Towards Novel Techniques for Reasoning in expressive Description Logics[1]

Uwe Keller

uwe.keller@deri.at

Digital Enterprise Research Institute (DERI)
University of Innsbruck, Austria

June 21, 2007

[1] Presentation at the KnowledgeWeb Ph.D Symposium (KWEPSY 2007) at the European Semantic Web Conference (ESWC), June 6, 2007, Innsbruck, Austria.

Problem Statement

- **Description Logics** = well-behaved, logic-based KR frameworks with a number of apps (e.g. OWL)
- **Research on Reasoning** in DLs focussed for a long while on a few classical approaches:
  - Structural subsumption tests, Tableau-, Automata-based and SMT satisfiability tests
- [Hor05]: **Finding efficient ways of reasoning with DLs is still challenging and needed**
  - even for **existing languages**
  - required **especially for expressive DLs**
- **Clearly:** High worst-case complexity of reasoning in expressive DLs $\Rightarrow$ *A single method that works well on all possible problems is impossible!*
- **Shift of focus can be very successful**: DL Reasoning with (FOL-)Resolution (e.g. KAON2, HermiT)

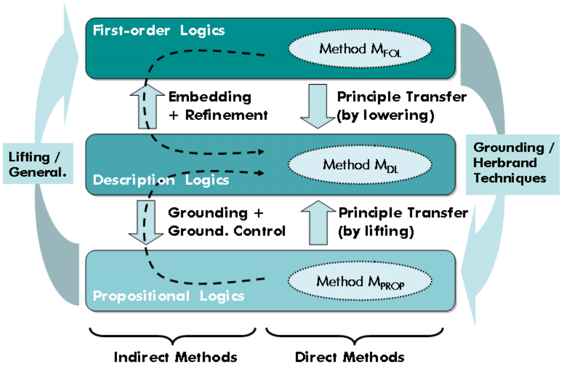
Expected Contribution & Research Questions

Enrich the available machinery for DL reasoning by a couple of novel techniques and understand these techniques for DLs in detail: KB satisfiability, TBoxes, ABoxes, RBoxes, $\mathcal{SHIQ}$, $\mathcal{SROIQ}$

- Are there other families of reasoning methods (that are work differently from Tableau- and Resolution-Methods) for standard DL reasoning tasks?
- How can we use them for DL reasoning and for which DLs can they be used?
- How can we support certain expressive features that are relevant for many practical applications, such as concrete datatypes?
- How do they perform in comparison to the current approaches (practically or even theoretically)?
- How can we tweak them to become efficient methods for DL reasoning?
- Are there specific fragments of DLs where the new techniques are particularly efficient?

Proposed Solution: How to find novel methods?

Consider DLs as intermediate logics



Examples for inference calculi:

- Semantic Tableau, Resolution, (FOL)-Davis-Putnam Procedure, ...

Candidate Formalisms

Proposal for investigation:

- **Binary-Decision-Diagrams (BDDs) [Bry86]** (as kind of a „knowledge structure")
- Stålmarck's Method [SS90] (as an inference calculus, not considered here)

Motivation / Rationale:

- Both have proven to be very successful techniques for propositional reasoning
- For both there are already proposals for variants of the calculi for First-order Logic with encouraging results.

Intuition / Hope:

- Lifting to DL (instead of FOL) / Restriction of the FOL formalisms „in a proper way" preserves significant portion of the power of the propositional method

Binary Decision Diagrams (BDDs) [Bry86]

- Every formula ϕ with n free prop. vars represents a n-ary boolean function $\phi(x_1, \ldots, x_n)$
- BDD = simple data structure for representing an (n-ary) boolean function $f : \{0,1\}^n \to \{0,1\}$

Definition (Binary Decision Diagram (BDD))

A *Binary Decision Diagram* is a rooted, directed, acyclic graph where leaf nodes are labeled either with 0 or 1. Each non-leaf node n has exactly two child nodes $low(n)$ and $high(n)$ and is labeled with a boolean variable $var(n)$. The edge from a node to a low (high) child represents an assignment of the variable to 0 (1).

