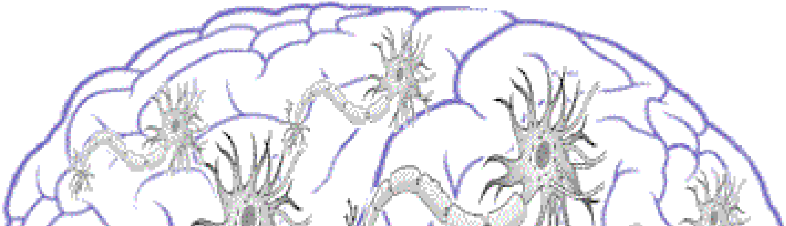
Distributed Resolution for Expressive Ontology Networks

Anne Schlicht, Heiner Stuckenschmidt

UNIVERSITÄT MANNHEIM

October 25, 2009



Problem

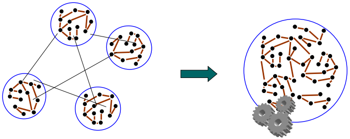
Reasoning on Interlinked Description Logic Ontologies

Common Approach: Merge

Problem

Reasoning on Interlinked Description Logic Ontologies

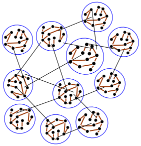
Common Approach: Merge



Problem

Reasoning on Interlinked Description Logic Ontologies

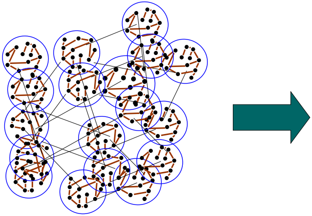
Common Approach: Merge



Problem

Reasoning on Interlinked Description Logic Ontologies

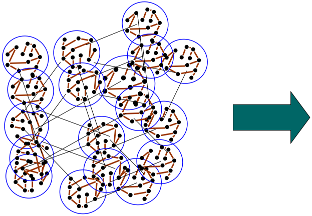
Common Approach: Merge



Problem

Reasoning on Interlinked Description Logic Ontologies

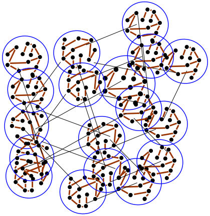
Common Approach: Merge



Problem

Reasoning on Interlinked Description Logic Ontologies

Common Approach: Merge



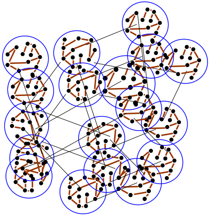
Problem:

- Load ontology

Problem

Reasoning on Interlinked Description Logic Ontologies

Common Approach: Merge



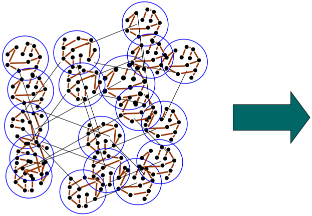
Problem:

- Load ontology
- Query runtime

Problem

Reasoning on Interlinked Description Logic Ontologies

Common Approach: Merge



Problem:

- Load ontology
- Query runtime

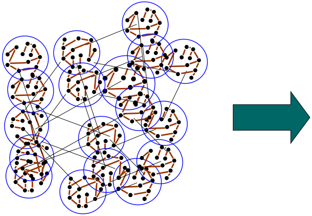
Standard Solution:

- Take advantage of specific structure

Problem

Reasoning on Interlinked Description Logic Ontologies

Common Approach: Merge



Problem:

- Load ontology
- Query runtime

Standard Solution:

- Take advantage of specific structure
- Use incomplete reasoning methods

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Related Work

Approaches

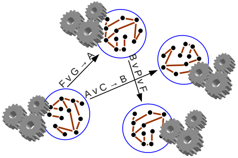
- DRAGO (C-OWL)
- MARVIN (RDF)
- Partition Based Reasoning (FOL)
- Distributed EL (Polynomial DL subset)
- Distributed A-box

Limitations

- Severe limitations on links (no subsumption between ontologies)
- FOL approaches not efficient for DL (FOL is not decidable)
- Limited expressivity
- Restrictions on distribution

Reasoning on Large DL Ontologies

Idea



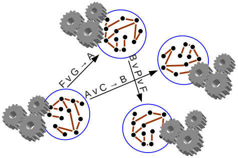
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



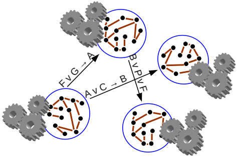
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



