Logics for AI: From Dreams to Formal Methods

Stéphane Demri
CNRS, Laboratoire Méthodes Formelles
https://cv.hal.science/stephane-demri

Séminaire ARISTOTE — “IA Hybride”
January 18th, 2024
Symbolic AI

Symbolic AI: knowledge representation with symbols and rules manipulating symbols for reasoning.

Traditionally, symbolic AI convenient for settings where the rules are clear cut, and you can easily obtain input and transform it into symbols.

Logics for AI:
- formal languages,
- semantics,
- proof procedures to reason about it.
Subfields in IJCAI’2023 proceedings

- Agent-based and Multi-agent Systems
- AI Ethics, Trust, Fairness
- Computer Vision
- Constraint Satisfaction and Optimization
- Data Mining
- Game Theory and Economic Paradigms
- Humans and AI
- Knowledge Representation and Reasoning
- Machine Learning
- Multidisciplinary Topics and Applications
- Natural Language Processing
- Planning and Scheduling
- Robotics
- Search
- Uncertainty in AI
- etc.
Subfields in KR’23 proceedings

- Argumentation
- Automated reasoning
- Belief merging / revision
- Conditionals
- Description logics
- Epistemic logic
- Knowledge representation and machine learning
- Multi-agent systems
- Strategic reasoning
- Systems and robotics
- Temporal reasoning
Reasoning on Ontologies

▶ Ontology: formal specification of some domain with concepts, objects, relationships between concepts, objects, etc.

▶ Backbone of ontologies includes:
  – taxonomy (classification of objects),
  – axioms (to constrain the models of the defined terms).

▶ Well-known ontologies:
  – Medical ontology SNOMED-CT.
  – NCI Thesaurus (National Cancer Institute, USA).
  – Gene ontology (world largest source of information on the functions of genes).

(classification of medical terms: diseases, body parts, drugs, etc.)

▶ Free ontology editor Protégé
  http://protege.stanford.edu/
Challenges with Ontologies

- How to define ontologies and to reason on it?
- How to repair faulty ontologies?
- How to add new concepts or axioms without affecting the old inferences?
- More generally, how to extract from the ontologies more knowledge than what is explicitly specified?
  - inferences about individuals,
  - concept subsumptions, non-redundancy,
  - concept hierarchy, consistency of concepts,
  - etc.
Why Description Logics?

▶ Formal languages for concepts, relations and instances.

▶ DLs have all one needs to formalise ontologies.

▶ Computational properties.
  – Acceptable trade-off between expressivity and complexity.
  – Decidability and often tractability.
  – Implementation in tools of the main reasoning tasks.

▶ A remarkable suite of languages and tools.
  See e.g.,
  – OWL: Web Ontology Language.
  – Protégé: ontology editor.
  – FaCT++: DL reasoner supporting OWL DL.
Description Logic $\mathcal{ALC}$ in a Nutshell

▶ Language of complex concepts.

$$C ::= \top | \bot | A | \neg C | C \sqcap C | C \sqcup C | \exists r.C | \forall r.C,$$

with concept names $A$ and role names $r$.

▶ Interpretation $\mathcal{I} \overset{\text{def}}{=} (\Delta^\mathcal{I}, \cdot^{\mathcal{I}})$

- $\Delta^\mathcal{I}$: non-empty set (the domain).
- $\mathcal{I}$: interpretation function such that

$$A^\mathcal{I} \subseteq \Delta^\mathcal{I} \quad r^\mathcal{I} \subseteq \Delta^\mathcal{I} \times \Delta^\mathcal{I}$$

▶ $C^\mathcal{I} \subseteq \Delta^\mathcal{I}$ defined inductively providing the semantics to complex concepts.
Concept Inclusions and Decision Problems

▶ General concept inclusions $C \sqsubseteq D$ (GCIs).

E.g., $\text{Employee} \sqsubseteq \exists \text{WorksFor} \top$

\[ \mathcal{I} \models C \sqsubseteq D \quad \overset{\text{def}}{\iff} \quad C^\mathcal{I} \subseteq D^\mathcal{I} \]

▶ Terminological Box (TBox) $\mathcal{T}$: finite set of GCIs.

