Proof-search and countermodel generation for non-normal modal logics

Gian Luca POZZATO

Progetti GNCS 2019, Hotel Belvedere, Montecatini Terme

13 febbraio 2020
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   Basic System E
   Extensions
3. Sequent calculi LSE*
4. PRONOM
   Finding a derivation
   Counter-model construction
5. Hypersequent Calculi $\mathcal{H}_E^*$
6. HYPNO
   Finding a derivation
   Counter-model construction
7. Conclusions
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   - Basic System E
   - Extensions
3. Sequent calculi LSE*
4. PRONOM
   - Finding a derivation
   - Counter-model construction
5. Hypersequent Calculi $\mathcal{H}_E^*$
6. HYPNO
   - Finding a derivation
   - Counter-model construction
7. Conclusions
Non-normal modal logics

Generalization of ordinary modal logics that do not satisfy some axioms or rules of normal modal logic $K$

**Why NNML**

Avoid the commitment to some logic principles that are theorems of $K$ but are not acceptable under possible interpretations of the modality:

- Deontic
- Epistemic
- …
Non-normal modal logics

• Epistemic reasoning
  • $\square A$ read as “the agent knows/believes $A$”
  • NNML offer a partial solution to the problem of omniscience
    • $\square B$ does not follow from $\square A$ and $\square A \rightarrow \square B$
    • rejection of the rule of monotonicity: $A \rightarrow B$ implies $\square A \rightarrow \square B$

• Deontic logic
  • $\square A$ interpreted as “it is obligatory that $A$”
  • NNML offers solutions to some paradoxes
    • Ross’ paradox
    • gentle-murder paradox
Non-normal modal logics

Systems considered

\[ E := \text{CPL} + \]

- **Extensions:**
  - **M** \[\blacklozenge(A \land B) \rightarrow \blacklozenge A\]
  - **C** \[\blacklozenge A \land \blacklozenge B \rightarrow \blacklozenge(A \land B)\]
  - **N** \[\blacklozenge \top\]

\[
\text{RE} \quad \frac{A \rightarrow B}{\blacklozenge A \rightarrow \blacklozenge B} \quad \frac{B \rightarrow A}{\blacklozenge A \rightarrow \blacklozenge B}
\]
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   Basic System $\text{E}$
   Extensions
3. Sequent calculi $\text{LSE}^*$
4. PRONOM
   Finding a derivation
   Counter-model construction
5. Hypersequent Calculi $\mathcal{H}_E^*$
6. HYPNO
   Finding a derivation
   Counter-model construction
7. Conclusions
Bi-neighbourhood semantics

Bi-neighbourhood models

Triples $\mathcal{M} = \langle W, N, V \rangle$

- $W$ is a non-empty set of worlds (states)
- $V$ is a valuation function
- $N$ is a bi-neighbourhood function $W \rightarrow \mathcal{P}(\mathcal{P}(W) \times \mathcal{P}(W))$ such that $(\alpha, \beta) \in N(w)$ implies $\alpha \cap \beta = \emptyset$

Intuition: $\alpha$ and $\beta$ provide independent positive and negative evidence for $A$

Truth of $\Box A$

$w \in [\Box A]$ iff there is $(\alpha, \beta) \in N(w)$ such that for all $u \in \alpha$, $u \in [A]$, and for all $v \in \beta$, $v \not\in [A]$
Model properties

C-models: $(\alpha_1, \beta_1), (\alpha_2, \beta_2) \in \mathcal{N}(w)$ implies $(\alpha_1 \cap \alpha_2, \beta_1 \cup \beta_2) \in \mathcal{N}(w)$

N-models: $(W, \emptyset) \in \mathcal{N}(w)$ for all $w \in W$

M-models: $(\alpha, \beta) \in \mathcal{N}(w)$ implies $\beta = \emptyset$

Completeness

Logic $E(M, C, N)$ is sound and complete w.r.t. bi-neighbourhood $(M, C, N)$-models
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   Basic System $\mathbf{E}$
   Extensions
3. Sequent calculi LSE$^*$
4. PRONOM
   Finding a derivation
   Counter-model construction
5. Hypersequent Calculi $\mathcal{H}_E^*$
6. HYPNO
   Finding a derivation
   Counter-model construction
7. Conclusions
Desiderata