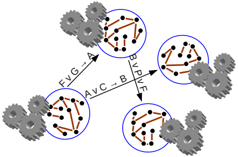
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



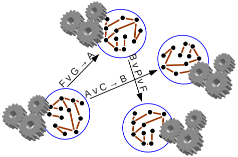
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



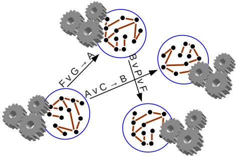
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



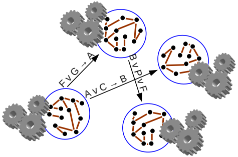
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



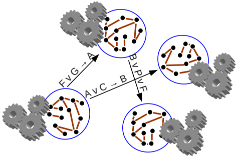
- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Reasoning on Large DL Ontologies

Idea



- Keep ontologies distributed
- Distribute computation load across several reasoners
- Every reasoner processes a small local set of axioms
- Axioms are sent to other reasoners if necessary

Goal

- Completeness and termination for expressive DL
- No restriction on link axioms
- No restriction on distribution
- Performance

Distributed Resolution Reasoning

Outline

- Local reasoning method
- Communication strategy
- Experiments

Distributed Resolution Reasoning

Outline

⇒ Local reasoning method
- Communication strategy
- Experiments

Distributed Reasoning

Standard Description Logic Reasoning: Tableaux

- Extensively investigated and optimized for DL
- No efficient distribution known

Alternative: Resolution

- Extensively investigated for first order logic.
- Succesfully applied to DL
- Distributable

Distributed Reasoning

Standard Description Logic Reasoning: Tableaux

- Extensively investigated and optimized for DL
- No efficient distribution known

Alternative: Resolution

- Extensively investigated for first order logic.
- Succesfully applied to DL
- Distributable

Distributed Reasoning

Standard Description Logic Reasoning: Tableaux

- Extensively investigated and optimized for DL
- No efficient distribution known

Alternative: Resolution

- Extensively investigated for first order logic.
- Succesfully applied to DL
- Distributable

Distributed Reasoning

Standard Description Logic Reasoning: Tableaux

- Extensively investigated and optimized for DL
- No efficient distribution known

Alternative: Resolution

- Extensively investigated for first order logic.
- Succesfully applied to DL
- Distributable

Distributed Reasoning

Standard Description Logic Reasoning: Tableaux

- Extensively investigated and optimized for DL
- No efficient distribution known

Alternative: Resolution

- Extensively investigated for first order logic.
- Succesfully applied to DL
- Distributable

Preliminaries: The description logic $\mathcal{ALCHIQ}$

Translating DL Axioms to clauses

$A \sqsubseteq B$	$\neg A(x) \vee B(x)$
$A \sqsubseteq B \sqcap C$	$\neg A(x) \vee B(x),\ \neg A(x) \vee C(x)$
$A \sqsubseteq B \sqcup C$	$\neg A(x) \vee B(x) \vee C(x)$
$A \sqsubseteq \exists r.B$	$\neg A(x) \vee r(x, f(x)),\ \neg A(x) \vee B(f(x))$
$A \sqsubseteq \forall r.B$	$\neg A(x) \vee \neg r(x, y) \vee B(y)$
$A \sqsubseteq \exists_{\leq n} r.B$	$\neg A(x) \vee \neg r(x, y_i) \vee y_i = y_j \vee \neg B(y_i)$ $i = 1..n{+}1 \quad j = 1..i{-}1$
$A \sqsubseteq \exists_{\geq n} r.B$	$\neg A(x) \vee r(x, f_i(x)),\ \neg A(x) \vee f_i(x) \neq f_j(x),\ \neg A(x) \vee B(f_i(x))$ $i = 1..n \quad j = 1..i{-}1$
$r \sqsubseteq s$	$\neg r(x, y) \vee s(x, y)$
$r \equiv Inv(s)$	$\neg r(x, y) \vee s(y, x),\ \neg s(x, y) \vee r(y, x)$