▶ Interpretation $\mathcal{I} = (\Delta^\mathcal{I}, \cdot^\mathcal{I})$, TBox $\mathcal{T}$.

\[ \mathcal{I} \models \mathcal{T} \quad \overset{\text{def}}{\iff} \quad \text{for all } C \sqsubseteq D \in \mathcal{T}, \mathcal{I} \models C \sqsubseteq D \]

▶ Concept satisfiability problem w.r.t. general TBoxes:

Input: A concept $C_0$ and a TBox $\mathcal{T}$.

Question: Is there an interpretation $\mathcal{I}$ such that $\mathcal{I} \models \mathcal{T}$ and $C_0^\mathcal{I} \neq \emptyset$?

▶ This problem is $\text{EXPSPACE}$-complete.
Description Logics with Concrete Domains

- Need to express concrete properties about data in ontologies (e.g. age, duration, name, size, etc.)

- Examples of concrete domains:
  \((\mathbb{N}, <, +1)\), \((\mathbb{Q}, <, =)\), \((\mathbb{N}, <, =)\), \((\{0, 1\}^*, <_{\text{pre}}, <_{\text{suf}})\).

- General scheme for integrating concrete domains in DLs.
  [Baader & Hanschke, IJCAI’91]
  - declarative semantics close to the usual semantics for DLs,
  - generic extensions of DLs with various concrete domains,
  - tableaux-based algorithms combined with theory reasoning.
Methods for Handling Concrete Domains

- Tableaux-based calculi for “ω-admissible” domains. [Lutz & Miličić, JAR 2007]

- Translation into decidable logics [Carapelle & Turhan, ECAI’16]
  - Decidability of concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\mathbb{N}, <, =, (=n)_{n \in \mathbb{N}})$.

- Translation into automata-based problems.
  - Concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\mathbb{N}, <, =, (=n)_{n \in \mathbb{N}})$ in EXPTIME. [Labai & Ortiz & Šimkus, KR’20]
  - Concept satisfiability problem w.r.t. general TBoxes for $\mathcal{ALC}(\{0, 1\}^*, <_{\text{pre}})$ in EXPTIME. [Demri & Quaas, JELIA’23]

(constraint automata for data words or data trees)
Temporal Logics with Concrete Domains

- Concrete domains in TCS:
  - Satisfiability Modulo Theory (SMT) solvers. String theories, arithmetical theories, array theories, etc.
  - Verification of database-driven systems.
  - Temporal logics with arithmetical constraints.

- “Infinitely often $x$ is a prefix of the next value for $y$”:
  $\text{GF}(x <_{\text{pre}} X_y)$.

- Satisfiability problem for $\text{CTL}^*(\mathbb{Z}, <, =, (=n)_{n \in \mathbb{Z}})$ is decidable in $2\text{EXP\text{-}TIME}$.

[Carapelle et al, JCSS 2016; Demri & Quaas, CONCUR’23]

($\text{CTL}^*$: well-known logic related to model-checking)
Another Success Story: Logics of Strategic Ability

- To express that a coalition of agents has a collective strategy to achieve some goal and to reason on it.

- A strategy is a conditional plan intended to work whatever the other agents do.

- Well-known specimens.
  - Coalition Logic CL. (one-step strategies)
  - Alternating-time temporal logic ATL. (generalisation of temporal logics)
  - Strategy Logic SL. (explicit quantification over strategies)

(with a huge amount of variants)
Multi-Agents Systems

Multi-agent systems are transition systems in which transitions are fired when simultaneous actions are performed by different agents.

\[
\begin{align*}
\text{pos}_0 & \quad \text{pos}_1 \\
\text{pos}_2 & \quad \text{pos}_3
\end{align*}
\]

- (wait, wait), (push, push)
- (push, wait)
- (wait, push)
- (wait, push)
- (push, wait)
- (wait, wait)
- (push, push)
- (push, push)
- (wait, push)

14
ATL-like logics

- \( \langle A \rangle \Phi \): coalition \( A \) has a collective strategy to enforce the temporal property \( \Phi \).

