First-Order Deduction for Large Knowledge Bases

Stephan Schulz
schulz@eprover.org

Interest in Large Theories

2007 ESARLT at CADE-21 in Bremen

- Ontologies (SUMO)
- Common Sense Reasoning (CYC)
- Mizar

2008 First CASC LTB at IJCAR in Sydney

- MaLARea, SInE, Vampire LTB

2009 CASC LTB at CADE-22 in Montreal

2010 CASC LTB in IJCAR in Edinburgh

- Rules change every year!

Interest in Large Theories

2007 ESARLT at CADE-21 in Bremen

- Ontologies (SUMO)
- Common Sense Reasoning (CYC)
- Mizar

2008 First CASC LTB at IJCAR in Sydney

- MaLARea, SInE, Vampire LTB

2009 CASC LTB at CADE-22 in Montreal

2010 CASC LTB in IJCAR in Edinburgh

- Rules change every year!

Interest in Large Theories

2007 ESARLT at CADE-21 in Bremen

- Ontologies (SUMO)
- Common Sense Reasoning (CYC)
- Mizar

2008 First CASC LTB at IJCAR in Sydney

- MaLARea, SInE, Vampire LTB

2009 CASC LTB at CADE-22 in Montreal

2010 CASC LTB in IJCAR in Edinburgh

- Rules change every year!

Interest in Large Theories

2007 ESARLT at CADE-21 in Bremen

- Ontologies (SUMO)
- Common Sense Reasoning (CYC)
- Mizar

2008 First CASC LTB at IJCAR in Sydney

- MaLARea, SInE, Vampire LTB

2009 CASC LTB at CADE-22 in Montreal

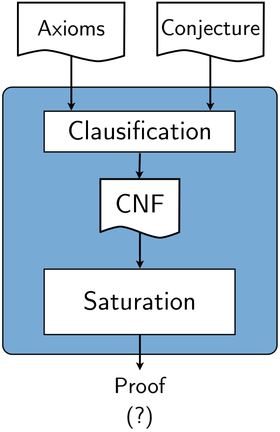
2010 CASC LTB in IJCAR in Edinburgh

- Rules change every year!

Saturation-Based Theorem Proving

Leading approach for first-order reasoning

- Proof by refutation
- Conjecture and axioms converted to flat CNF
- Saturation with destructive simplification
- Goal: Empty clause
- Powerful calculi
- Highly advanced data structures
- Decent search heuristics



A Classical Example: Reasoning in Rings

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

A ring with $x^3 = x$ is commutative

- 9 formulas
- 3 KB of text (with comments)

Proof search

- ~200000 steps
- ~50 MB of text

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

A ring with $x^3 = x$ is commutative

- 9 formulas
- 3 KB of text (with comments)

Proof search

- ~200000 steps
- ~50 MB of text

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

A ring with $x^3 = x$ is commutative

- 9 formulas
- 3 KB of text (with comments)

Proof search

- ~200000 steps
- ~50 MB of text

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

A ring with $x^3 = x$ is commutative

- 9 formulas
- 3 KB of text (with comments)

Proof search

- ~200000 steps
- ~50 MB of text

The Matter of Scaling

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

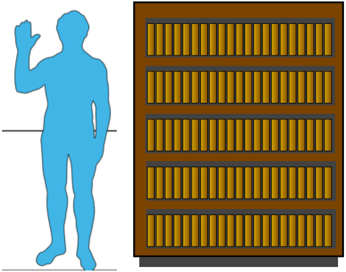
RNG009-5: Peterson/Stickel, 1981

```
%----Right identity and inverse
cnf(right_identity,axiom,
    ( add(X,additive_identity) = X )).

cnf(right_additive_inverse,axiom,
    ( add(X,additive_inverse(X)) = additive_identity )).

%----Distributive property of product over sum
cnf(distribute1,axiom,
    ( multiply(X,add(Y,Z)) = add(multiply(X,Y),multiply(X,Z)) )).

cnf(distribute2,axiom,
    ( multiply(add(X,Y),Z) = add(multiply(X,Z),multiply(Y,Z)) )).

%----Associativity of addition
cnf(associative_addition,axiom,
    ( add(add(X,Y),Z) = add(X,add(Y,Z)) )).

%----Commutativity of addition
cnf(commutative_addition,axiom,
    ( add(X,Y) = add(Y,X) )).

%----Associativity of product
cnf(associative_multiplication,axiom,
    ( multiply(multiply(X,Y),Z) = multiply(X,multiply(Y,Z)) )).

cnf(x_cubed_is_x,hypothesis,
    ( multiply(X,multiply(X,X)) = X )).

cnf(prove_commutativity,negated_conjecture,
    (  multiply(a,b) != multiply(b,a) )).
```

