OpenRuleBench: An Analysis of the Performance of Rule Engines

Senlin Liang, Paul Fodor, Hui Wan, Michael Kifer

Department of Computer Science
State University of New York at Stony Brook

sliang@cs.sunysb.edu

April 24, 2009

Motivations

- Semantic web is HOT!
- Rules are powerful in processing semantic information.
- How rule technologies perform on the Web scale?
- Previous comparisons were superficial [Bishop 2008, Sure 2002].

[Bishop 2008] B. Bishop and F. Fischer.
Iris - Integrated Rule Inference System.
International Workshop on Advancing Reasoning on the Web: Scalability and Commonsense (ARea 2008), June 2008.

[Sure 2002] Y Sure, S Staab, and J Angele.
OntoEdit: Guiding Ontology Development by Methodology and Inferencing.
1st International Conf. on Ontologies, Databases, and Applications of Semantics, 2002.

Outline

OpenRuleBench

- Five technologies:
  - Prolog-based
  - Deductive database
  - Production rules
  - Triple engines
  - General knowledge base
- Twelve systems:
  - XSB, Yap, SWI
  - DLV, IRIS, Ontobroker
  - Drools, Jess, Prova
  - Jena, SwiftOWLIM
  - CYC

OpenRuleBench

- Five packages:
  - Large joins
  - Datalog recursion
  - Default negation
  - Dynamic indexing
  - Database interfaces
- Open community resource:
  - Programs
  - Scripts
  - Results
  - Manuals

Outline

Test Principles & System Capabilities

- Test principles:
  - Loading vs. inference times
  - Using the best settings for each system
- System capabilities:
  - Predicate arity constraints
  - Negation handling
  - Automatic optimizations:
    Cost-based optimizations, Subgoal reordering, Query filtering, Magic Sets, Indexing

Large Joins

- Join1 and Join2:

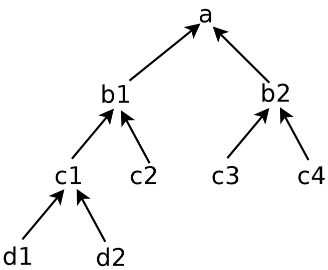

Fig.1 Join1

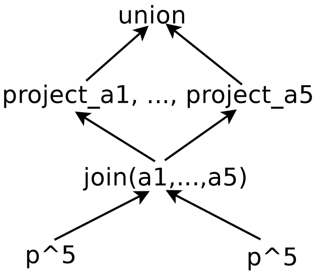

Fig.2 Join2

Large Joins Cont'd

- 3 queries from LUBM: [Guo 2005].
- Mondial: a geographical database.
- DBLP: a publication database.

```
query(Id,T,A,Y,M) :- att(Id,title,T), att(Id,year,Y),
                     att(Id,author,A), att(Id,month,M).
```

[Guo 2005] Y. Guo, Z. Pan, and J. Heflin.
LUBM: A Benchmark for OWL Knowledge Base Systems.
Journal of Web Semantics, 2005.

Datalog Recursion

- Transitive closure:
```
ancestor(X,Y) :- parent(X,Y).
ancestor(X,Y) :- parent(X,Z),
                 ancestor(Z,Y).
```
  queries: `ancestor(X,Y)`, `ancestor(1,X)`, and `ancestor(X,1)`.
- Same generation:
```
sg(X,Y) :- sib(X,Y).
sg(X,Y) :- par(X,Z), sg(Z,Z1), par(Y,Z1).
```
  queries: `sg(X,Y)`, `sg(1,X)`, and `sg(X,1)`.
- WordNet tests: hypernyms, hyponyms, etc.
- Wine ontology: many mutually recursive rules.

Default Negation

- Modified same generation:
```
non_sg(X,Y) :- ancestor(X,Y).
non_sg(X,Y) :- ancestor(Y,X).
sg2(X,Y) :- sg(X,Y), not non_sg(X,Y).
```
- Win-not-win:
```
win(X) :- move(X,Y), not win(Y).
```

Default Negation Cont'd

- A complex program from [Balbin 2008].

```
fb(X) :- magicfb(X), d(X), not ab(X),
         h(X,Y), ab(Y).
ab(X) :- magicab(X), g(X).
ab(X) :- magicab(X), b(X,Y), ab(Y).
magicab(Y) :- magicab(X), b(X,Y).
magicab(Y) :- magicfb(X), d(X), not ab(X),
              h(X,Y).
magicab(X) :- magicfb(X), d(X).
```

[Balbin 2008] I. Balbin, G. S. Port, K. Ramamohanarao, and K. Meenakshi.
Efficient bottom-up computation of queries on stratified databases.
J. Log. Program., 2008.

Miscellaneous Tests

- 16-Puzzle
- N-Queens
- Bitrev
- Dynamic indexing
- Database interfaces

Outline

Results Summary

No system was the best for all the tests.

- Three overall winners: Yap, XSB, and Ontobroker.
- DLV was also close.

No optimization was the best for all tests.

Promising technologies:

- Tabling Prolog technology: XSB and Yap.
- Deductive database technology: Ontobroker and DLV.

Scalability and performance issues:

- Indexing.
- Memory management.
- Query optimization.

Results Summary

No system was the best for all the tests.

- Three overall winners: Yap, XSB, and Ontobroker.
- DLV was also close.

No optimization was the best for all tests.

Promising technologies:

- Tabling Prolog technology: XSB and Yap.
- Deductive database technology: Ontobroker and DLV.

Scalability and performance issues:

- Indexing.
- Memory management.
- Query optimization.