- A collective strategy is a tuple of individual strategies.

\[ \varphi ::= p \mid \neg \varphi \mid \varphi \land \varphi \mid \langle A \rangle X \varphi \mid \langle A \rangle G \varphi \mid \langle A \rangle \varphi U \varphi \]

\[ p \in \text{PROP} \quad A \subseteq \text{Agt} \]

- \( \mathcal{M}, s \models \langle A \rangle G \varphi \) def
  \emph{exists strategy} \( \sigma \) such that
  \emph{forall computations} \( \lambda = s_0 \rightarrow s_1 \ldots \) from \( s \) respecting \( \sigma \),
  \emph{forall positions} \( i \), we have \( \mathcal{M}, s_i \models \varphi \).

- Tractable model-checking problems and automata-based satisfiability-checking decision procedures.
More Ingredients

- Resource-aware logics:
  - actions have costs/weights,
  - formulae may specify constraints about such (cumulative) costs/weights.

  E.g. [Belardinelli & Demri, AI 2021; Bulling & Goranko, AAMAS 2022]

  (relationships with energy games, counter machines)

- Strategic reasoning with knowledge operators.
  See e.g. [Agotnes, Synthese 2006]

- Restriction on agents’ knowledge.
  - Strategy logics with imperfect information.
  - Undecidability can be obtained easily.
  - Fragments including those with hierarchies of knowledge leads to less expensive reasoning tasks.

  See e.g. [Berthon et al, TOCL 2021]
In My Reading List

▶ Well-identified potential interactions between machine learning and symbolic AI.
  – Machine learning can be used to solve logical problems and to accelerate verification/automated techniques.
  – Logical methods can be used to complement learning algorithms to improve the precision and explainability.

▶ Unique characterisability and learnability of temporal instance queries

[Fortin et al., KR’22]

(Example sets ($E^+$, $E^-$) to characterise temporal formulae)

▶ An SMT-Based approach for verifying binarized neural networks

[Amir et al., TACAS’21]

(SMT-based approach for formal verification of binarized neural networks)
Logics for AI: the Great Return?

- Beyond knowledge representation and reasoning for description logics, strategy logics, dynamic epistemic logics, etc.?

- Arithmetical theories for the verification of neural networks.

- Learning logical formulae and unique characterisations.
A Selection of Bibliographical References
Logical Formalisms with Concrete Domains

- “Description Logics with Concrete Domains—A Survey”
  [Lutz, AiML’02]
  (a classical paper on description logics with concrete domains)

- “Concrete domains in logics: a survey”
  [Demri & Quaas, SIGLOG News 2021]
  (a brief survey)

- “Using Model Theory to Find Decidable and Tractable Description Logics with Concrete Domains”
  [Baader & Rydval, JAR 2022]
  (model theory for DLs with concrete domains)
Strategic Reasoning and Resources

▶ “Alternating-time temporal logic”
  [Alur & Henzinger & Kupferman, JACM 2022]
  (the classical paper about ATL)

▶ “Combining quantitative and qualitative reasoning in concurrent multi-player games”
  [Bulling & Goranko, AAMAS 2022]
  (how to mix quantitative and qualitative objectives)

▶ “Strategic reasoning with a bounded number of resources: The quest for tractability”
  [Belardinelli & Demri, AI 2021]
  (complexity analysis to reason with resources in ATL)
Learning and Modal/Temporal Logics

- “Scalable Anytime Algorithms for Learning Fragments of Linear Temporal Logic” [Raha et al., JOSS 2024]
  (how to learn temporal formulae)

- Unique characterisability and learnability of temporal instance queries
  [Fortin et al., KR’22]
  (Example sets \((E^+, E^-)\) to characterise temporal formulae)

- Logic of “Black Box” classifier systems
  [Liu & Lorini, WoLLIC’22]
  (product modal logic for multi-classifier models)
Logic, Verification and Neural Networks

▶ “Neural Network Verification with Proof Production”  
[Isac et al., FMCAD’22]  
(how to add proof production capabilities)

▶ “An SMT-Based approach for verifying binarized neural networks”  
[Amir et al., TACAS’21]  
(SMT-based approach for formal verification of binarized neural networks)

▶ “Simplifying neural networks using formal verification”  
[Gokulanathan et al., NFM’20]  
(how to remove components in deep neural networks)