Distributed Resolution for ALC

Anne Schlicht, Heiner Stuckenschmidt

UNIVERSITÄT MANNHEIM

April 15, 2008

Motivation

Efficient Reasoning on a Distributed DL Terminology

- Partitioned ontology
- Set of linked ontologies
  - owl:import
  - bridge rules, ε-connections
  - using terms defined in another ontology

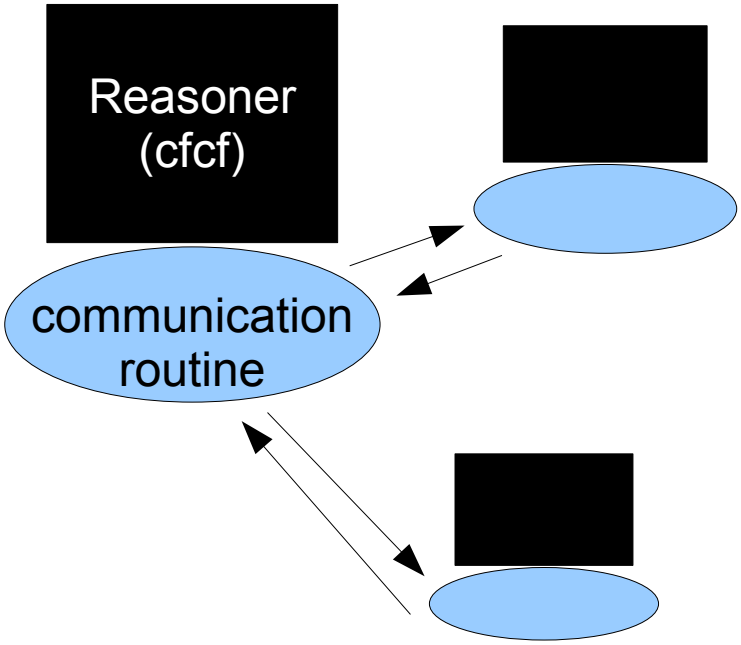
Tasks

- Satisfiability test
- Query answering

Black Box Approach

Guarantee Completeness by

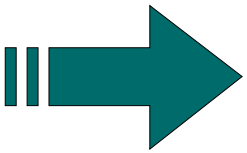
- property of local reasoning
- appropriate communication routine



Black Box: Partition-Based Reasoning

Based on completeness for consequence finding

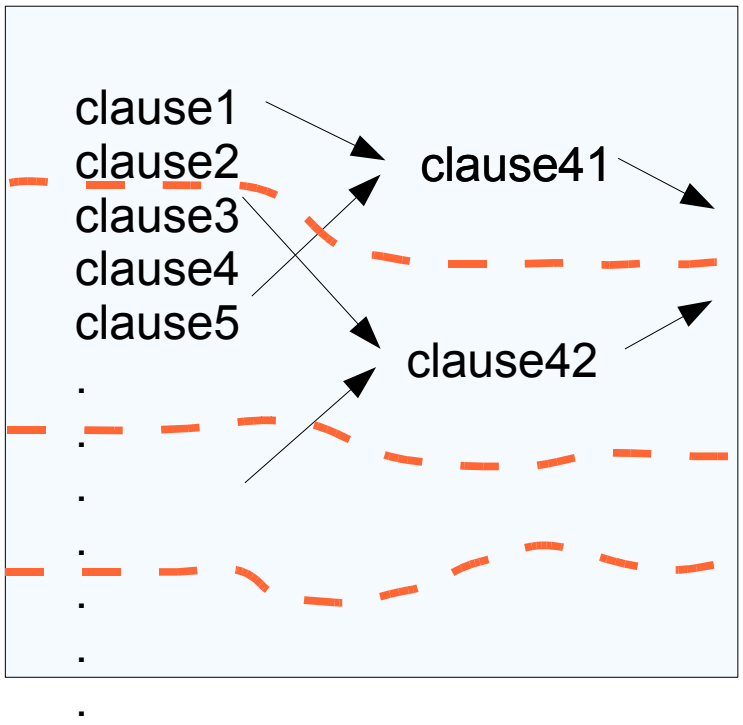
- Requires prolific derivation which is in general the opposite of efficiency. (a lot more derivations than necessary for refutational completeness)
- Efficient resolution methods do not have this property.



completeness for consequence finding is not an appropriate property for an efficient distributed reasoner.

