Non-classical logics

Lecture 7+8: Many-valued logics (Part 3)

27.11.2013

Viorica Sofronie-Stokkermans

sofronie@uni-koblenz.de

Until now

- Many-valued logic (finitely-valued; infinitely-valued)

  History and Motivation

  Syntax

  Semantics

  Functional completeness

  Automated reasoning: Tableaux

Automated reasoning

Classical logic:

Task: prove that F is valid

Method: prove that $\neg F$ is unsatisfiable:

$-$ assume $\neg F$; derive a contradiction.

Many-valued logic:

Task: prove that F is valid

(i.e. $\mathcal{A}(\beta)(F) \in D$ for all $\mathcal{A}, \beta$)

Method: prove that it is not possible that $\mathcal{A}(\beta) \in M \backslash D$:

$-$ assume $F \in M \backslash D$; derive a contradiction.

Problem: How do we express the fact that $F \in M \backslash D$

1) $\bigvee_{v \in M \backslash D} (F = v)$

2) more economical notation?

Automated reasoning

Idea: Use signed formulae

- F^v, where F is a formula and $v \in M$

  $\mathcal{A}, \beta \models F^v$ iff $\mathcal{A}(\beta)(F) = v$.

- $S{:}F$, where F is a formula and

  $\emptyset \neq S \subseteq M$ (set of truth values)

  $\mathcal{A}, \beta \models S{:}F$ iff $\mathcal{A}(\beta)(F) \in S$.

Semantic tableaux

For every $\emptyset \neq S \subseteq M$ and every logical operator f we have a tableau rule:

$$\frac{S{:}f(F_1, \ldots, F_n)}{T(F_1, \ldots, F_n)}$$

where $T(A_1, \ldots, A_n)$ is a finite extended tableau containing only formulae of the form $S_i{:}F_i$.

Informally: Exhaustive list of conditions which ensure that the value of $f(F_1, \ldots, F_n)$ is in S.

Example

Let Ł$_5$ be the 5-valued Łukasiewicz logic with $M = \{0, 1, 2, 3, 4\}$.

$\Rightarrow$	0	1	2	3	4
0	4	4	4	4	4
1	3	4	4	4	4
2	2	3	4	4	4
3	1	2	3	4	4
4	0	1	2	3	4

$$\{4\}(p \Rightarrow q)$$

$\{0\}p$	$\{0, 1\}p$	$\{0, 1, 2\}p$	$\{0, 1, 2, 3\}p$	
	$\{1, 2, 3, 4\}q$	$\{2, 3, 4\}q$	$\{3, 4\}q$	$\{4\}q$

Labelling sets

Let $V \subseteq \mathcal{P}(M)$ be the set of all sets of truth values which are used for labelling the formulae.

Remarks:

1. In general not all subsets of truth values are used, so $V \neq \mathcal{P}(M)$.

2. Proof by contradiction:
   Goal: Prove that F is valid, i.e. $\mathcal{A}(\beta)(F) \in D$.
   We start from $(M \backslash D)$:F and build the tableau
   $\Rightarrow$ We assume that $(M \backslash D) \in V$.

3. Need to make sure that the new signs introduced by the tableau rules are in V.

Tableau rules: Soundness

$$\frac{S{:}f(F_1, \ldots, F_n)}{T(F_1, \ldots, F_n)}$$

where $T(F_1, \ldots, F_n)$ is a finite extended tableau containing only formulae of the form $S_i{:}F_i$.

$$\begin{array}{c}
S{:}f(F_1, \ldots, F_n) \\
\hline
\begin{array}{c|c|c|c}
S_{11}{:}C_{11} & S_{21}{:}C_{21} & \ldots & S_{q1}{:}C_{q1} \\
\ldots & \ldots & & \ldots \\
S_{1k_1}{:}C_{1k_1} & S_{2k_2}{:}C_{2k_2} & & S_{qk'}{:}C_{qk'}
\end{array}
\end{array}$$

where $C_{i,j} \in \{F_1, \ldots, F_n\}$

Tableau rules: Soundness

$$\frac{S{:}f(F_1, \ldots, F_n)}{T(F_1, \ldots, F_n)}$$

where $T(F_1, \ldots, F_n)$ is a finite extended tableau containing only formulae of the form $S_i{:}F_i$.

