right] \Longrightarrow \bigwedge_{i \in I} \bigvee_{j \in J} S_{ij}\left[A'_{ij}\right], \qquad A'_{ij} \in \{A_1,...,A_n\}.$$

- A quantifier TR is an expression of the form

$$S\left[\left(Qx\right)A\left(x\right)\right] \Longrightarrow \bigwedge_{i \in I}\left(\bigvee_{j \in J}\left(\exists x\right)S_{ij}\left[A\left(x\right)\right] \vee \bigvee_{k \in K}\left(\forall x\right)S_{ik}\left[A\left(x\right)\right]\right).$$

Translation into SCL

- For $\phi = S[O(A_1, ..., A_n)]$:
  - $DNF(\phi) := \bigvee_{\substack{v_1, ..., v_n \in W \\ \tilde{O}(v_1, ..., v_n) \in S}} \bigwedge_{i=1}^{n} \{v_i\}[A_i]$
  - $CNF(\phi) := \bigwedge_{\substack{v_1, ..., v_n \in W \\ \tilde{O}(v_1, ..., v_n) \in \overline{S}}} \bigvee_{i=1}^{n} \overline{\{v_i\}}[A_i]$
- For $\phi = S[(Qx)A(x)]$, $\mathscr{V}$ is the distribution of ϕ:
  - $DNF(\phi) :=$ $\bigvee_{\substack{\emptyset \subset \mathscr{V} \subseteq W \\ \tilde{Q}(\mathscr{V}) \in S}} \left((\forall x)\mathscr{V}[A(x)] \wedge \bigwedge_{v_i \in \mathscr{V}} (\exists x)\{v_i\}[A(x)]\right)$
  - $CNF(\phi) :=$ $\bigwedge_{\substack{\emptyset \subseteq \mathscr{V} \subseteq W \\ \tilde{Q}(\mathscr{V}) \in \overline{S}}} \left((\exists x)\overline{\mathscr{V}}[A(x)] \vee \bigvee_{v_i \in \mathscr{V}} (\forall x)\overline{\{v_i\}}[A(x)]\right)$

Example 2

We want to compute the CNF of $\{\mathrm{I}\}[A \to_{Ł3} B]$.

1. We compute the DNF of $\{\mathrm{T},\mathrm{F}\}[A \to_{Ł3} B]$, i.e.,

$$\bigvee_{\substack{v_1, v_2 \in \{\mathrm{T},\mathrm{I},\mathrm{F}\} \\ v_1 \to v_2 \neq \mathrm{I}}} (\{v_1\}[A] \wedge \{v_2\}[B])$$

   The examination of the truth table gives us the DNF:

$$(\{\mathrm{T}\}[A] \wedge \{\mathrm{T}\}[B]) \vee (\{\mathrm{T}\}[A] \wedge \{\mathrm{F}\}[B]) \vee (\{\mathrm{I}\}[A] \wedge \{\mathrm{T}\}[B]) \vee (\{\mathrm{I}\}[A] \wedge \{\mathrm{I}\}[B]) \vee$$
$$(\{\mathrm{F}\}[A] \wedge \{\mathrm{T}\}[B]) \vee (\{\mathrm{F}\}[A] \wedge \{\mathrm{I}\}[B]) \vee (\{\mathrm{F}\}[A] \wedge \{\mathrm{F}\}[B])$$

2. We now compute the CNF of $\{\mathrm{I}\}[A \to_{Ł3} B]$:

$$(\{\mathrm{I},\mathrm{F}\}[A] \vee \{\mathrm{I},\mathrm{F}\}[B]) \wedge (\{\mathrm{I},\mathrm{F}\}[A] \vee \{\mathrm{T},\mathrm{I}\}[B]) \wedge (\{\mathrm{T},\mathrm{F}\}[A] \vee \{\mathrm{I},\mathrm{F}\}[B]) \wedge$$
$$(\{\mathrm{T},\mathrm{F}\}[A] \vee \{\mathrm{T},\mathrm{F}\}[B]) \wedge (\{\mathrm{T},\mathrm{I}\}[A] \vee \{\mathrm{I},\mathrm{F}\}[B]) \wedge$$
$$(\{\mathrm{T},\mathrm{I}\}[A] \vee \{\mathrm{T},\mathrm{F}\}[B]) \wedge (\{\mathrm{T},\mathrm{I}\}[A] \vee \{\mathrm{T},\mathrm{I}\}[B]) \equiv (\{\mathrm{T}\}[A] \vee \{\mathrm{F}\}[B]) \wedge (\{\mathrm{I}\}[A] \vee \{\mathrm{I}\}[B])$$