Glass Box Approach

Perform the same inferences as a complete centralized reasoner



Glass Box: Distributed Resolution

Perform the same reasoning steps as a centralized reasoner but on different machines with seperate storages

- Allocate every inference to a module.
- Allocate axioms to modules such that:
  - for every inference, there is a module that contains all premises. (→ completeness)
  - inferences are not duplicated (→ efficiency)
  - clauses are not duplicated (→ representation)
  - only few clauses are propagated (→ communication)
  - local subsumption elimination is sufficient (→ efficiency)

Ordered Resolution

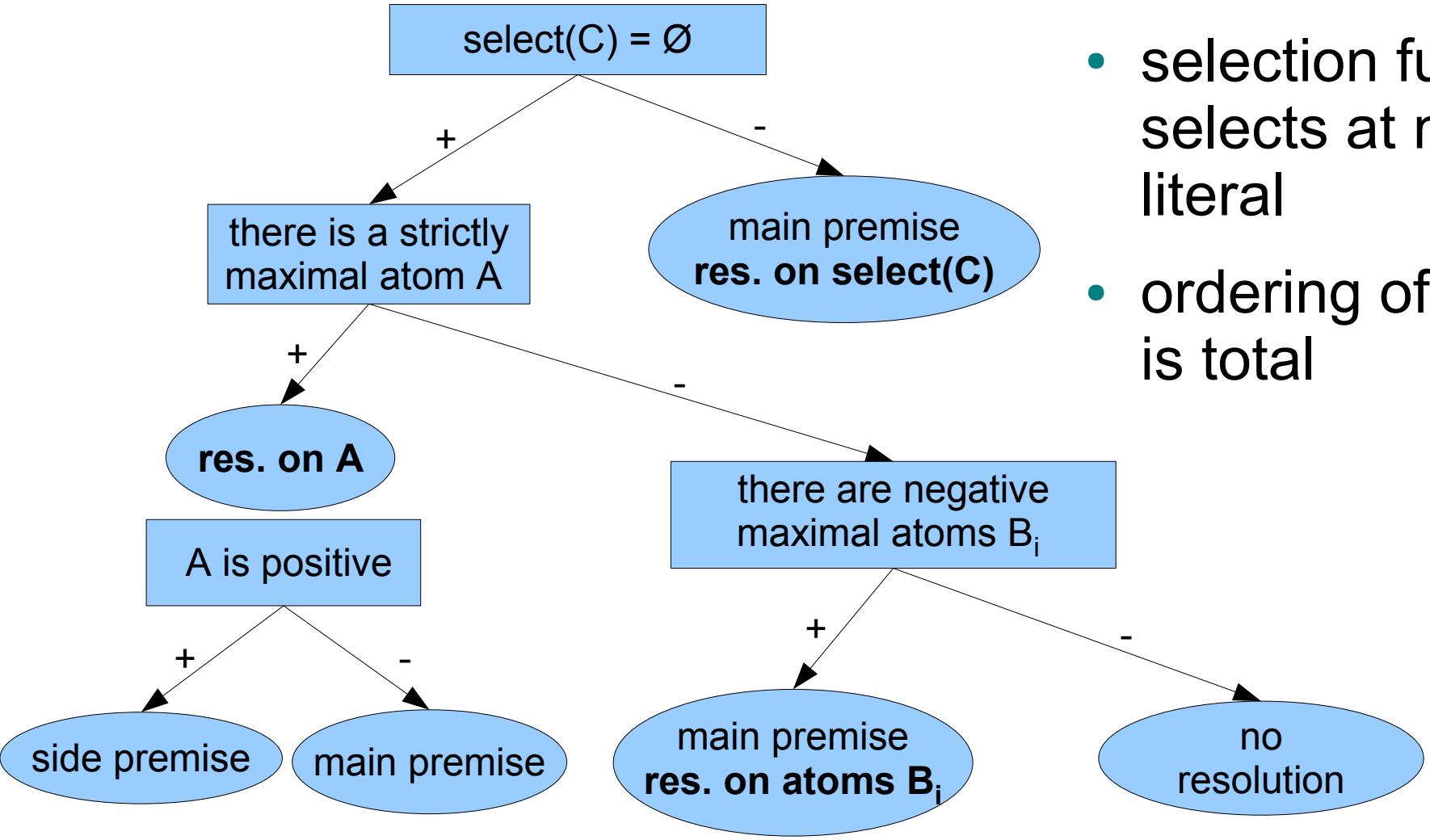
Parameters:

- selection function: subset of the negative literals of each clause
- ordering of literals

Ordered resolution $\dfrac{C \vee A \quad D \vee \neg B}{C\sigma \vee D\sigma}$

1. σ *is the most general unifier of* A *and* B
2. *either* B *is selected in* $D \vee \neg B$ *or else nothing is selected in* $D \vee \neg B$ *and* $B\sigma$ *is maximal w.r.t.* $D\sigma$
3. $A\sigma$ *is strictly maximal with respect to* $C\sigma$
4. *nothing is selected in* $C\sigma \vee A\sigma$

Resolvable Literal of a given clause C...



...is unique if

- selection function selects at most one literal
- ordering of literals is total

Method

Distributed Resolution Method:

- Based on a partitioning of predicates
- Every module is responsible for all inferences resolving upon literals *(¬)P(t)* with *P* contained in the local set of predicates.
- Derived clauses are propagated to the module responsible for the *unique* resolvable literal.

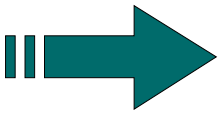
Adaption to ALC → Termination

- No termination for first order logic
- Adaption to ALC (Tammet):
  - simplify ontology prior to clausification ('folding', definitorial form)
  - calculus: ordered resolution + factoring
  - select negative binary literals
  - ordering: $R(x,f(x)) > \neg C(x)$ and $D(f(x)) > \neg C(x)$
  - set of *ALC clauses* is closed under this calculus

⇒ Terminates for ALC

Distributed Resolution for ALC

- ALC clauses contain at most one selected literal
- Ordering can be extended to total ordering without affecting completeness or termination



Distributed ALC-Resolution is complete and terminates

Glass Box: Distributed Resolution

Perform the same reasoning steps as a centralized reasoner but on different machines with seperate storages

- Allocate every inference to a module.
- Allocate axioms to modules such that:
  - for every inference, there is a module that contains all premises. (→ completeness)
  - inferences are not duplicated (→ efficiency)
  - clauses are not duplicated (→ representation)
  - only few clauses are propagated (→ communication)
  - local subsumption elimination is sufficient (→ efficiency)