$$\begin{array}{c}
S{:}f(F_1, \ldots, F_n) \\
\hline
\begin{array}{c|c|c|c}
S_{11}{:}C_{11} & S_{21}{:}C_{21} & \ldots & S_{q1}{:}C_{q1} \\
\ldots & \ldots & & \ldots \\
S_{1k_1}{:}C_{1k_1} & S_{2k_2}{:}C_{2k_2} & & S_{qk'}{:}C_{qk'}
\end{array}
\end{array}$$

where $C_{i,j} \in \{F_1, \ldots, F_n\}$

For every $\mathcal{A}, \beta$: $\mathcal{A}(\beta)(F) \in S$ then there exists i such that for all j: $\mathcal{A}(\beta)(C_{ij}) \in S_{ij}$.

Tableau rules: Soundness

$$
\begin{array}{c|c|c|c}
\multicolumn{4}{c}{S{:}f(F_1, \dots, F_n)} \\
\hline
S_{11}{:}C_{11} & S_{21}{:}C_{21} & \dots & S_{q1}{:}C_{q1} \\
\dots & \dots & & \dots \\
S_{1k_1}{:}C_{1k_1} & S_{2k_2}{:}C_{2k_2} & & S_{qk'}{:}C_{qk'}
\end{array}
$$

where $C_{i,j} \in \{F_1, \dots, F_n\}$

Every model of $S{:}f(F_1, \dots, F_n)$ is also a model of the formulae on one of the branches

If there is no expansion rule for a premise: premise is unsatisfiable ($\mathcal{A}(\beta)(F) \notin S$ for all $\mathcal{A}, \beta$).

If $f(F_1, \dots, F_n)$ satisfiable then there is an expansion rule.

$\mathcal{L}_3$: Tableau rules for $\wedge$

$$\frac{\{1\}A \wedge B}{\begin{array}{c}\{1\}A\\ \{1\}B\end{array}} \qquad \frac{\{u\}A \wedge B}{\begin{array}{c|c|c}\{u\}A & \{u\}B & \{u\}A\\ \{1\}B & \{1\}A & \{u\}B\end{array}} \qquad \frac{\{0\}A \wedge B}{\{0\}A|\{0\}B} \qquad \frac{\{u,0\}A \wedge B}{\{u,0\}A|\{u,0\}B}$$

$\mathcal{L}_3$: Tableau rules for $\vee$

$$\frac{\{1\}A \vee B}{\{1\}A \mid \{1\}B}$$

$$\frac{\{u\}A \vee B}{\begin{array}{c|c} \{u,0\}A & \{u\}A \\ \{u\}B & \{u,0\}B \end{array}}$$

$$\frac{\{0\}A \vee B}{\begin{array}{c} \{0\}A \\ \{0\}B \end{array}}$$

$$\frac{\{u,0\}A \vee B}{\begin{array}{c} \{u,0\}A \\ \{u,0\}B \end{array}}$$

$\mathcal{L}_3$: Tableau rules for $\neg, \sim$

$$\frac{\{1\} \sim A}{\{u,0\}A} \qquad \frac{\{0\} \sim A}{\{1\}A} \qquad \frac{\{u\} \sim A}{} \qquad \frac{\{u,0\} \sim A}{\{1\}A}$$

$$\frac{\{1\}\neg A}{\{0\}A} \qquad \frac{\{0\}\neg A}{\{1\}A} \qquad \frac{\{u\}\neg A}{\{u\}A} \qquad \frac{\{u,0\}\neg A}{\{1\}A|\{u\}A}$$

$\mathcal{L}_3$: Tableau rules for $\supset$

$$\frac{\{1\}A \supset B}{\{u,0\}A \mid \{1\}B} \qquad \frac{\{0\}A \supset B}{\begin{array}{c}\{1\}A\\ \{0\}B\end{array}} \qquad \frac{\{u\}A \supset B}{\begin{array}{c}\{1\}A\\ \{u\}B\end{array}} \qquad \frac{\{u,0\}A \supset B}{\begin{array}{c}\{1\}A\\ \{u,0\}B\end{array}}$$

$\mathcal{L}_3$: Tableau rules for $\exists$

$$\frac{\{1\}\exists x A(x)}{\{1\}A(f(y_1,\ldots,y_k))} \qquad \frac{\{0\}\exists x A(x)}{\{0\}A(z)} \qquad \frac{\{u\}\exists x A(x)}{\begin{array}{c}\{u\}A(f(y_1,\ldots y_k))\\ \{u,0\}A(z)\end{array}} \qquad \frac{\{u,0\}\exists x A(x)}{\{u,0\}A(z)}$$

where

- z is a new free variable
- $y_1,\ldots,y_k$ are the free variables in $\exists x A(x)$
- f is a new function symbol