- Analytic
- Standard
  - Finite numbers of rules each one with finite and fixed number of premises
- Modular
  - Fixed set of basic rules
  - Extensions obtained by adding suitable rules expressing semantic properties
- Terminating
- Optimal
- Countermodel generation
  - Obtain directly a countermodel from a failed proof search
The Sequent Calculi LSE* 

Labels
- World labels $x, y, z, \ldots$
- Neighbourhood labels $a, b, c, \ldots$

Neighbourhood terms
- Positive terms $[a_1, \ldots, a_n] = \text{finite sets of neighbourhood labels}$
- Negative terms $\overline{t}$, where $t$ is a positive term
- Neighbourhood constants $\tau, \overline{\tau}$

Intuition
- $t$ and $\overline{t}$ are the two members of a pair of neighbourhoods
- $[a_1, \ldots, a_n] \leadsto a_1 \cap \ldots \cap a_n$
- $[a_1, \ldots, a_n] \leadsto \overline{a_1} \cup \ldots \cup \overline{a_n}$
- $\tau \leadsto W$
The Sequent Calculi LSE*

**Initial sequents:**

- \( x : p, \Gamma \Rightarrow \Delta, x : p \)
- \( x : \bot, \Gamma \Rightarrow \Delta \)
- \( \Gamma \Rightarrow \Delta, x : \top \)

**L \text{ L}**

- \( x : \Gamma \Rightarrow \Delta, t \models^\forall A \)
- \( x : \Gamma \Rightarrow \Delta, x : A \)
- \( \Gamma \Rightarrow \Delta, t \models^\forall A \)

**L \text{ L}***

- \( x : \Gamma \Rightarrow \Delta, t \models^\exists A \)
- \( x : \Gamma \Rightarrow \Delta, x : A, t \models^\exists A \)

**L \text{ L}**

- \( [a] \in \mathcal{N}(x), [a] \models^\forall A, \Gamma \Rightarrow \Delta, [a] \models^\exists A \)
- \( x : \square A, \Gamma \Rightarrow \Delta \)

**R \text{ R}**

- \( x : \Gamma \Rightarrow \Delta, x : \square A \)
- \( x : \Gamma \Rightarrow \Delta, t \models^\exists A \)

**M**

- \( t \in \mathcal{N}(x), x : \square A \)
- \( t \in \mathcal{N}(x), x : \square A \)

**N\tau**

- \( \tau \in \mathcal{N}(x), \Gamma \Rightarrow \Delta \)

**C**

- \( [a_1, \ldots, a_n] \in \mathcal{N}(x), [a_1] \in \mathcal{N}(x), \ldots, [a_n] \in \mathcal{N}(x), \Gamma \Rightarrow \Delta \)
- \( [a_1] \in \mathcal{N}(x), \ldots, [a_n] \in \mathcal{N}(x), \Gamma \Rightarrow \Delta \)

**dec**

- \( x \in [a_1, \ldots, a_n], \Gamma \Rightarrow \Delta \)

**dec**

- \( x \in [a_1, \ldots, a_n], \Gamma \Rightarrow \Delta \)

Application conditions:

- \( x \) is fresh in \( \text{L} \models^\forall \) and \( \text{R} \models^\exists \), and \( x \) occurs in the conclusion of \( N\tau \).
Examples of derivations