RNG009-5: Peterson/Stickel, 1981

RNG009-5: Peterson/Stickel, 1981

RNG009-5: Peterson/Stickel, 1981

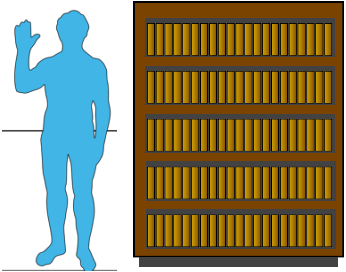


RNG009-5: Peterson/Stickel, 1981

CSR066-6: Smith et al, 2007

OpenCYC "Common sense" reasoning

- 3341990 formulas
- 480 MB of text (with few comments)



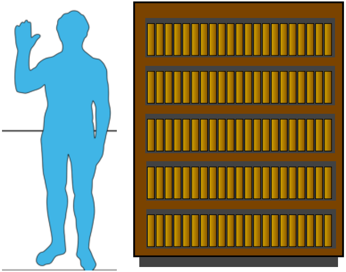
CSR066-6: Smith et all, 2007

OpenCYC "Common sense" reasoning

- 3341990 formulas
- 480 MB of text (with few comments)

Proof search

?



CSR066-6: Smith et all, 2007

Bad Ideas for Mega-Axiom Problems

“First simplify the whole specification”

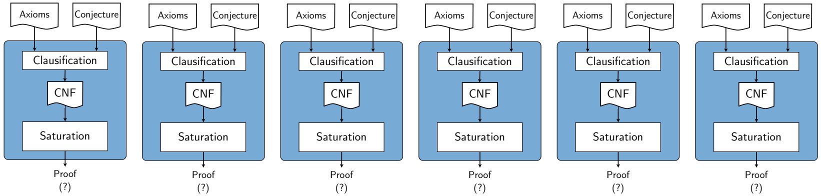
“First consider *all* axioms”