$\mathcal{L}_3$: Tableau rules for $\forall$

$$\frac{\{1\}\forall x A(x)}{\{1\}A(z)} \qquad \frac{\{0\}\forall x A(x)}{\{0\}A(f(y_1,\ldots,y_k))} \qquad \frac{\{u\}\forall x A(x)}{\begin{array}{c}\{u\}A(f(y_1,\ldots y_k))\\ \{u,1\}A(z)\end{array}} \qquad \frac{\{u,0\}\forall x A(x)}{\{u,0\}A(f(y_1,\ldots,y_k))}$$

where

- z is a new free variable
- $y_1,\ldots,y_k$ are the free variables in $\forall x A(x)$
- f is a new function symbol

Soundness

Theorem (Soundness of the tableau calculus for $\mathcal{L}_3$)
Let F be a $\mathcal{L}_3$-formula without free variables. If there exists a closed tableau T for $\{u, 0\}F$, then F is an $\mathcal{L}_3$-tautology (it is valid).

Proof: Let T be a tableau for F. The following are equivalent:

(1) F is satisfiable

(2) T is satisfiable (i.e. there exists a Σ-structure $\mathcal{A}$ such that for each assignment β there is a path P_β in T such that $\mathcal{A}, \beta \models F$, for each formula F on P_β.

(2) $\Rightarrow$ (1) is obvious.

(1) $\Rightarrow$ (2) can be proved by induction on the structure of the tableau T.

Refutational completeness

Theorem (Refutational completeness)
Let F be a $\mathcal{L}_3$-tautology. Then we can construct a closed tableau for $\{u, 0\}F$. (The order in which we apply the expansion rules is not important).

Proof (Idea): Assume that we cannot construct a closed tableau. If we can construct a finite tableau which is not closed, from the previous result we know that F is clearly satisfiable.

Otherwise, as in the proof for classical logic, we define a fair tableau expansion process which "converges" towards an infinite tableau T. We analyze all non-closed paths of T (on which the "γ"-rules are applied an infinite number of times); we show that for every such path we can order the formula on such path according to a certain ordering and incrementally construct a model for the formulae on that path. This model will then be a model of the formula F.

(The argument can be used for every non-classical logic.)

Resolution

Goal:
Extend the resolution rule such that it takes into account sets of truth values.

Resolution

Classical logic:

Task: prove that F is valid

Method: prove that $\neg F$ is unsatisfiable:

– assume $\neg F$; derive a contradiction.

Many-valued logic:

Task: prove that F is valid

(i.e. $\mathcal{A}(\beta)(F) \in D$ for all $\mathcal{A}, \beta$)

Method: prove that it is not possible that $\mathcal{A}(\beta) \in M \backslash D$:

– assume $F \in M \backslash D$; derive a contradiction.

F^v: abbreviation for $\{v\}{:}F$.

$S{:}F = \bigvee_{v \in S} F^v$.

Resolution

Natural generalization of the resolution rule:

Signed resolution

$$\frac{L_1^{v_1} \vee C \qquad L_2^{v_2} \vee D}{(C \vee D)\sigma}$$

if $v_1 \neq v_2$, and $\sigma = \mathrm{mgu}(L_1, L_2)$

Signed factoring

$$\frac{C \vee L_1^{v} \vee L_2^{v}}{(C \vee L_1^{v})\sigma}$$

if $\sigma = \mathrm{mgu}(L_1, L_2)$

Resolution

Needed:

Method for computing a conjunctive normal form

Example: Classical propositional logic

$F: \quad (P \vee Q) \wedge ((\neg P \wedge Q) \vee R)$

P	Q	R	$(P \vee Q)$	$\neg P$	$(\neg P \wedge Q)$	$((\neg P \wedge Q) \vee R)$	F
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	0	0	0	0
1	1	1	1	0	0	1	1

Example: Classical propositional logic

$F: \quad (P \vee Q) \wedge ((\neg P \wedge Q) \vee R)$

P	Q	R	$(P \vee Q)$	$\neg P$	$(\neg P \wedge Q)$	$((\neg P \wedge Q) \vee R)$	F
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	0	0	0	0
1	1	1	1	0	0	1	1