Axiom M

\[
\begin{align*}
L^\land & \quad \ldots, y : A, y : B, y : \lceil a \rceil, \lceil a \rceil : A \land B \Rightarrow y : A, \ldots \\
\vdash L & \quad \ldots, y : A \land B, y : \lceil a \rceil, \lceil a \rceil : A \land B \Rightarrow y : A, \ldots \\
R^\forall & \quad \ldots, y : \lceil a \rceil, \lceil a \rceil : A \land B \Rightarrow y : A, \ldots \\
\vdash R & \quad \ldots, [a] : A \land B \Rightarrow [a] : A, \ldots \\
M & \quad \ldots, y : [a], y : A, [a] : x \Rightarrow \ldots \\
\vdash L \exists & \quad \ldots, [a] : A, [a] : x \Rightarrow \ldots \\
R^\Box & \quad [a] : x, [a] : A \land B \Rightarrow x : \Box A, [a] : A \land B \\
\vdash L^\Box & \quad x : \Box (A \land B) \Rightarrow x : \Box A
\end{align*}
\]
Example of failed proof search and countermodel

**Axiom M in LSE**

\[
\begin{array}{c}
B \quad \text{saturated branch } B \\
A' \text{ cl.} \\
y : [a], y : p, [a] : x, [a] : p \land q \Rightarrow x : \Box p, [a] : p \land q, y : q \\
y : [a], y : p, [a] : x, [a] : p \land q \Rightarrow x : \Box p, [a] : p \land q, y : p \land q \\
y : [a], y : p, [a] : x, [a] : p \land q \Rightarrow x : \Box p, [a] : p \land q \\
\end{array}
\]

\[
\begin{array}{c}
A \text{ cl.} \\
[a] : x, [a] : p \land q, [a] : p \Rightarrow x : \Box p, [a] : p \land q \\
\end{array}
\]

\[
\begin{array}{c}
x : \Box(p \land q) \Rightarrow x : \Box p \\
\end{array}
\]

**Bi-neighbourhood countermodel \( \mathcal{M} \)**

\[
W = \{x, y\} \\
\mathcal{N}(x) = \{\langle \alpha[a], \alpha[a]\rangle\} \\
\mathcal{N}(y) = \emptyset \\
\alpha[a] = \emptyset \\
\mathcal{V}(p) = \{y\} \\
\alpha[a] = \{y\} \\
\mathcal{V}(q) = \emptyset
\]

We have \( \mathcal{M}, x \models \Box(p \land q) \) and \( \mathcal{M}, x \not\models \Box p \)
Results

Non-Normal Modal Logics
Bi-neighbourhood semantics
Sequent calculi LSE*
PRONOM
Hypersequent Calculi $\mathcal{H}_E^*$
HYPNO
Conclusions

AXIOMATIC SYSTEMS $E^*$

SEMANTICS
Bi-neighbourhood models

PROOF SYSTEMS
Labelled calculi

Decision procedures
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   - Basic System E
   - Extensions
3. Sequent calculi LSE*
4. PRONOM
   - Finding a derivation
   - Counter-model construction
5. Hypersequent Calculi $\mathcal{H}_E^*$
6. HYPNO
   - Finding a derivation
   - Counter-model construction
7. Conclusions
Basic ideas

- inspired by the “lean” methodology of lean $\mathcal{TAP}$
- set of clauses, each one implementing a sequent rule or an axiom of LSE
- proof search provided for free by DFS of Prolog
- predicate
  \[
  \text{terminating\_proof\_search}(\Gamma,\Delta,\text{Derivation}).
  \]
- when it fails PRONOM executes the predicate
  \[
  \text{build\_saturate\_branch}(\Gamma,\Delta,\text{Model}).
  \]
Clause for axiom

```
terminating_proof_search(Neigh, Gamma, Delta, tree(axiom), __, __, __):-
    member([X,A], Gamma),
    member([X,A], Delta), !.
```
Clause for $R□$

```prolog
terminating_proof_search(Neigh, Gamma, Delta,
  tree(rbox, LeftTree, RightTree), RBox, RExist, LAll):-
  member([X, box A], Delta),
  member([X, SpOfX], Neigh),
  member(T, SpOfX),
  \+member([X, A, T], RBox),
  !,
  terminating_proof_search(Neigh, Gamma, [[forall, T, 0, A]|Delta],
    LeftTree, [[X, A, T]|RBox], RExist, LAll),
  terminating_proof_search(Neigh, [[exists, T, 1, A]|Gamma], Delta,
    RightTree, [[X, A, T]|RBox], RExist, LAll).
```
Counter-model