Example 3

The following are the correct TRs for $q\text{Ł}_3$:

$$\begin{array}{lcl}
\{\mathrm{T}\}[(\forall x)A(x)] & \Longrightarrow & (\forall x)\{\mathrm{T}\}[A(x)] \\
\{\mathrm{I}\}[(\forall x)A(x)] & \Longrightarrow & (\exists x)\{\mathrm{I}\}[A(x)] \wedge (\forall x)\{\mathrm{T},\mathrm{I}\}[A(x)] \\
\{\mathrm{F}\}[(\forall x)A(x)] & \Longrightarrow & (\exists x)\{\mathrm{F}\}[A(x)] \\
\{\mathrm{T}\}[(\exists x)A(x)] & \Longrightarrow & (\exists x)\{\mathrm{T}\}[A(x)] \\
\{\mathrm{I}\}[(\exists x)A(x)] & \Longrightarrow & (\exists x)\{\mathrm{I}\}[A(x)] \wedge (\forall x)\{\mathrm{I},\mathrm{F}\}[A(x)] \\
\{\mathrm{F}\}[(\exists x)A(x)] & \Longrightarrow & (\forall x)\{\mathrm{F}\}[A(x)]
\end{array}$$

Example 4

- Let $F = (\forall x) P(x) \rightarrow_{q\text{Ł}_3} (\exists y) P(y)$.
- In Example 2, we obtained the CNF of
  $\{\mathrm{I}\}[A \rightarrow_{\text{Ł}3} B] \equiv (\{\mathrm{T}\}[A] \vee \{\mathrm{F}\}[B]) \wedge (\{\mathrm{I}\}[A] \vee \{\mathrm{I}\}[B])$.
- Thus,
  $\{\mathrm{I}\}[F] \equiv (\{\mathrm{T}\}[(\forall x) P(x)] \vee \{\mathrm{F}\}[(\exists y) P(y)]) \wedge (\{\mathrm{I}\}[(\forall x) P(x)] \vee \{\mathrm{I}\}[(\exists y) P(y)])$.
- By applying the TRs for quantified formulae (Example 3) together with the laws of distributivity, skolemization, and simplifications, we obtain the equisatisfiable formula

$$\{\mathrm{I}\}[F] \equiv_{sat}$$

$$(\{\mathrm{T}\}[P(x)] \vee \{\mathrm{F}\}[P(y)]) \wedge (\{\mathrm{I}\}[P(a)]) \wedge (\{\mathrm{T},\mathrm{I}\}[P(x)] \vee \{\mathrm{I},\mathrm{F}\}[P(y)])$$

The signed SAT problem

☛

A signed literal $S[P]$ *is satisfied* exactly by the interpretations $\mathscr{I}$ such that $val_{\mathscr{I}}(P) \in S$. An interpretation *satisfies* a signed clause iff it satisfies at least one of its signed literals, and it satisfies a signed CNF formula if it satisfies all its clauses. A signed CNF formula is *satisfiable* iff there exists at least one interpretation that satisfies all its signed clauses; otherwise, it is *unsatisfiable*. The signed empty clause $\{\}[C]$ is always unsatisfiable and the signed empty CNF formula is always satisfiable.

Signed resolution: Main inference rules

- Signed binary resolution:

$$\text{(R1)} \qquad \frac{S_1\,[P_1] \vee C_1 \quad S_2\,[P_2] \vee C_2}{((S_1 \cap S_2)\,[P_1] \vee C_1 \vee C_2)\,\sigma}, \quad \sigma = umg\,(P_1, P_2)$$

- Simplification rule:

$$\text{(R2)} \qquad \frac{\{\}\,[P] \vee C}{C}$$

Signed resolution: refinements

- (R3) $\frac{S_1[P_1] \vee C_1 \ldots S_m[P_m] \vee C_m}{(C_1 \vee \ldots \vee C_m)\sigma}$ if $\bigcap_{1 \leq i \leq m} S_i = \emptyset$, $\sigma = mgu(P_1, \ldots, P_m)$
- (R4) $\frac{S_1[P_1] \vee C_1 \quad S_2[P_2] \vee C_2}{(C_1 \vee C_2)\sigma}$ if $S_1 \cap S_2 = \emptyset$, $\sigma = mgu(P_1, P_2)$
- (R5) $\frac{S_1[P_1] \vee \ldots \vee S_m[P_m] \vee C}{((S_1 \cup \ldots \cup S_m)[P_1] \vee C)\sigma}$, $\sigma = mgu(P_1, .., P_m)$
- (R6) $\frac{S_1[P_1] \vee C_1 \ldots S_k[P_k] \vee C_k}{(C_1 \cup \ldots \cup C_k)\sigma}$, $\bigcap_{1 \leq i \leq k} S_i = \emptyset$, $\sigma = mgu(P_i (1 \leq i \leq k))$
- (R7) $\frac{S_1[P_1] \vee C_1 \ldots S_k[P_k] \vee C_k}{(C_1 \cup \ldots \cup C_k)\sigma}$, $\bigcap_{1 \leq i \leq k} S_i = \emptyset$, $\sigma = mgu(P_i (1 \leq i \leq k))$, $P_i\sigma \not<_A Q$ for all $R[Q] \in C_i\sigma$

Example 5

We apply signed resolution to $\{\mathrm{I}\}[F]$ in order to solve MVSAT with respect to this formula (cf. Example 4):