Example: Classical propositional logic

$F: \quad (P \vee Q) \wedge ((\neg P \wedge Q) \vee R)$

P	Q	R	$(P \vee Q)$	$\neg P$	$(\neg P \wedge Q)$	$((\neg P \wedge Q) \vee R)$	F
0	0	0	0	1	0	0	0
0	0	1	0	1	0	1	0
0	1	0	1	1	1	1	1
0	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0
1	0	1	1	0	0	1	1
1	1	0	1	0	0	0	0
1	1	1	1	0	0	1	1

DNF: $(\neg P \wedge Q \wedge \neg R) \vee (\neg P \wedge Q \wedge R) \vee (P \wedge \neg Q \wedge R) \vee (P \wedge Q \wedge R)$

Example: Classical propositional logic

$F: \quad (P \vee Q) \wedge ((\neg P \wedge Q) \vee R)$

P	Q	R	$(P \vee Q)$	$\neg P$	$(\neg P \wedge Q)$	$((\neg P \wedge Q) \vee R)$	F	$\neg F$
0	0	0	0	1	0	0	0	1
0	0	1	0	1	0	1	0	1
0	1	0	1	1	1	1	1	0
0	1	1	1	1	1	1	1	0
1	0	0	1	0	0	0	0	1
1	0	1	1	0	0	1	1	0
1	1	0	1	0	0	0	0	1
1	1	1	1	0	0	1	1	0

CNF: (1) DNF of $\neg F$:

$$(\neg P \wedge \neg Q \wedge \neg R) \vee (\neg P \wedge \neg Q \wedge R) \vee (P \wedge \neg Q \wedge \neg R) \vee (P \wedge Q \wedge \neg R)$$

(2) negate:

$$(P \vee Q \vee R) \wedge (P \vee Q \vee \neg R) \wedge (\neg P \vee Q \vee R) \wedge (\neg P \vee \neg Q \vee R)$$

Signed resolution: Propositional logic

Translation to signed clause form.

$\Psi = S{:}f(F_1, \ldots, F_n)$

$$DNF(\Psi) := \bigvee_{\substack{v_1,\ldots,v_n \in M \\ f_M(v_1,\ldots,v_n) \in S}} F_1^{v_1} \wedge \cdots \wedge F_n^{v_n}$$

$$CNF(\Psi) := \bigwedge_{\substack{v_1,\ldots,v_n \in M \\ f_M(v_1,\ldots,v_n) \notin S}} (M \setminus \{v_1\}){:}F_1 \vee \cdots \vee (M \setminus \{v_n\}){:}F_n$$

(negate $DNF(M \setminus S{:}f(F_1, \ldots, F_n))$)

Example

$\Rightarrow$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	1	1
1	0	$\frac{1}{2}$	1

Compute CNF for $\{0\}{:}(F_1 \rightarrow F_2)$:

DNF for $\{\frac{1}{2}, 1\}{:}(F_1 \rightarrow F_2)$: $\displaystyle\bigvee_{\substack{v_1, v_2 \in \{0, \frac{1}{2}, 1\} \\ v_1 \Rightarrow v_2 \neq 0}} \{v_1\}{:}F_1 \wedge \{v_2\}{:}F_2$

$$
\begin{array}{ccccc}
(F_1^0 \wedge F_2^0) & \vee & (F_1^0 \wedge F_2^{\frac{1}{2}}) & \vee & (F_1^0 \wedge F_2^1) \\
(F_1^{\frac{1}{2}} \wedge F_2^0) & \vee & (F_1^{\frac{1}{2}} \wedge F_2^{\frac{1}{2}}) & \vee & (F_1^{\frac{1}{2}} \wedge F_2^1) \\
(F_1^1 \wedge F_2^{\frac{1}{2}}) & \vee & (F_1^1 \wedge F_2^1) & &
\end{array}
$$