Basic ideas

- predicate build_saturate_branch
- build an open, saturated branch
- clauses are the same as the predicate terminating_proof_search, however rules introducing a branch in a backward proof search are implemented by pairs of (disjoint) clauses, each one representing an attempt to build an open saturated branch
Clauses for $R\square$

\[
\text{build}_\text{satrate}_\text{branch}(\text{Neigh}, \text{Gamma}, \text{Delta}, \text{Model}, \text{RBox}, \text{RExist}, \text{LAll}) :- \\
\quad \text{member}([X, \text{box } A], \text{Delta}), \\
\quad \text{member}([X, \text{SpOfX}], \text{Neigh}), \\
\quad \text{member}(T, \text{SpOfX}), \\
\quad \langle + \rangle \text{member}([X, A, T], \text{RBox}), \\
\quad \text{build}_\text{satrate}_\text{branch}(\text{Neigh}, \text{Gamma}, [[\forall [T, 0, A]] | \text{Delta}], \text{Model}, \\
\quad \quad \quad [[X, A, T] | \text{RBox}], \text{RExist}, \text{LAll}).
\]

\[
\text{build}_\text{satrate}_\text{branch}(\text{Neigh}, \text{Gamma}, \text{Delta}, \text{Model}, \text{RBox}, \text{RExist}, \text{LAll}) :- \\
\quad \text{member}([X, \text{box } A], \text{Delta}), \\
\quad \text{member}([X, \text{SpOfX}], \text{Neigh}), \\
\quad \text{member}(T, \text{SpOfX}), \\
\quad \langle + \rangle \text{member}([X, A, T], \text{RBox}), \\
\quad \text{build}_\text{satrate}_\text{branch}(\text{Neigh}, [[\exists [T, 1, A]] | \text{Gamma}], \text{Delta}, \text{Model}, \\
\quad \quad \quad [[X, A, T] | \text{RBox}], \text{RExist}, \text{LAll}).
\]
PRONOM A THEOREM PROVER FOR NONNORMAL MODAL LOGICS

BY TIZIANO DALMONT, SARA NEGR, NICOLA OLIVETTI, AND GIAN LUCA POZZATO, WITH THE VALUABLE HELP OF CYRIL TERRIOUX

A Prolog implementation of labelled sequent calculi for logic E and standard extensions with M, N, and C

NONNORMAL MODAL LOGIC:  ♦ E  ○ EM  ○ EM (OPT)  ○ EN  ○ EC  ○ EMN  ○ EMC  ○ ENC  ○ EMNC

Type here your modal formula

run PRONOM

DO NOT USE CAPITAL LETTERS FOR PROPOSITIONAL VARIABLES.
Use: false for bottom, true for top, ∧ for conjunction, ? for disjunction, ¬ for negation, -> for material implication, box for modality □.

EXAMPLES

VALID FORMULAS

(□ (a ∧ (b ? c))) -> (□ ((a ∧ b) ? (a ∧ c)))
(□ a ∧ □ b) -> (□ (a ∧ b))
□ (□ a) -> □ □ a

http://193.51.60.97:8000/pronom/
Basic ideas

- internal calculi
- direct interpretation of a sequent into a NNML formula
- easier to describe a decision procedure
- block $\langle \Sigma \rangle$, where $\Sigma$ is a multiset of formulas of $\mathcal{L}$
- sequent: pair $\Gamma \Rightarrow \Delta$, where $\Gamma$ is a multiset of formulas and blocks, and $\Delta$ is a multiset of formulas
Basic ideas

- **hypersequent** $S_1 \mid \ldots \mid S_n$, where $S_1, \ldots, S_n$ are sequents
- single sequents interpreted as:

  $i(A_1, \ldots, A_n, \langle \Sigma_1 \rangle, \ldots, \langle \Sigma_m \rangle \Rightarrow B_1, \ldots, B_k) =$

  $$= \bigwedge_{i \leq n} A_i \land \bigwedge_{j \leq m} \Box \bigwedge \Sigma_j \rightarrow_{\ell \leq k} B_\ell$$

- validity of a sequent $S$ in a bi-neighbourhood model $\mathcal{M}$
  $\mathcal{M} \models S$: for all $w \in \mathcal{M}$, $\mathcal{M}, w \not\models i(S)$
- if $\mathcal{M} \models S$ for some $S \in H$, $H$ is valid in $\mathcal{M}$
- $H$ is valid if it is valid in all models of that kind
Hypersequent Calculi $\mathcal{H}_{E^*}$

**Init**

$\frac{G \mid p, \Gamma \Rightarrow \Delta, p}{G \mid p, \Gamma \Rightarrow \Delta}$

**$\bot_L$**

$\frac{G \mid \bot, \Gamma \Rightarrow \Delta}{G \mid \bot, \Gamma \Rightarrow \Delta}$

**$\top_R$**

$\frac{G \mid \Gamma \Rightarrow \Delta, \top}{G \mid \Gamma \Rightarrow \Delta, \top}$

**$\rightarrow_L$**

$\frac{G \mid A \rightarrow B, \Gamma \Rightarrow \Delta, A}{G \mid A \rightarrow B, \Gamma \Rightarrow \Delta}$

$\frac{G \mid B, A \rightarrow B, \Gamma \Rightarrow \Delta}{G \mid B, A \rightarrow B, \Gamma \Rightarrow \Delta}$

**$\rightarrow_R$**

$\frac{G \mid A, \Gamma \Rightarrow \Delta, A \rightarrow B, B}{G \mid \Gamma \Rightarrow \Delta, A \rightarrow B}$

**$\square_L$**

$\frac{G \mid \langle A \rangle, \square A, \Gamma \Rightarrow \Delta}{G \mid \square A, \Gamma \Rightarrow \Delta}$

**$\square_R$**

$\frac{G \mid \langle \Sigma \rangle, \Gamma \Rightarrow \Delta, \square B \mid \Sigma \Rightarrow B}{G \mid \langle \Sigma \rangle, \Gamma \Rightarrow \Delta, \square B \mid \Sigma \Rightarrow B}$

**$\Rightarrow_1$**

$\frac{G \mid A \Rightarrow B}{G \mid A \Rightarrow B}$

**$\Rightarrow_2$**

$\frac{G \mid A \Rightarrow B}{G \mid A \Rightarrow B, \Sigma}$

$\frac{G \mid A \Rightarrow \Sigma}{|\Sigma| \geq 1}$

$\frac{G \mid \langle \top \rangle, \Gamma \Rightarrow \Delta}{G \mid \Gamma \Rightarrow \Delta}$

$\frac{G \mid \langle \Sigma, \Pi \rangle, \langle \Sigma \rangle, \langle \Pi \rangle, \Gamma \Rightarrow \Delta}{G \mid \langle \Sigma \rangle, \langle \Pi \rangle, \Gamma \Rightarrow \Delta}$

**Figure:** Rules of $\mathcal{H}_{E^*}$. 
Table of Contents

1. Non-Normal Modal Logics
2. Bi-neighbourhood semantics
   Basic System \( E \)
   Extensions
3. Sequent calculi \( \text{LSE}^* \)
4. PRONOM
   Finding a derivation
   Counter-model construction
5. Hypersequent Calculi \( \mathcal{H}_E^* \)
6. HYPNO
   Finding a derivation
   Counter-model construction
7. Conclusions
Basic ideas