Preliminaries: The description logic $\mathcal{ALCHIQ}$

Translating DL Axioms to clauses

$A \sqsubseteq B$	$\neg A(x) \vee B(x)$
$A \sqsubseteq B \sqcap C$	$\neg A(x) \vee B(x),\ \neg A(x) \vee C(x)$
$A \sqsubseteq B \sqcup C$	$\neg A(x) \vee B(x) \vee C(x)$
$A \sqsubseteq \exists r.B$	$\neg A(x) \vee r(x, f(x)),\ \neg A(x) \vee B(f(x))$
$A \sqsubseteq \forall r.B$	$\neg A(x) \vee \neg r(x, y) \vee B(y)$
$A \sqsubseteq \exists_{\leq n} r.B$	$\neg A(x) \vee \neg r(x, y_i) \vee y_i = y_j \vee \neg B(y_i)$
$A \sqsubseteq \exists_{\geq n} r.B$	$\neg A(x) \vee r(x, f_i(x)),\ \neg A(x) \vee f_i(x) \neq f_j(x),\ \neg A(x) \vee B(f_i(x))$ $i = 1..n \quad j = 1..i-1$
$r \sqsubseteq s$	$\neg r(x, y) \vee s(x, y)$
$r \equiv Inv(s)$	$\neg r(x, y) \vee s(y, x),\ \neg s(x, y) \vee r(y, x)$

$$\boxed{GlobalCompany \sqsubseteq \exists_{\geq 2} situatedIn.Continent}$$

Resolution

Ordered Resolution

$$\frac{C(x) \vee A(x) \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))} \quad x \rightarrow f(y)$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee f(x) = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee \overset{x \rightarrow f(y)}{A(x)} \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))}$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee f(x) = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee A(x) \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))} \quad x \longrightarrow f(y)$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee f(x) = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee \overbrace{A(x)}^{x \rightarrow f(y)} \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))}$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee f(x) = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee \overset{x \to f(y)}{A(x)} \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))}$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee \overset{x \to y}{f(x)} = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee \overset{x \,\to\, f(y)}{A(x)} \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))}$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee \overset{x \,\to\, y}{f(x)} = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Ordered Resolution

$$\frac{C(x) \vee \overset{x \rightarrow f(y)}{A(x)} \qquad D(y, f(y)) \vee \neg A(f(y))}{C(f(y)) \vee D(y, f(y))}$$

- Literals are ordered (based on symbol precedence), unified literals have to be maximal
- Complete and terminates for $\mathcal{ALCHI}$
- Problem: no support for equalities (cardinality restrictions)

Basic Superposition (Bachmair et al. 1995)

$$\frac{C(x) \vee \overset{x \rightarrow y}{f(x)} = g(x) \qquad D(y, f(y)) \vee P(f(y))}{C(y) \vee D(y, f(y)) \vee P(g(y))}$$

- Extension of ordered resolution to deal with equalities
- Complete, terminates for expressive DL $\mathcal{ALCHIQ}$ (Hustadt et al. 2007)

Resolution

Basic Superposition

$$\text{Positive superposition} \quad \frac{(C \vee s \approx t) \cdot \rho \quad D \vee (w \approx v) \cdot \rho}{(C \vee D \vee w[t]_p \approx v) \cdot \theta}$$

where

1. σ is the most general unifier of $s\rho$ and $w\rho|_p$ and $\theta = \rho\sigma$
2. $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$
3. in $(C \vee s \approx t) \cdot \theta$ nothing is selected and $(s \approx t) \cdot \theta$ is strictly maximal
4. in $D \vee (w \approx v) \cdot \theta$ nothing is selected and $(w \approx v) \cdot \theta$ is strictly maximal
5. $w|_p$ is not a variable.
6. $s\theta \approx t\theta \not\succeq w\theta \approx v\theta$

$$\text{Negative superposition} \quad \frac{(C \vee s \approx t) \cdot \rho \quad D \vee (w \not\approx v) \cdot \rho}{(C \vee D \vee w[t]_p \not\approx v) \cdot \theta}$$

where

1. σ is the most general unifier of $s\rho$ and $w\rho|_p$ and $\theta = \rho\sigma$
2. $t\theta \not\succeq s\theta$ and $v\theta \not\succeq w\theta$
3. in $(C \vee s \approx t) \cdot \theta$ nothing is selected and $(s \approx t) \cdot \theta$ is strictly maximal
4. $(w \not\approx v) \cdot \theta$ is selected or maximal and no other literal is selected in $D \vee (w \not\approx v) \cdot \theta$
5. $w|_p$ is not a variable.