CNF for $\{0\}{:}(F_1 \rightarrow F_2)$:

$$
\begin{array}{ccccc}
(\{\frac{1}{2}, 1\}{:}F_1 \vee \{\frac{1}{2}, 1\}{:}F_2) & \wedge & (\{\frac{1}{2}, 1\}{:}F_1 \vee \{0, 1\}{:}F_2) & \wedge & (\{\frac{1}{2}, 1\}{:}F_1 \vee \{0, \frac{1}{2}\}{:}F_2) \\
(\{0, 1\}{:}F_1 \vee \{\frac{1}{2}, 1\}{:}F_2) & \wedge & (\{0, 1\}{:}F_1 \vee \{0, 1\}{:}F_2) & \wedge & (\{0, 1\}{:}F_1 \vee \{0, \frac{1}{2}\}{:}F_2) \\
(\{0, \frac{1}{2}\}{:}F_1 \vee \{0, 1\}{:}F_2) & \wedge & (\{0, \frac{1}{2}\}{:}F_1^1 \vee \{0, \frac{1}{2}\}{:}F_2^1) & &
\end{array}
$$

Example

$\Rightarrow$	0	$\frac{1}{2}$	1
0	1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	1	1
1	0	$\frac{1}{2}$	1

Compute CNF for $\{0\}{:}(F_1 \rightarrow F_2)$:

DNF for $\{\frac{1}{2},1\}{:}(F_1 \rightarrow F_2)$: $\displaystyle\bigvee_{\substack{v_1,v_2 \in \{0,\frac{1}{2},1\} \\ v_1 \Rightarrow v_2 \neq 0}} \{v_1\}{:}F_1 \wedge \{v_2\}{:}F_2$

$$= \quad (F_1^0 \wedge F_2^{\{0,\frac{1}{2},1\}}) \vee (F_1^{\frac{1}{2}} \wedge F_2^{\{0,\frac{1}{2},1\}}) \vee (F_1^1 \wedge F_2^{\{\frac{1}{2},1\}})$$

$$= \quad F_1^0 \vee F_1^{\frac{1}{2}} \vee (F_1^1 \wedge F_2^{\{\frac{1}{2},1\}})$$

CNF for $\{0\}{:}(F_1 \rightarrow F_2)$:

$$\{\tfrac{1}{2},1\}{:}F_1 \ \wedge \ \{0,1\}{:}F_1 \ \wedge \ (\{0,\tfrac{1}{2}\}{:}F_1 \vee \{0\}{:}F_2)$$

Optimization

$$\Psi = S{:}f(F_1, \ldots, F_n)$$

$$DNF(\Psi) := \bigvee_{v_1,\ldots,v_{n-1}\in M} \{v_1\}{:}F_1 \wedge \cdots \wedge \{v_{n-1}\}{:}F_{n-1} \wedge \{v_n \mid f_M(v_1, \ldots, v_n) \in S\}{:}F_n$$

$$CNF(\Psi) := \bigwedge_{v_1,\ldots,v_{n-1}\in M} (M\backslash\{v_1\}){:}F_1 \vee \cdots \vee (M\backslash\{v_{n-1}\}){:}F_{n-1} \vee \{v_n \mid f_M(v_1, \ldots, v_n)\in S\}{:}F_n$$

(negate $DNF(M\backslash S{:}f(F_1, \ldots, F_n)))$)

Soundness

Signed resolution (propositional form)

$$\frac{P^{v_1} \vee C \qquad P^{v_2} \vee D}{C \vee D}$$

if $v_1 \neq v_2$

Signed factoring (propositional form)

$$\frac{C \vee P^{v} \vee P^{v}}{C \vee P^{v}}$$

Soundness

Theorem. The signed resolution inference rule is sound.

Proof (propositional case)
Let $\mathcal{A}$ be a valuation such that $\mathcal{A} \models P^{v_1} \vee C$ and $\mathcal{A} \models P^{v_2} \vee D$.

Case 1: $\mathcal{A} \models P^{v_1}$. Then $\mathcal{A}(P) = v_1$, hence $\mathcal{A}(P) \neq v_2$. Therefore, $\mathcal{A} \models D$.
Hence, $\mathcal{A} \models C \vee D$.

Case 2: $\mathcal{A} \not\models P^{v_1}$. Then $\mathcal{A} \models C$.
Hence also in this case $\mathcal{A} \models C \vee D$.

Soundness of signed factoring is obvious.