- Same ideas of PRONOM
- Prolog implementation inspired by lean $TAP$
- Hypersequent represented with a Prolog list whose elements are Prolog terms:
  - `singleSeq(Gamma,Delta,Additional)`
  - each one represents a sequent $\Gamma \Rightarrow \Delta$ in the hypersequent
  - `Additional`: Prolog list for termination
- Two main predicates for finding either a derivation or a countermodel
Clause for axiom

\[
\text{terminating\_proof\_search}(\text{Hyper}, \text{tree}(\text{axiom}, \text{PrintableHyper}, \text{no}, \text{no})):\n\text{- member}(\text{singleSeq}([[\Gamma,\Delta], _], \text{Hyper}), \text{- member}(P, \Gamma), \text{member}(P, \Delta), !, \text{extractPrintableSequents}(\text{Hyper}, \text{PrintableHyper})).
\]
Clause for $\Box_R$

```prolog
terminating_proof_search(Hyper, tree(rbox, PrintableHyper, Sub1, Sub2)):-
    select(singleSeq([Gamma, Delta], Additional), Hyper, NewHyper),
    member(Sigma, Gamma),
    is_list(Sigma),
    member(box B, Delta),
    list_to_ord_set(Sigma, SigmaOrd),
    \+ member(apdR(SigmaOrd, B), Additional), !,
    terminating_proof_search([singleSeq([Sigma, [B]], [])|]
        [singleSeq([Gamma, Delta], [apdR(SigmaOrd, B)|Additional]]|NewHyper], Sub1),
    terminating_proof_search([singleSeq([], [B => Sigma]], [])|]
        [singleSeq([Gamma, Delta], [apdR(SigmaOrd, B)|Additional]]|NewHyper], Sub2),
    extractPrintableSequents(Hyper, PrintableHyper).
```

Gian Luca POZZATO
13 febbraio 2020
Countermodel construction

\[
\text{build\_saturated\_branch}(\text{Hyper}, \text{Model}) :- \\
\quad \text{select}\left(\text{singleSeq}(\left[\Gamma, \Delta\right], \text{Additional}), \text{Hyper}, \text{NewHyper}\right), \\
\quad \text{member}(\text{Sigma}, \Gamma), \text{is\_list}(\text{Sigma}), \text{member}(\text{box B}, \Delta), \\
\quad \text{list\_to\_ord\_set}(\text{Sigma}, \text{SigmaOrd}), \\
\quad \text{\textbackslash+member}(\text{apdR}(\text{SigmaOrd, B}), \text{Additional}), \\
\quad \text{build\_saturated\_branch}(\left[\text{singleSeq}(\left[\text{Sigma}, [B]\right], []), \text{\textbackslash}\text{singleSeq}(\left[\Gamma, \Delta\right], [\text{apdR}(\text{SigmaOrd, B})|\text{Additional}])|\text{NewHyper}\right], \text{Model}).
\]

\[
\text{build\_saturated\_branch}(\text{Hyper}, \text{Model}) :- \\
\quad \text{select}\left(\text{singleSeq}(\left[\Gamma, \Delta\right], \text{Additional}), \text{Hyper}, \text{NewHyper}\right), \\
\quad \text{member}(\text{Sigma}, \Gamma), \text{is\_list}(\text{Sigma}), \text{member}(\text{box B}, \Delta), \\
\quad \text{list\_to\_ord\_set}(\text{Sigma}, \text{SigmaOrd}), \\
\quad \text{\textbackslash+member}(\text{apdR}(\text{SigmaOrd, B}), \text{Additional}), \\
\quad \text{build\_saturated\_branch}(\left[\text{singleSeq}(\left[[], [B \Rightarrow \text{Sigma}]\right], []), \text{\textbackslash}\text{singleSeq}(\left[\Gamma, \Delta\right], [\text{apdR}(\text{SigmaOrd, B})|\text{Additional}])|\text{NewHyper}\right], \text{Model}).
\]
Implementation

- Last clause:

```
Countermodel
build_saturated_branch(Hyper, model(Hyper)):-
\+instanceOfAnAxiom(Hyper).
```
Web application