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

$$\begin{array}{cc} \neg P(..) & P(..) \\ \neg f(..) = g(..) & f(..) = h(..) \\ \neg P(f(..)) & f(..) = g(..) \end{array}$$

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

$$\begin{array}{cc} \neg P(..) & P(..) \\ \neg f(..) = g(..) & f(..) = h(..) \\ \neg P(f(..)) & f(..) = g(..) \\ \neg P(f(g(..))) & g(..) = h(..) \end{array}$$

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

$$
\begin{array}{cc}
\neg P(..) & P(..) \\
\neg f(..) = g(..) & f(..) = h(..) \\
\neg P(f(..)) & f(..) = g(..) \\
\cancel{\neg P(f(g(..)))} & \cancel{g(..) = h(..)}
\end{array}
$$

Resolution

Basic Superposition: Observations on $\mathcal{ALCHIQ}$ Clauses

- For all rules with more than one premise, the resolved literals have to be stricly maximal.
- Strictly maximal equation literals have comparable arguments.
- There are only 3 types of unifications in BS inferences:

$$
\begin{array}{cc}
\neg P(..) & P(..) \\
\neg f(..) = g(..) & f(..) = h(..) \\
\neg P(f(..)) & f(..) = g(..) \\
\cancel{\neg P(f(g(..)))} & \cancel{g(..) = h(..)}
\end{array}
$$

$\Rightarrow$ Communication Strategy Based on Symbol Allocation

Distributed Resolution Reasoning

Outline

- $\Rightarrow$ Local reasoning method
- Communication strategy
- Experiments

Distributed Resolution Reasoning

Outline

- Local reasoning method
- ⇒ Communication strategy
- Experiments

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

Example

$C(x) \vee D(f(x)) \vee A(x)$

precedence

$f > g > A > B > C > D$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Example

$C(x) \vee D(f(x)) \vee A(x)$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1and 2

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$$\boxed{f > g > A > B > C > D}$$

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1and 2

$C(x) \vee f(x) = g(x)$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1and 2

$C(x) \vee f(x) = g(x)$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1and 2
$C(x) \vee f(x) = g(x)$

Communication Strategy

Allocation of Clauses to Reasoners

Based on allocation and precedence of symbols

1. Pick the maximal literal of the clause
2. Pick the top predicate of that literal
3. Allocate the clause to the reasoner that the predicate is allocated to

2b Pick the top function symbol of that literal

3b Allocate the clause to the reasoner that the symbol is allocated to

allocation

symbol	reasoner
A	1
B	2
C	2
D	1
f	2
g	1

precedence

$f > g > A > B > C > D$

Example

$C(x) \vee D(f(x)) \vee A(x) \rightarrow$ reasoner 1and 2
$C(x) \vee f(x) = g(x) \rightarrow$ reasoner 2

Method

Distributed Basic Superposition

Calculus: Basic Superposition

- Saturating the local clause set in every reasoner

Communication strategy: Based on maximal literal

- Every reasoner is responsible for a subset of symbols
- Every input/derived clause is allocated to
  1. the reasoner responsible for the top predicate of the literal (if predicate literal)
  2. the reasoner responsible for the top function sympbol of the literal (if literal contains function)

Distributed Resolution Reasoning

Outline

- Local reasoning method
- ⇒ Communication strategy
- Experiments

Distributed Resolution Reasoning

Outline

- Local reasoning method
- Communication strategy

⇒ Experiments

Experiments

Implementation

- Based on the first order reasoner SPASS
- Communication via TCP connections (completely connected at startup)
- Reasoner waits for new clauses when saturated locally
- The system terminates when one reasoner finds a proof or all are saturated
- Clauses are only send when the destination reasoner is idle

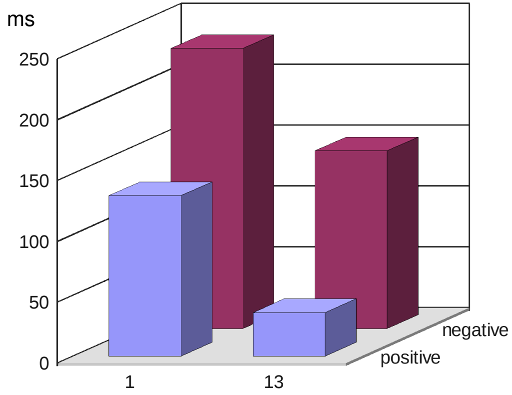
Experiments

Dataset

- SWEET (Semantic Web for Earth and Environmental Terminology) dataset published by the NASA Jet Propulsion Laboratory
- chemical ontology (chem.owl) and the ontologies that are directly or indirectly imported by chem.
- 13 ontologies liked by 34 import statements.
- The ontology network describes 480 classes and 99 individuals.
- Datatype properties replaced by object properties, nominals replaced by common concepts
- Expressivity: $\mathcal{SHIN}$