Quadratic or cubic algorithms

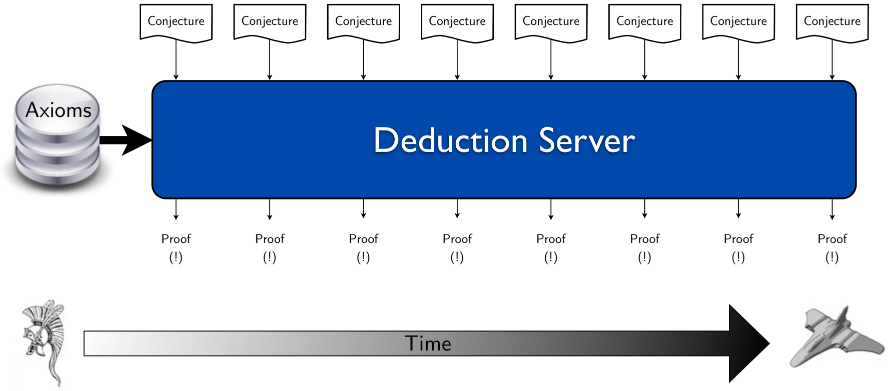
Undirected saturation

Reload the specification for each new conjecture

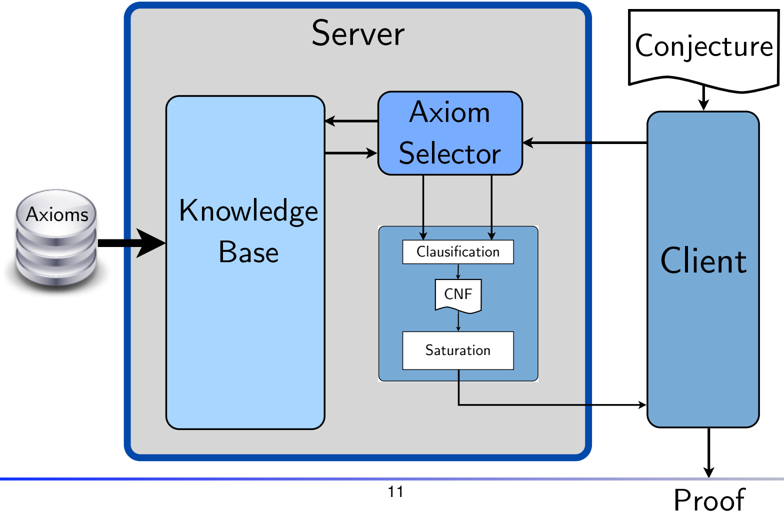
Multiple Queries the Hard Way



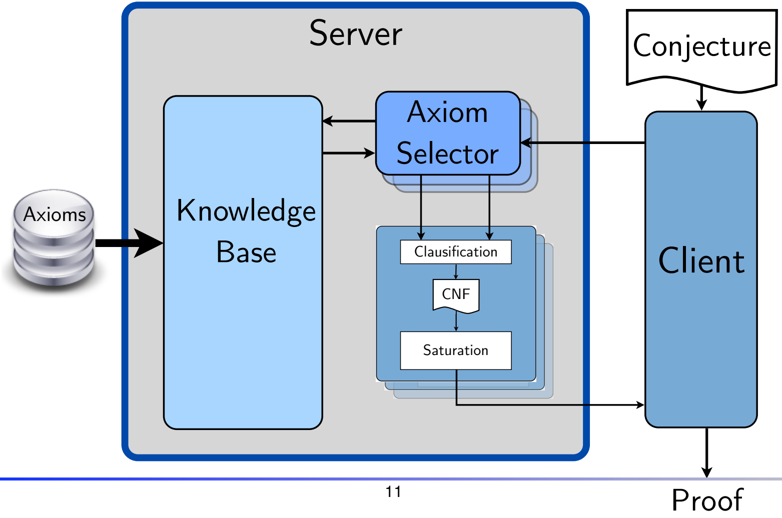
Deduction as a Service



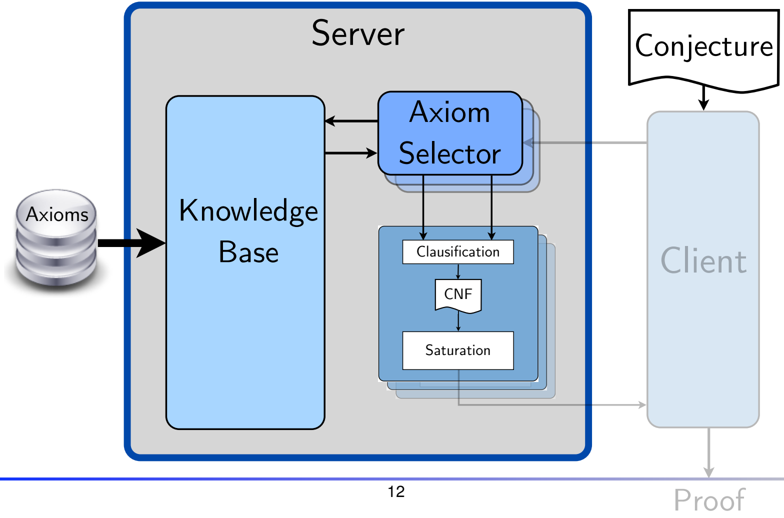
Deduction Server Architecture



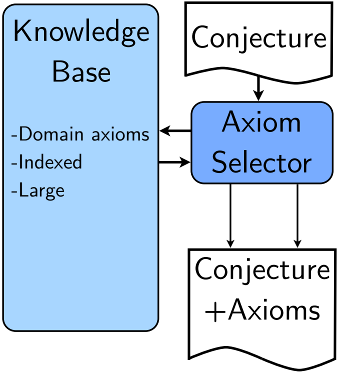
Deduction Server Architecture



Current State



Axiom Selection



Input

- Conjecture
- Optionally hypotheses

Output

- "Relevant" axioms
- Conjecture

Constraints

- Backtrackable
- Fast (enough)

Relevancy Analysis

Algorithm:

- Start with the conjecture
- All symbols in the conjecture are relevant
- All formulas containing relevant symbols are relevant
- Iterate for n steps or up to fixpoint

Relevancy Analysis

Algorithm:

- Start with the conjecture
- All symbols in the conjecture are relevant
- All formulas containing relevant symbols are relevant
- Iterate for n steps or up to fixpoint