Completeness: Propositional Logic

Encoding into first-order logic with equality

Signed resolution

$$\frac{P \approx v_1 \vee C \qquad P \approx v_2 \vee D}{(C \vee D)} \qquad \text{if } v_1 \neq v_2$$

Signed factoring

$$\frac{C \vee P \approx v \vee P \approx v}{C}$$

Idea: Signed resolution can be simulated by a version of resolution which handles equality efficiently (superposition). Completeness then follows from the completeness of this refinement of resolution.
This also guarantees completeness of refinements of signed resolution with ordering and selection functions

Compact form of signed resolution

Propositional logic

Signs: sets of truth values

Resolution

$$\frac{S_1{:}P \vee C \qquad S_2{:}P \vee D}{(S_1 \cap S_2){:}P \vee C \vee D} \qquad \text{if } S_1 \cap S_2 = \emptyset$$

Simplificaton

$$\frac{C \vee \emptyset{:}P}{C}$$

Merging

$$\frac{S_1{:}P \vee S_2{:}P \vee C}{(S_1 \cup S_2){:}P \vee C}$$

First-order logic

Translation to clause form:

need to take into account also the truth tables of the quantifiers.

$S : QxF(x)$

DNF: $\bigvee_{\substack{\emptyset \neq V \subseteq M \\ Q_M(V) \in S}} (\forall x \; V : F(x) \wedge \bigwedge_{a \in V} \exists x \; \{a\} : F(x))$

CNF: computed by negating the DNF for $M \backslash S : \forall x F(x)$

CNF: $\bigwedge_{\substack{\emptyset \neq V \subseteq M \\ Q_M(V) \in (M \backslash S)}} (\exists x (M \backslash V) : F(x) \vee \bigvee_{a \in V} \forall x (M \backslash \{a\}) : F(x))$

$\mapsto$ leave out quantifiers (Skolem functions for existential quantifier)

Example

In $\mathcal{L}_3$, with truth values $M = \{0, u, 1\}$:

$\{1, u\}\forall x\ p(x)$

$$\Rightarrow \quad \bigwedge_{\substack{\emptyset \neq V \subseteq M \\ min(V) \in \{0\}}} (\exists x(M\setminus V) : F(x) \vee \bigvee_{a\in\{0\}} \forall x(M\setminus\{a\}) : F(x))$$

$$\Rightarrow \quad (\exists x\{1, u\}{:}p(x) \vee \forall x(M\setminus\{0\}){:}p(x))\wedge \qquad V = \{0\}$$

$$(\exists x\{u\}{:}p(x) \vee \forall x\{1, u\}{:}p(x) \vee \forall x\{0, u\}{:}p(x))\wedge \qquad V = \{0, 1\}$$

$$(\exists x\{1\}{:}p(x) \vee \forall x\{1, u\}{:}p(x) \vee \forall x\{0, 1\}{:}p(x))\wedge \qquad V = \{0, u\}$$

$$\forall x\{1, u\}{:}p(x) \vee \forall x\{0, 1\} : p(x) \vee \forall x\{0, u\} : p(x)) \qquad V = M$$

Structure-preserving translation

In order to avoid rapid growth of the number of clauses, a structure-preserving translation to clause form is used.

Idea

$$S : F[G(x)] \quad \Rightarrow \quad S : F[P_{G(x)}(x)] \wedge \bigwedge_{a \in M} \forall x(\{a\} G(x) \leftrightarrow \{a\} : P_{G(x)}(x))$$

where $P_{G(x)}$ new predicate symbol.

$$S{:}F[\underbrace{f(F_1, \dots, F_n)}_{G}]$$

$$\quad \Rightarrow \quad S : F[P_G] \wedge \bigwedge_{a \in M} \forall x(DNF(\{a\} : f(F_1, \dots, F_n) \leftrightarrow \{a\} : P_G)$$

Resolution for first-order clauses

Natural generalization of the resolution rule:

Signed resolution

$$\frac{L_1^{v_1} \vee C \qquad L_2^{v_2} \vee D}{(C \vee D)\sigma}$$

if $v_1 \neq v_2$, and $\sigma = \text{mgu}(L_1, L_2)$

Signed factoring

$$\frac{C \vee L_1^{v} \vee L_2^{v}}{(C \vee L_1^{v})\sigma}$$

if $\sigma = \text{mgu}(L_1, L_2)$

Regular logics

Many-valued logics for which an order $\leq$ exists on the sets of truth values and for which signed CNF's can be found which contain as signs only the sets

$$\uparrow i = \{j \in M \mid j \geq i\} \text{ and } \downarrow i = \{j \in M \mid j \leq i\}$$