HYPNO

THEOREM PROVING WITH HYPERSEQUENT CALCULI FOR NON-NORMAL MODAL LOGICS

A PROLOG IMPLEMENTATION OF HYPERSEQUENT SEQUENT CALCULI FOR LOGIC E AND STANDARD EXTENSIONS WITH M, N, AND C

Non-Normal Modal Logics
Bi-neighbourhood semantics
Sequent calculi LSE*
PRONOM
Hypersequent Calculi $\mathcal{H}_E^*$
HYPNO
Conclusions

Universitá degli Studi di Torino

http://193.51.60.97:8000/HYPNO/
### Table of Contents

1. **Non-Normal Modal Logics**
   - Basic System ε
   - Extensions

2. **Bi-neighbourhood semantics**
   - Basic System ε
   - Extensions

3. **Sequent calculi LSE**

4. **PRONOM**
   - Finding a derivation
   - Counter-model construction

5. **Hypersequent Calculi Hε**

6. **HYPNO**
   - Finding a derivation
   - Counter-model construction

7. **Conclusions**
Non-normal modal logics

Our contribution

• sequent and hypesequent calculi for NNMLs
• first programs providing both proof-search and countermodel generation for the whole cube of NNML

Future Works

• use of free variables for term instantiation
• implement an automated transformation of bi-neighbourhood countermodels into standard neighbourhood models
AIIA 2019 Best Student Paper Award

is given to
Gian Luca Pozzato

for co-authoring the paper

PRONOM: proof-search and countermodel generation

for Non-Normal Modal Logics

Mario Alviano  Gianluigi Greco  Francesco Scarcello

Rende, 21 November 2019
Description Logics

- Important formalisms of knowledge representation
- Two key advantages:
  - well-defined semantics based on first-order logic
  - good trade-off between expressivity and complexity
- at the base of languages for the semantic (e.g. OWL)
Description Logics

• Important formalisms of knowledge representation
• Two key advantages:
  • well-defined semantics based on first-order logic
  • good trade-off between expressivity and complexity
• at the base of languages for the semantic (e.g. OWL)

Knowledge bases

• Two components:
Description Logics

- Important formalisms of knowledge representation
- Two key advantages:
  - well-defined semantics based on first-order logic
  - good trade-off between expressivity and complexity
- at the base of languages for the semantic (e.g. OWL)

Knowledge bases

- Two components:
  - TBox=inclusion relations among concepts (e.g. Dog ⊑ Mammal)
Description Logics

- Important formalisms of knowledge representation
- Two key advantages:
  - well-defined semantics based on first-order logic
  - good trade-off between expressivity and complexity
- at the base of languages for the semantic (e.g. OWL)

Knowledge bases

- Two components:
  - TBox= inclusion relations among concepts (e.g. \( \text{Dog} \sqsubseteq \text{Mammal} \))
  - ABox= instances of concepts and roles = properties and relations among individuals (e.g. \( \text{Dog(saki)} \))
Description Logics

DLs with nonmonotonic features

- need of representing prototypical properties and of reasoning about defeasible inheritance
- handle defeasible inheritance needs the integration of some kind of nonmonotonic reasoning mechanism
  - DLs + MKNF
  - DLs + circumscription
  - DLs + default
- all these methods present some difficulties ...
Description Logics

DLs with typicality

- Non-monotonic extensions of Description Logics for reasoning about prototypical properties and inheritance with exceptions
  - Basic idea: to extend DLs with a typicality operator $T$
  - $T(C)$ singles out the “most normal” instances of the concept $C$

Basic notions

- A KB comprises assertions $T(C) \sqsubseteq D$
- $T(Dog) \sqsubseteq Affectionate$ means “normally, dogs are affectionate”
DLs with typicality

- Non-monotonic extensions of Description Logics for reasoning about prototypical properties and inheritance with exceptions
- Basic idea: to extend DLs with a typicality operator $T$
- $T(C)$ singles out the “most normal” instances of the concept $C$
- semantics of $T$ defined by a set of postulates that are a restatement of Lehmann-Magidor axioms of rational logic