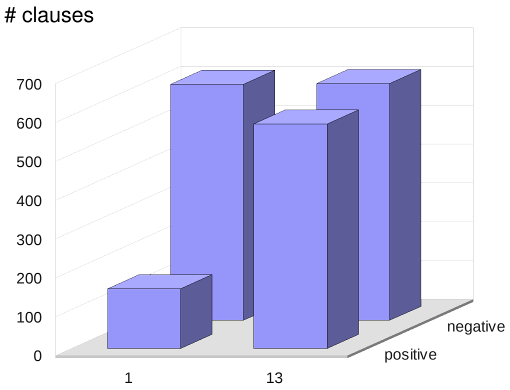
Experiments

Runtime of Positive and Negative Queries



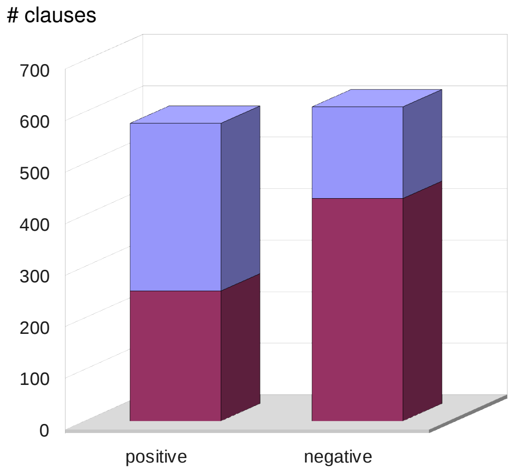
Experiments

Number of Derived Clauses



Experiments

Number of Propagated Clauses

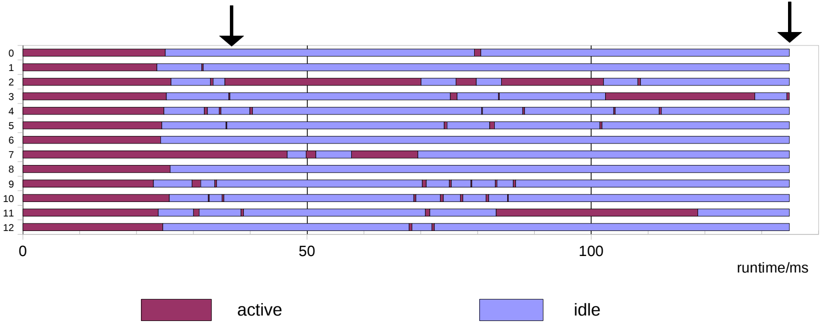


Experiments

Parallel Computation

positive queries: 60% busy

negative queries: 20% busy



Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Conclusion

Distributed Resolution for DL Ontologies

- The approach is complete and terminates for $\mathcal{ALCHIQ}$ ontologies
- No restriction on link axioms
- No strong restriction on distribution
- First experiments show that runtime speedup from parallel computation trades off the communication overhead.

Next Steps

- Connection on Demand
- Dynamic Allocation

Discussion

Thank You!

Questions?

Allocating Symbols to Reasoners

Straight Forward Allocation

In ontology networks, the ontologies are usually identified by different namespaces
$\Rightarrow$ Allocation of namespaces to reasoners defines an allocation of symbols.

Dependency Based Allocation

1. Create dependency graph from axioms or clauses, each node represents a symbol
2. Graph partitioning (e.g. minimal edge cut)
3. Allocate every part (node set) to a reasoner

Communication Based Allocation

1. Allocate every symbol to a different reasoner
2. Create dependency graph based on communication
3. Graph partitioning, ...

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

Unavoidable Redundancy

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

Unavoidable Redundancy

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

Unavoidable Redundancy

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected

Redundancy

Avoiding Redundant Inferences and Clauses

The communication strategy occassionally allocates a clause to two different reasoners.
(i.e. clauses with maximal literal $P(f(x))$ and different allocation of P and f)

- A clause is never allocated to more than two reasoners
- Duplication of clauses does not duplicate inferences

Unavoidable Redundancy

- Application of reduction rules is restricted by distributing the clause sets.
- e.g. $a \vee b \vee c$ is redundant if $a \vee b$ given
- if the two clauses are allocated to different reasoners, the redundancy is not detected