Regular logics

Many-valued logics for which an order $\leq$ exists on the sets of truth values and for which signed CNF's can be found which contain as signs only the sets

$$\uparrow i = \{j \in M \mid j \geq i\} \text{ and } \downarrow i = \{j \in M \mid j \leq i\}$$

Example

Łukasiewicz logics $\mathcal{L}_n$

- Set of truth values $M = \{0, \frac{1}{n-1}, \frac{2}{n-1}, \ldots, 1\}$
- Logical operations: $\vee, \wedge, \neg, \Rightarrow$
  - $\vee_{Ł_n} = \max$
  - $\wedge_{Ł_n} = \min$
  - $\neg_{Ł_n} x = 1 - x$
  - $x \Rightarrow_{Ł_n} y = \min(1, 1 - x + y)$
- First-order version: $\mathcal{Q} = \{\forall, \exists\}$

Łukasiewicz logics

Łukasiewicz implication $x \Rightarrow_{Ł_n} y = \min(1, 1 - x + y)$

$\mathcal{L}_n$

$\Rightarrow$	0	$\frac{1}{n-1}$	$\frac{2}{n-1}$	$\dots$	$\frac{n-2}{n-1}$	1
0	1	1	1	$\dots$	1	1
$\frac{1}{n-1}$	$\frac{n-2}{n-1}$	1	1	$\dots$	1	1
$\frac{2}{n-1}$	$\frac{n-3}{n-1}$	$\frac{n-2}{n-1}$	1	$\dots$	1	1
$\dots$						
1	0	$\frac{1}{n-1}$	$\frac{2}{n-1}$	$\dots$	$\frac{n-2}{n-1}$	1

Example

$\uparrow i : (F_1 \wedge F_2) \mapsto (\uparrow i : F_1) \wedge (\uparrow i : F_2)$

$\uparrow i : (F_1 \vee F_2) \mapsto (\uparrow i : F_1) \vee (\uparrow i : F_2)$

$\uparrow i : \neg F \quad\quad \mapsto \downarrow (1 - i) : F$

$\uparrow i : F_1 \Rightarrow F_2 \quad\quad \mapsto \bigvee_{j \in M} (\downarrow j : F_1 \wedge \uparrow (i + j - 1) : F_2$

Similar for $\downarrow i : F$

signed CNFs can be obtained using the transformation rules above (and possibly negation).

Applications of many-valued logic

- independence proofs
- modeling undefined function and predicate values (program verification)
- semantic of natural languages
- theory of logic programming: declarative description of operational semantics of negation
- modeling of electronic circuits
- modeling vagueness and uncertainly
- shape analysis (program verification)

Applications of many-valued logic

- independence proofs
- modeling undefined function and predicate values (program verification)
- semantic of natural languages
- theory of logic programming: declarative description of operational semantics of negation
- modeling of electronic circuits
- modeling vagueness and uncertainly
- shape analysis (program verification)

Independence proofs

Task: Check independence of axioms in axiom systems [Bernays 1926]

Here: Example: Axiom system for propositional logic K_1

Ax1 $p_1 \Rightarrow (p_2 \Rightarrow p_1)$

Ax2 $((p_1 \Rightarrow p_2) \Rightarrow p_1) \Rightarrow p_1$

Ax3 $(p_1 \Rightarrow p_2) \Rightarrow ((p_2 \Rightarrow p_3) \Rightarrow (p_1 \Rightarrow p_3))$

Ax4 $(p_1 \wedge p_2) \Rightarrow p_1$

Ax5 $(p_1 \wedge p_2) \Rightarrow p_2$

Ax6 $(p_1 \Rightarrow p_2) \Rightarrow ((p_1 \Rightarrow p_3) \Rightarrow p_1 \Rightarrow p_2 \wedge p_3))$

Ax7 $p_1 \Rightarrow (p_1 \vee p_2)$

Ax8 $p_2 \Rightarrow (p_1 \vee p_2)$

Axiom system: K_1

Ax9 $(p_1 \Rightarrow p_3) \Rightarrow ((p_2 \Rightarrow p_3) \Rightarrow p_1 \vee p_2 \Rightarrow p_3))$

Ax10 $(p_1 \approx p_2) \Rightarrow (p_1 \Rightarrow p_2)$

Ax11 $(p_1 \approx p_2) \Rightarrow (p_2 \Rightarrow p_1)$

Ax12 $(p_1 \Rightarrow p_2) \Rightarrow ((p_2 \Rightarrow p_1) \Rightarrow p_1 \approx p_2))$

Ax13 $(p_1 \Rightarrow p_2) \Rightarrow (\neg p_2 \Rightarrow \neg p_1)$

Ax14 $p_1 \Rightarrow \neg\neg p_1$

Ax15 $\neg\neg p_1 \Rightarrow p_1$

Inference rule: Modus Ponens: $\dfrac{H \quad H \Rightarrow G}{G}$

Independence

Definition: An axiom system K is independent iff for every axiom $A \in K$, A is not provable from $K \backslash \{A\}$.