First Results: Setting

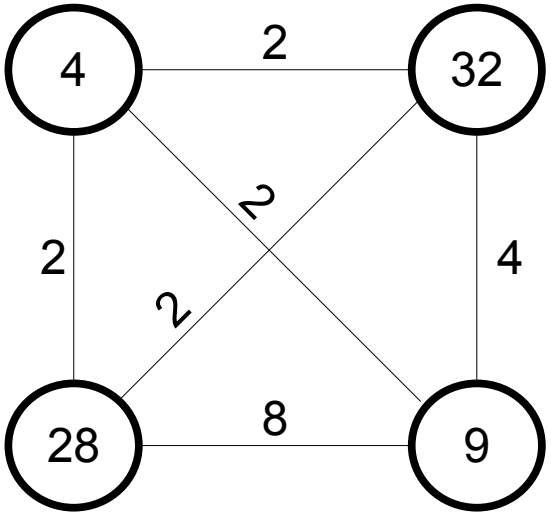
Simulation on Partitioned Ontology

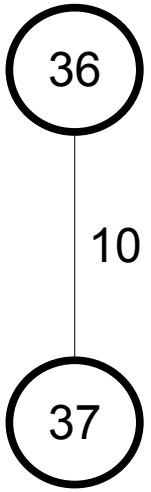
- Partitioning (using Pato):
  - Module numbers inserted into ontology terms
- Normalization: implemented in seperate tool
- Resolution (with SPASS):
  - Adapted for extended ordering
  - Configuration for ALC Resolution
  - Logging: pairs of module numbers
    - Communication: source and destination of derived
    - Reduction: destinations of subsumee and subsumer

First Results

Total number of clauses:

- input: 96
- derived: 84
- deleted: 20





- Communication:

  of derived clauses are propagated

  24%

  12%

- Reduction:

  of subsumptions are inter-module

  10%

  10%

Optimization / Extension

- Optimize partitioning
  - In advance
  - Dynamically
- Adapt selection function: select all literals hosted by the same module as the resolvable literal (from clauses with empty selection)
- Add inter-module reduction
- Extend to number restrictions (basic superposition)

Discussion

Thanks!

Questions?

Comments?

References

- E. Amir and S. McIlraith. Partition-based logical reasoning for first-order and propositional theories. Artificial Intelligence, 162(1-2):49–88, 2005.
- Tanel Tammet. Resolution methods for Decision Problems and Finite Model Building. PhD thesis, Chalmers University of Technology and University of Göteborg, 1992.
- Boris Motik. Reasoning in Description Logics using Resolution and Deductive Databases. PhD thesis, Univesität Karlsruhe (TH), Karlsruhe, Germany, January 2006.

Completeness of Distributed Resolution

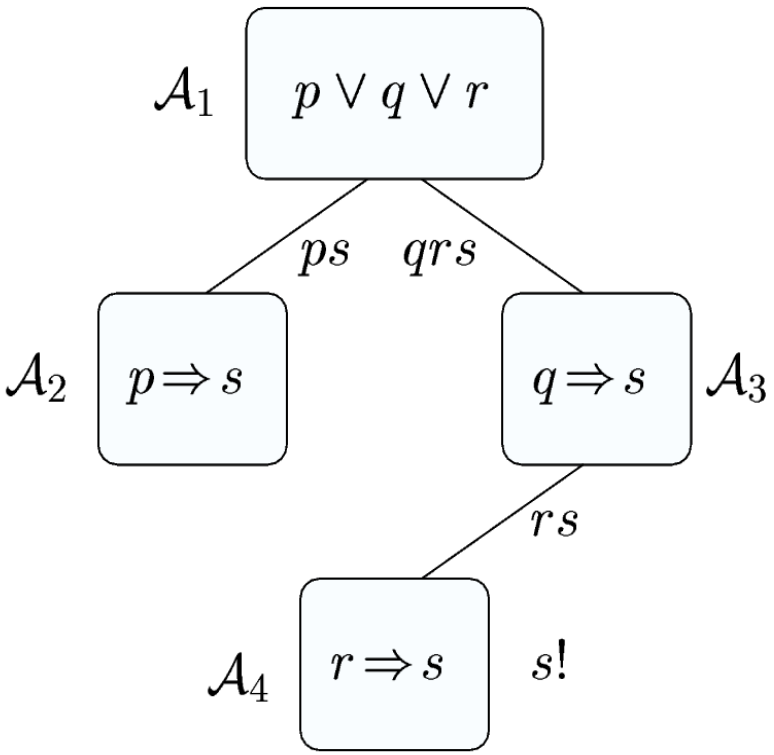
Proof:

- Assume the empty clause in not derived from an unsatisfiable KB by distributed ordered resolution.
- The first inference in the corresponding centralized reasoner that does not occur in the distributed version.
- The unified literals (*P(t1)* and ¬*P(t2)*) are the unique resolvable literals of their respective clauses.
- Both premises are send to the module responsible for predicate *P*.