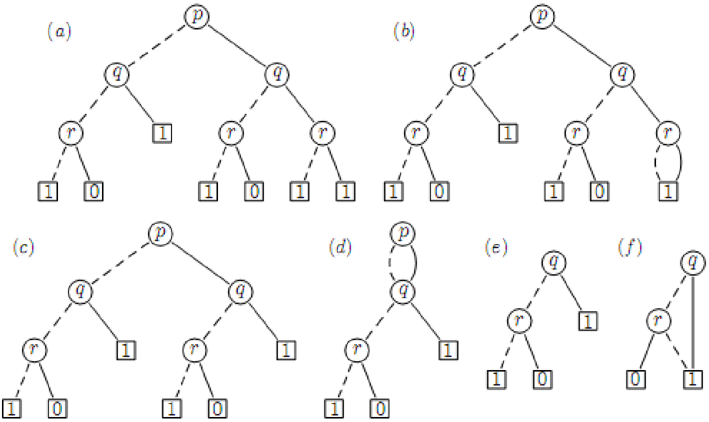
Such a BDD is called *ordered* if on each path from the root to the leaf nodes all variables are consistent with a linear order on the boolean variables. It is called *reduced* if the graph is reduced according to two rules: (i) merge any isomorphic subgraphs, and (ii) eliminate any node whose two children are the same.

In the literature: most often „BDD" actually refers to ROBDD!

What is a BDD? Representing a formula as a BDD!

Stepwise transformation to a (RO)BDD for the formula

$\phi(p, q, r) = (p \rightarrow q) \wedge r \rightarrow (p \leftrightarrow r) \wedge q$



Construction can be done efficiently (in the size of the graph), that in turn depends critically on the chosen variable order!

Properties of BDDs

- Given a fixed ordering on the variables, BDDs are a canonical representation of a boolean function, i.e. for any boolean function f there is *exactly one* ROBDD B for the given ordering s.t. $I(B) = f$
- This makes BDDs unique, semantic representations of formulae (= independent of syntax) since any two equivalent formulae have the same ROBDD.
- In practice BDDs are often small representations (e.g. polynomial) wrt. truth tables (given a suitable var ordering)
- BDDs are are semantically rich representations: $1/0$-pathes, complete coverage, DNF and CNF
- Can be used as the basis for various inference calculus dealing with cubes or clauses, especially tableau- and resolution-like inferences, *even at the same time*

Richness of BDDs (*as a knowledge structure*) should enable novel and perhaps better techniques for guiding proof search in an inference calculus

How to Reason with BDDs ?

- **Propositional Logic**: ϕ unsatisfiable iff. ROBDD B_ϕ of ϕ contains only the leaf node 0
- **First-order Logic** [GT03]: Unsatisfiability test via direct application of *Herbrand's Theorem* which reduces FOL unsatisfiability to propositional unsatisfiability

Theorem (Herbrand's Theorem)

A first-order formula of the form $\phi = \forall x_1, \ldots, x_i.M$ where M is quantifier-free is unsatisfiable ***iff.*** *there exists a $k \geq 0$ and a var substitution σ such that $(M^1 \wedge M^2 \wedge \ldots \wedge M^k)\sigma$ is a propositionally unsatisfiable formulae, whereby M^i denotes a „new" (variable-distinct) copy of M.*

- Idea: Replace propositional variables as labels of the BDD by atomic formulae (e.g. $person(john), hasAncestor(x, y)$)

  1. Transform ϕ into universal prenex form, i.e. $\phi' = \forall x_1, \ldots, x_i.M$ where M is quantifier-free
  2. Construct BDD B_M for M. B_M and M are equi-satisfiable.
  3. For $k = 0, 1, \ldots$ search for a substitution σ such that the BDD of $(M^1 \wedge M^2 \wedge \ldots \wedge M^k)\sigma$ contains only the node 0

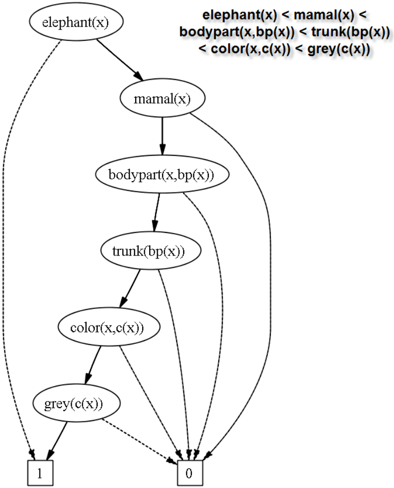
How to Reason with BDDs in DLs?