C_1	$\{\mathrm{T}\}[P(x)] \vee \{\mathrm{F}\}[P(y)]$	
C_2	$\{\mathrm{I}\}[P(a)]$	
C_3	$\{\mathrm{T},\mathrm{F}\}[P(x)] \vee \{\mathrm{I},\mathrm{F}\}[P(y)]$	
C_4	$\{\}[P(a)] \vee \{\mathrm{F}\}[P(y)]$	Res. $C_1\theta$ and $C_2\theta$, $\theta = \{x \mapsto a\}$
C_5	$\{\mathrm{T}\}[P(x)] \vee \{\}[P(a)]$	Res. $C_1\lambda$ and $C_2\lambda$, $\lambda = \{y \mapsto a\}$
C_6	$\{\mathrm{F}\}[P(y_1)]$	C_4, (R2) and renaming
C_7	$\{\mathrm{T}\}[P(x_1)]$	C_5, (R2) and renaming
C_8	$\square$	Res. $C_6\sigma$ and $C_7\sigma$, $\sigma = \{x_1 \mapsto c, y_1 \mapsto c\}$, by (R3)

Soundness of signed resolution

☛

Theorem (*soundness of the mvres calculus*). For any set of clauses $\mathscr{C}$, if $\mathscr{C} \vdash_{mvres} \square$, then $\mathscr{C}$ is H-unsatisfiable.

*

Proof.

There is no interpretation that satisfies the empty clause. Thus, $\mathscr{C}$ is unsatisfiable whenever $\square$ is derivable. Besides, given that $\square$ does not have any atom belonging to $A(\mathscr{C}) \subseteq H(\mathscr{C})$ that can be satisfied by an H-interpretation, if $\square$ can be derived from $\mathscr{C}$, then $\mathscr{C}$ is H-unsatisfiable, namely through the subset $\mathscr{C}' \subseteq \mathscr{C}$, $\mathscr{C}'$ is the set of ground clauses of $\mathscr{C}$. □

Completeness of signed resolution

☛

Theorem (*completeness of the mvres calculus*). For any set of clauses $\mathscr{C}$, if $\mathscr{C}$ is H-unsatisfiable, then $\mathscr{C} \vdash_{resmv} \Box$.

Proof.

The proof is by the notion of semantic tree. □

Main theorem of signed resolution

☛

Let ϕ be any closed formula and let $\mathscr{C}_{\overline{U}\Phi}$ be the set of clauses of the clausal translation $\overline{U}\Phi$ of $v[\phi]$ for any truth value $v \in \overline{U}$, $U \subset W$. Then, all interpretations give a truth value $u \in U$ to ϕ iff $\mathscr{C}_{\overline{U}\Phi} \vdash_{mvres} \square$, where *mvres* designates any of the rules (R1)-(R7).

Proof.

($\Rightarrow$) The proof is by the completeness of *mvres*.
($\Leftarrow$) The proof is by the soundness of *mvres*.

□

Example 6

- In Example 5, we obtained the result that F cannot take the truth value I in q$Ł_3$, i.e. $\{\mathrm{I}\}[F]$ is unsatisfiable in q$Ł_3$.
- A look at the matrix of q$Ł_3$ shows that $\overline{D} = \{\mathrm{I}, \mathrm{F}\}$.
- We therefore conclude that $\{\mathrm{T}\}[F]$ is a valid formula in q$Ł_3$.

mvres algorithm

Given any formula ϕ in a many-valued logical system $\mathcal{S}$ with a set of truth values W:

1. Obtain the clausal form $\overline{D}\Phi$ of the signed formula $v[\phi]$, $v \in \overline{D}$, where $D \subset W$ is the set of designated values.
2. Obtain the set of clauses $\mathscr{C}_{\overline{D}\Phi}$ from $\overline{D}\Phi$.
3. Apply the mvres calculus (rules (R1)-(R7)) to $\mathscr{C}_{\overline{D}\Phi}$ to test for unsatisfiability: if $\mathscr{C}_{\overline{D}\Phi}$ is unsatisfiable, then $u[\phi]$, $u \in D$, is a valid formula in $\mathcal{S}$.

References I

- Bolc, L. & Borowik, P. (1992). *Many-valued logics 1: Theoretical Foundations.* Berlim, etc.: Springer.
- Chang, C.-L. & Lee, R. C.-T. (1973). *Symbolic logic and mechanical theorem proving*. New York & London: Academic Press.
- Leitsch, A. (1997). *The resolution calculus*. Berlin, etc.: Springer.
- Malinowski, G. (1993). *Many-valued logics*. Oxford: Clarendon Press.
- Rescher, N. (1969). *Many-valued logic*. McGraw-Hill.

References II

Baaz, M., Fermüller, C. G., & Salzer, G. (2001). Automated deduction for many-valued logics. In A. Robinson & A. Voronkov (eds.), *Handbook of automated reasoning*, vol. II (p. 1357-1400). Amsterdam: Elsevier / Cambridge, MA: MIT Press.