Basic notions

- A KB comprises assertions $T(C) \sqsubseteq D$
- $T(Dog) \sqsubseteq Affectionate$ means “normally, dogs are affectionate”
- $T$ is nonmonotonic
  - $C \sqsubseteq D$ does not imply $T(C) \sqsubseteq T(D)$
Description Logics

Example

\[ T(Pig) \subseteq \neg FireBreathing \]
\[ T(Pig \cap Pokemon) \subseteq FireBreathing \]

Reasoning
Description Logics

Example

\[ T(Pig) \sqsubseteq \neg FireBreathing \]
\[ T(Pig \cap Pokemon) \sqsubseteq FireBreathing \]

Reasoning

- ABox:
  - Pig(tepig)
Example

\[ T(Pig) \sqsubseteq \neg FireBreathing \]
\[ T(Pig \cap Pokemon) \sqsubseteq FireBreathing \]

Reasoning

- ABox:
  - \( Pig(tepig) \)
- Expected conclusions:
  - \( \neg FireBreathing(tepig) \)
Example

\[ T(Pig) \sqsubseteq \neg \text{FireBreathing} \]
\[ T(Pig \sqcup \text{Pokemon}) \sqsubseteq \text{FireBreathing} \]

Reasoning

- **ABox:**
  - \( Pig(tepig), Pokemon(tepig) \)
- **Expected conclusions:**
  - \( FireBreathing(tepig) \)
Semantics

- $= \langle \Delta^I, <, . \rangle$
  
  - additional ingredient: preference relation among domain elements
  
  - $<$ is an irreflexive, transitive, modular and well-founded relation over $\Delta$:
    
    - for all $S \subseteq \Delta^I$, for all $x \in S$, either $x \in \text{Min}_<(S)$ or $\exists y \in \text{Min}_<(S)$ such that $y < x$
    
    - $\text{Min}_<(S) = \{ u : u \in S \text{ and } \not\exists z \in S \text{ s.t. } z < u \}$
    
  - Semantics of the $T$ operator: $(T(C))^I = \text{Min}_<(C^I)$
Rational closure

- Preference relation among models of a KB
  - $M_1 < M_2$ if $M_1$ contains less exceptional (not minimal) elements
  - $M$ minimal model of KB if there is no $M'$ model of KB such that $M' < M$
Rational closure

- Preference relation among models of a KB
  - $M_1 < M_2$ if $M_1$ contains less exceptional (not minimal) elements
  - $M$ minimal model of KB if there is no $M'$ model of KB such that $M' < M$
- Minimal entailment
  - $\text{KB} \models_{\text{min}} F$ if $F$ holds in all minimal models of KB
Rational closure

- Preference relation among models of a KB
  - $M_1 < M_2$ if $M_1$ contains less exceptional (not minimal) elements
  - $M$ minimal model of KB if there is no $M'$ model of KB such that $M' < M$

- Minimal entailment
  - $KB \models_{min} F$ if $F$ holds in all minimal models of KB

- Nonmonotonic logic
  - $KB \models_{min} F$ does not imply $KB' \models_{min} F$ with $KB' \supset KB$

- Corresponds to a notion of rational closure of KB
Probabilities for Concept combination

- Further extension: \( p :: T(C) \subseteq D \)
- \( p \in (0.5, 1) \) degree of belief
- prototype of concepts \( C_H \) and \( C_M \)
- distributed semantics
  - selection of typicality inclusion to be considered as true \( \Rightarrow \) different scenarios
  - probability distribution over scenarios
- DISPONTE semantics + heuristics from cognitive semantics for concept combination
  - Prototype of the combined concept \( C_H \cap C_M \)
Description Logics

Application to RaiPlay (with Centro Ricerche RAI)

- Recommender system for RaiPlay
- New genres obtained by combining basic ones
- Glass box approach
- Prototypes from data