Results Summary

No system was the best for all the tests.

- Three overall winners: Yap, XSB, and Ontobroker.
- DLV was also close.

No optimization was the best for all tests.

Promising technologies:

- Tabling Prolog technology: XSB and Yap.
- Deductive database technology: Ontobroker and DLV.

Scalability and performance issues:

- Indexing.
- Memory management.
- Query optimization.

Results Summary

No system was the best for all the tests.

- Three overall winners: Yap, XSB, and Ontobroker.
- DLV was also close.

No optimization was the best for all tests.

Promising technologies:

- Tabling Prolog technology: XSB and Yap.
- Deductive database technology: Ontobroker and DLV.

Scalability and performance issues:

- Indexing.
- Memory management.
- Query optimization.

The Effect of Indexing and Tabling

system	XSB	Yap	Ontobroker	DLV
time	0.004	0.037	0.042	1.045

Table: Mondial (Fully Optimized)

Case study: XSB

- Fully optimized: tabling and manual indexing.
- NO tabling: 1.713 seconds. (400 times slower!)
- NO tabling or manual indexing: 129.89 seconds (30,000 times slower!)

The Effect of Join Strategies

—Indexed-Nested-Loop vs. Sort-Merge

query	a(X,Y)		b1(X,Y)		b2(X,Y)	
size	50K	250K	50K	250K	50K	250K
ontobroker	4.089	28.385	0.213	4.806	0.019	0.168
xsb	12.774	timeout	0.122	14.920	0.013	0.269
yap	10.534	timeout	0.109	12.123	0.013	0.269
dlv	85.459	838.781	7.177	60.239	0.820	9.392

Table: Join1, no query bindings

- Sort-merge (Ontobroker): scales better.
- Indexed-nested-loop (XSB and Yap): low overhead.

Join1 & Join2

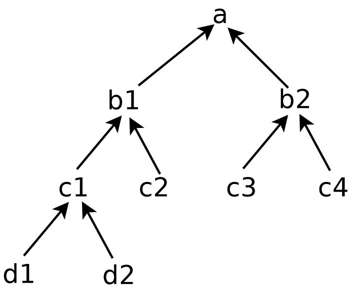

Fig.1 Join1

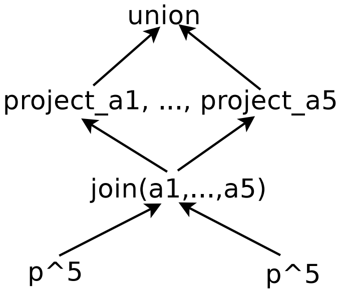

Fig.2 Join2

The Effect of Subgoal Reordering

query	a(1,Y)		b1(1,Y)		b2(1,Y)	
size	50K	250K	50K	250K	50K	250K
ontobroker	0.035	0.038	0.013	0.051	0.070	0.012
xsb	0.013	35.990	0.000	0.016	0.000	0.001
yap	0.021	30.233	0.007	0.050	0.004	0.025
dlv	0.287	6.014	0.014	0.112	0.008	0.066

Table: Join1, 1st argument bound.

- Subgoal reordering (Ontobroker): scales better, but has initial overhead.

Cartesian Product

system	yap	xsb	ontobroker	dlv
time	2.087	2.092	11.935	44.692

Table: Times for Join2.

- Database technology (Ontobroker) cannot do much for Cartesian products.
- The tabled SLG-WAM (XSB and Yap) has low overhead.

Naive Select-Join

system	ontobroker	xsb	yap	drools	dlv
time	1.602	1.752	2.447	0.186	2.201

Table: Times for DBLP

- `query(Id,T,A,Y,M) :- att(Id,title,T), att(Id,year,Y), att(Id,author,A), att(Id,month,M).`
- Drools: select, build indexing, and join.

Datalog Recursion

size	50K	50K	500K	500K
cyclic data	no	yes	no	yes
ontobroker	6.129	19.145	49.722	182.633
dlv	19.655	73.837	148.740	900.773
xsb	2.725	7.081	35.036	88.028
yap	2.066	13.026	33.128	82.900

Table: Transitive closure, no query bindings.

- Transitive closure: XSB and Yap perform the best.

Datalog Recursion Cont'd

No obvious overall winner.

- Same generation: Ontobroker performs and scales better.
- Wordnet: Yap performs significantly better than others.
- Wine ontology: XSB and Ontobroker perform better.

Default Negation

test	win-not-win			modified same gen	
size	100K	500K	2M	6K	24K
ontobroker	1.327	9.988	timeout	14.883	24.963
dlv	0.691	3.554	15.224	36.827	444.873
xsb	0.231	1.218	5.081	7.265	90.928
yap	0.103	0.654	2.866	3.339	44.605

Table: Locally- and predicate-stratified negation.

test	win-not-win			[Balbin 2008]	
size	50K	250K	1M	24K	504K
ontobroker	0.419	3.754	17.237	0.236	8.409
dlv	0.344	1.879	8.361	0.189	2.043
xsb	0.339	1.416	5.647	1.381	1.663

Table: Locally non-stratified rule sets.

- Top-down SLG-resolution (XSB and Yap) performs and scales better than bottom-up alternating fixed point computation (Ontobroker and DLV).

Outline

Conclusions

- Open community resource: OpenRuleBench.
- Identified two promising rule technologies:
  - Tabling Prolog
  - Deductive database
- Identified several important issues:
  - Indexing
  - Memory management
  - Query optimization
- Future work: more systems and tests.

Thank you!