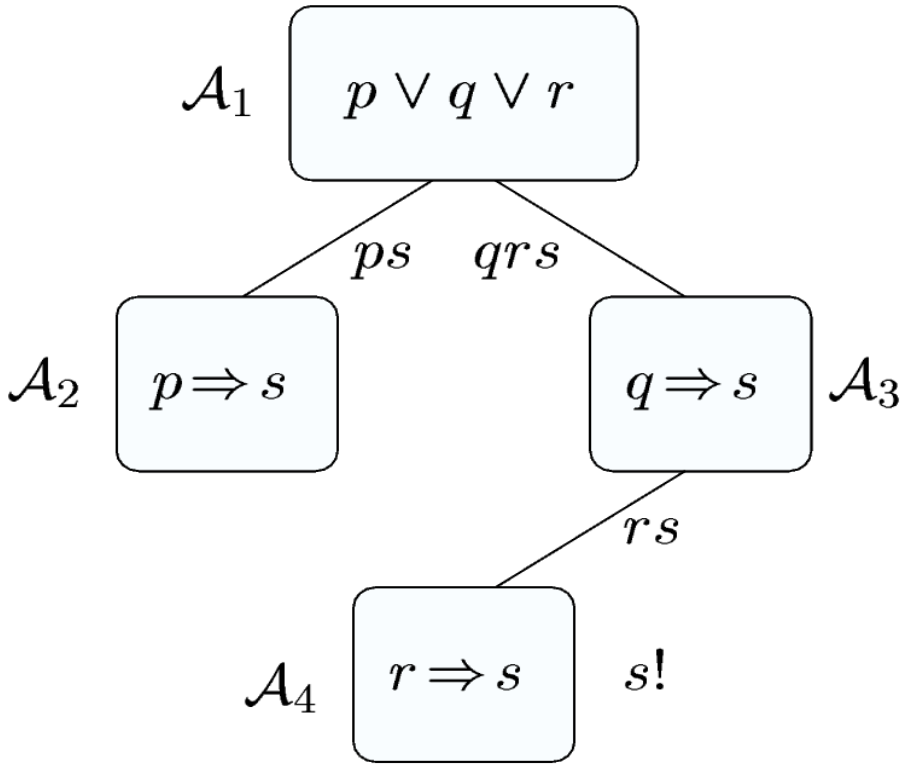
Black Box: Partition-Based Reasoning

- Compute module graph (based on shared language = symbols contained in more than one module)
- Treeify
- Concurrent local resolution, for every derived clause:
  - if clause is completely contained in a shared language, propagate to the corresponding module (direction to root)
- Prerequisite:
  - This method is refutationally complete, if the local resolution is *complete for consequence finding*

Module Graph (PBR)



Graph



Tree

Completeness for Consequence Finding

An inference method *R* is complete for consequence finding (cfcf) iff:

- For every clause φ that is a non-tautologous logical consequence of the KB, *R* derives a clause ψ that subsumes φ. ($\exists\theta, \psi\theta \subseteq \phi$)

FOL resolution is cfcf, but not on DL

Adapting PBR to DL

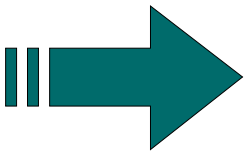
- Use special skolemization and resolution methods proposed for KAON2

Questions:

- is this resolution method cfcf?
- or modifiable to be cfcf?
- or are there other applicable cfcf resolution methods?

Completeness for Consequence Finding

- Requires prolific derivation which is in general the opposite of efficiency. (a lot more derivations than necessary for refutational completeness)
- Efficient resolution methods do not have this property.



completeness for consequence finding (cfcf) is not an appropriate property for an efficient distributed reasoner.