We will show that Ax2 is independent

Independence

Definition: An axiom system K is independent iff for every axiom $A \in K$, A is not provable from $K \backslash \{A\}$.

We will show that Ax2 is independent

Idea: We introduce a 3-valued logic L_{K_1} with truth values $\{0, u, 1\}$, $D = \{1\}$ and operations $\neg, \Rightarrow, \wedge, \vee, \approx$ as defined in the lecture.

To show:

1. Every axiom in K_1 except for $Ax2$ is a L_{K_1}-tautology.
2. Modus Ponens leads from L_{K_1} tautologies to a L_{K_1}-tautology.
3. $Ax2$ is not a L_{K_1}-tautology.

Independence

From 1,2,3 it follows that every formula which can be proved from $K_1 \backslash Ax2$ is a tautology.

Hence – since $Ax2$ is not a tautology – $K_1 \backslash \{Ax2\} \not\models Ax2$.

Proof

We introduce a 3-valued logic L_{K_1} with truth values $\{0, u, 1\}$, $D = \{1\}$ and operations $\neg, \Rightarrow, \wedge, \vee, \approx$ as defined in the lecture.

To show:

1. Every axiom in K_1 except for $Ax2$ is a L_{K_1}-tautology.
2. Modus Ponens leads from L_{K_1} tautologies to a L_{K_1}-tautology.
3. $Ax2$ is not a L_{K_1}-tautology.

Proof

We introduce a 3-valued logic L_{K_1} with truth values $\{0, u, 1\}$, $D = \{1\}$ and operations $\neg, \Rightarrow, \wedge, \vee, \approx$ as defined in the lecture.

To show:

1. Every axiom in K_1 except for $Ax2$ is a L_{K_1}-tautology.
2. Modus Ponens leads from L_{K_1} tautologies to a L_{K_1}-tautology.
3. $Ax2$ is not a L_{K_1}-tautology.

1. Routine (check all axioms in $K_1 \backslash \{Ax2\}$).

Proof

We introduce a 3-valued logic L_{K_1} with truth values $\{0, u, 1\}$, $D = \{1\}$ and operations $\neg, \Rightarrow, \wedge, \vee, \approx$ as defined in the lecture.

To show:

1. Every axiom in K_1 except for $Ax2$ is a L_{K_1}-tautology.
2. Modus Ponens leads from L_{K_1} tautologies to a L_{K_1}-tautology.
3. $Ax2$ is not a L_{K_1}-tautology.

2. Analyze the truth table of $\Rightarrow$.

Assume H is a tautology and $H \Rightarrow G$ is a tautology.

Let $\mathcal{A} : \Pi \to \{0, u, 1\}$.

Then $\mathcal{A}(H) = 1$ and $\mathcal{A}(H \Rightarrow G) = 1$, so $\mathcal{A}(G) = 1$.

Proof

We introduce a 3-valued logic L_{K_1} with truth values $\{0, u, 1\}$, $D = \{1\}$ and operations $\neg, \Rightarrow, \wedge, \vee, \approx$ as defined in the lecture.

To show:

1. Every axiom in K_1 except for $Ax2$ is a L_{K_1}-tautology.
2. Modus Ponens leads from L_{K_1} tautologies to a L_{K_1}-tautology.
3. $Ax2$ is not a L_{K_1}-tautology.

3. Let $\mathcal{A} : \Pi \to \{0, u, 1\}$ with $\mathcal{A}(p_1) = u$ and $\mathcal{A}(p_2) = 0$.

Then

$$\begin{aligned} \mathcal{A}(((p_1 \Rightarrow p_2) \Rightarrow p_1) \Rightarrow p_1) &= ((u \Rightarrow 0) \Rightarrow u) \Rightarrow u \\ &= (u \Rightarrow u) \Rightarrow u = u. \end{aligned}$$