Approach 1: Modification of the FOL-BDD Calculus: Use FOL procedure and any mapping of a DL knowledge base K to an equi-satisfiable FOL KB $\pi(K)$ (e.g. standard translation).

- For **unsatisfiable** K: how to find k and σ efficiently?
- For **satisfiable** K: in principle **no termination** of the FOL procedure
- However, DLs are decidable FOL fragments! How to regain termination in the satisfiable case?
- **Possible approach:** Saturation of 1-pathes, blocking to prevent unnecessarily big models.

An Example of a FOL-BDD for a DL axiom

The DL axiom $\alpha = elephant \sqsubseteq mamal \sqcap \exists bodypart.trunk \sqcap \exists color.grey$ can be represented in FOL by $\phi = \forall x.(elephant(x) \Rightarrow (mamal(x) \wedge \exists b.(bodypart(x, b) \wedge trunk(b)) \wedge \exists c.(color(x, c) \wedge grey(c))))$.



How to Reason with BDDs in DLs? (2)

Approach 2: BDD-Tableau: Combine BDDs with the tableau data structure used in Tableau-based methods

- Use BDDs (whose inner nodes are labeled with concepts) instead of sets of concepts as labels for the individuals in a tableau
- This way: BDDs take over propositional reasoning (propositional constructors), tableau-style search for constructing a model (modal constructors)
- Some powerful optimizations are inherently built-in (e.g semantic branching), some others can be transferred from standard tableaus (e.g. blocking)
- Should work for any DL where there is a tableau-procedure already defined
- Is more generally applicable than SMT-based approaches (e.g. does not require tree model property), since model is explicitly represented
- Relation between BDDs and DPLL might lead to some novel optomizations of current tableau methods too

Evaluation of the Approach

Evaluation question: How do the designed novel methods perform wrt. state-of-the-art techniques?

- **Option 1:** Empirical Evaluation (Measurement)

  Implement the methods and compare to implementations of SoA techniques (FACT++, Pellet, Racer, KAON2, HermiT, ...)

  To be done in multiple ways:

  1. Randomly-generated Problems
  2. Real-world Problem Sets, e.g. GALEN, SNOMED, NCI
  3. Specifically-created Benchmark Problems

- **Option 2:** Theoretical Evaluation (Understanding)

  Compare deduction process to the ones performed by SoA techniques, analyze similarities and differences. Answer why does the method perform well or poor on a specific class of problems?

Future Work

- **Study the use of BDDs for the design and implementation of DL inference calculi**
  - Different Possible Approaches: BDD-Tableau (currently ongoing), FOL-BDDs specialized to DLs, or something like Modal-DD?
  - Find and study novel optimizations based on BDDs (e.g. for Optimizing DL-Tableau)
- **Evaluation**
  - Implementation (currently ongoing for BDD-Tableau)
  - Design of a comprehensive suite of benchmark problems
  - Analysis and explanation of the measurements
  - Comparison of the methods with each other and with SoA
- **Dito: Study Stålmarks Method in the context of DLs**
- **Get a better understanding of SAT in DLs**
  - What are key differences in comparison to SAT in propositional logics?

References

Randal E. Bryant.
Graph-based algorithms for Boolean function manipulation.
IEEE Transactions on Computers, C-35(8):677–691, August 1986.

Jan Friso Groote and Olga Tveretina.
Binary decision diagrams for first-order predicate logic.
J. Log. Algebr. Program., 57(1-2):1–22, 2003.

Ian Horrocks.
Applications of description logics: State of the art and research challenges.
In Frithjof Dau, Marie-Laure Mugnier, and Gerd Stumme, editors, *Proc. of the 13th Int. Conf. on Conceptual Structures (ICCS'05)*, number 3596 in Lecture Notes in Artificial Intelligence, pages 78–90. Springer, 2005.

G. Stålmarck and M. Säflund.
Modeling and verifying systems and software in propositional logic.
In *In Proceedings of International Conference on Safety of Computer Control Systems (IFAC SafeComp'90)*, pages pp. 31–36. Pergamon Press, Oxford, 1990.