Result: Unsatisfactory

- Some successes
- But: Selection to coarse
- Typically 5-7 iterations to fixpoint
- Often fixpoint is whole specification

Generalized SInE

Basic d-relation:

- Generality measure $g : sig \rightarrow \mathbb{N}$
- Formula C is in d-relation to f if f is $g-$minimal among all symbols in C

Algorithm:

- Start with the conjecture
- All symbols in the conjecture are relevant
- All formulas in d-relation with a relevant symbol are relevant
- Iterate up to fixpoint

Generalized SInE

Basic d-relation:

- Generality measure $g : sig \rightarrow \mathbb{N}$
- Formula C is in d-relation to f if f is $g-$minimal among all symbols in C

Algorithm:

- Start with the conjecture
- All symbols in the conjecture are relevant
- All formulas in d-relation with a relevant symbol are relevant
- Iterate up to fixpoint

Generalization:

- Different generality measures
- Different relaxation criteria for d-relations
- Limit iteration
- Limit number of formulas selected

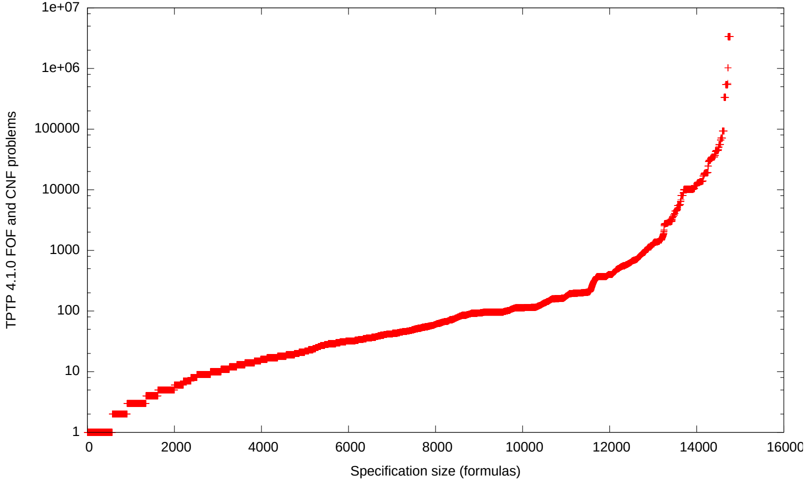
First Results

Platform:

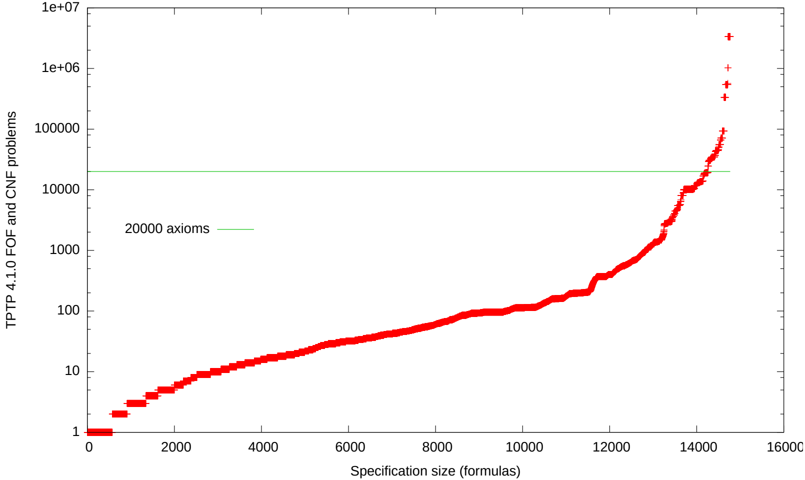
- 800 Mhz Quad-Core AMD Opteron
- 8GBytes RAM

CSR066+6	Plain E	Server Mode
Parsing Problem	109s	Amortised
Axiom Selection	-	35 s
Clausification	240s	0.9 s
Proof	Timeout ($>$1200 s)	348 s

Evaluation: Large TPTP Problems



Evaluation: Large TPTP Problems



Selected Results

TPTP 4.1.0

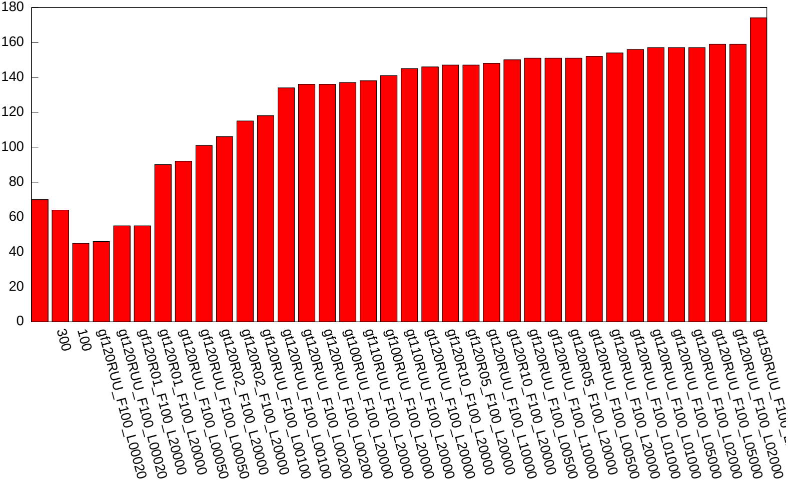
512 problems with $\geq$2000 axioms

Platform:

- 2.2 GHz Intel Xeon
- 16 GBytes

Strategy	Solutions
Full, 300 s	70
Full, 100 s	64
GSInE(f120,RUU,F100,L00020)	45
GSInE(t120,RUU,F100,L00050)	90
GSInE(f120,RUU,F100,L00100)	115
GSInE(t100,RUU,F100,L20000)	136
GSInE(t120,RUU,F100,L20000)	145
GSInE(t120,R10,F100,L20000)	148
GSInE(t120,RUU,F100,L00500)	151
GSInE(f120,RUU,F100,L01000)	154
GSInE(t120,RUU,F100,L05000)	157
GSInE(t150,RUU,F100,L20000)	159

All Results


180
160
140
120
100
80
60
40
20
0
300
100
gf120RUU_F100_L00020
gt120RUU_F100_L00020
gf120R01_F100_L20000
gt120R01_F100_L20000
gt120RUU_F100_L00050
gf120RUU_F100_L00050
gt120R02_F100_L20000
gf120R02_F100_L20000
gf120RUU_F100_L00100
gt120RUU_F100_L00100
gt120RUU_F100_L00200
gf120RUU_F100_L00200
gt100RUU_F100_L20000
gf110RUU_F100_L20000
gf100RUU_F100_L20000
gt110RUU_F100_L20000
gt120RUU_F100_L20000
gf120R10_F100_L20000
gf120R05_F100_L20000
gt120RUU_F100_L10000
gt120R10_F100_L20000
gf120RUU_F100_L00500
gf120RUU_F100_L10000
gt120R05_F100_L20000
gt120RUU_F100_L00500
gf120RUU_F100_L20000
gf120RUU_F100_L01000
gt120RUU_F100_L01000
gf120RUU_F100_L05000
gt120RUU_F100_L02000
gt120RUU_F100_L05000
gf120RUU_F100_L02000

Future Work

Complete server

- TCP connection
- Configurability
- Failure-tolerant parser

Implement client

- Telnet and netcat are sufficient but inconvenient

Evaluate different SInE variants

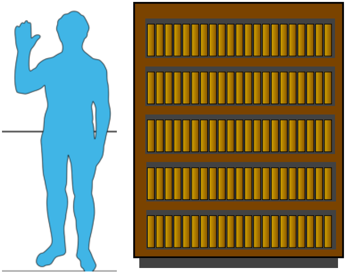
- Which paramters work for which problem clases?
- Which other generality measures are useful?

OpenCYC "Common sense" reasoning

- 3341990 formulas
- 480 MB of text (with few comments)

Proof search

?



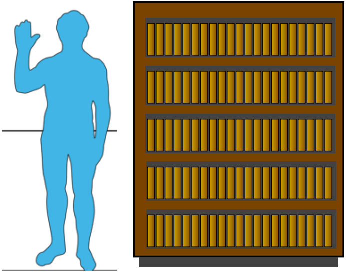
CSR066-6: Smith et all, 2007

OpenCYC "Common sense" reasoning

- 3341990 formulas
- 480 MB of text (with few comments)

Proof search

- ~512000 steps
- ~65 MB of text



CSR066-6: Smith et all, 2007