Extensional Crisis and Proving Identity

Ashutosh Gupta
Laura Kovács
Bernhard Kragl
Andrei Voronkov

Theories + Quantifiers

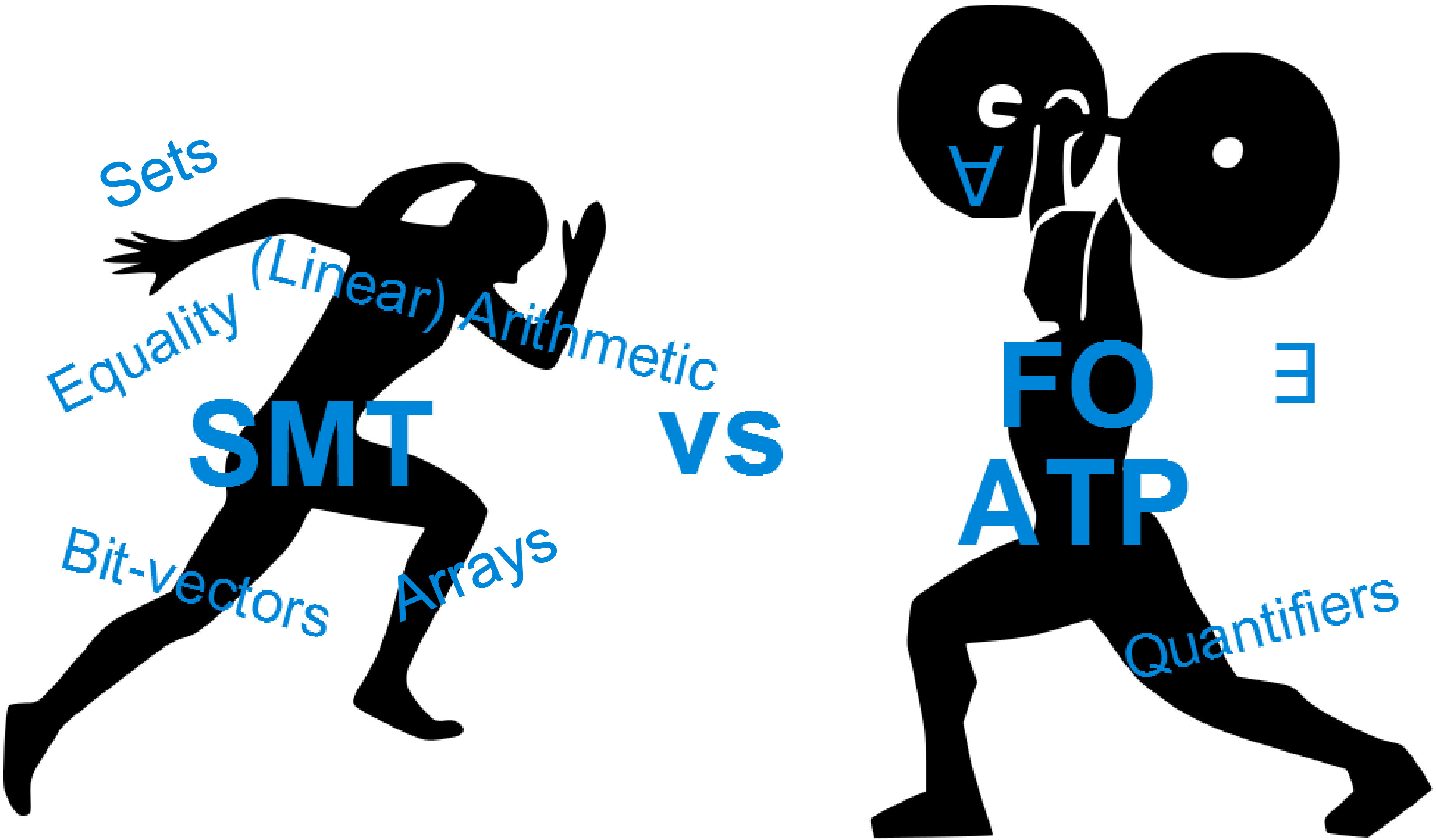
- Applications require theories and quantifiers
- Example: verification of sorting algorithm
  - Sortedness

    $$\forall i \forall j \ (i \leq j \rightarrow OUT[i] \leq OUT[j])$$

  - Value preservation

    $$\forall i \exists j \ (IN[i] = OUT[j])$$
    $$\forall i \exists j \ (OUT[i] = IN[j])$$

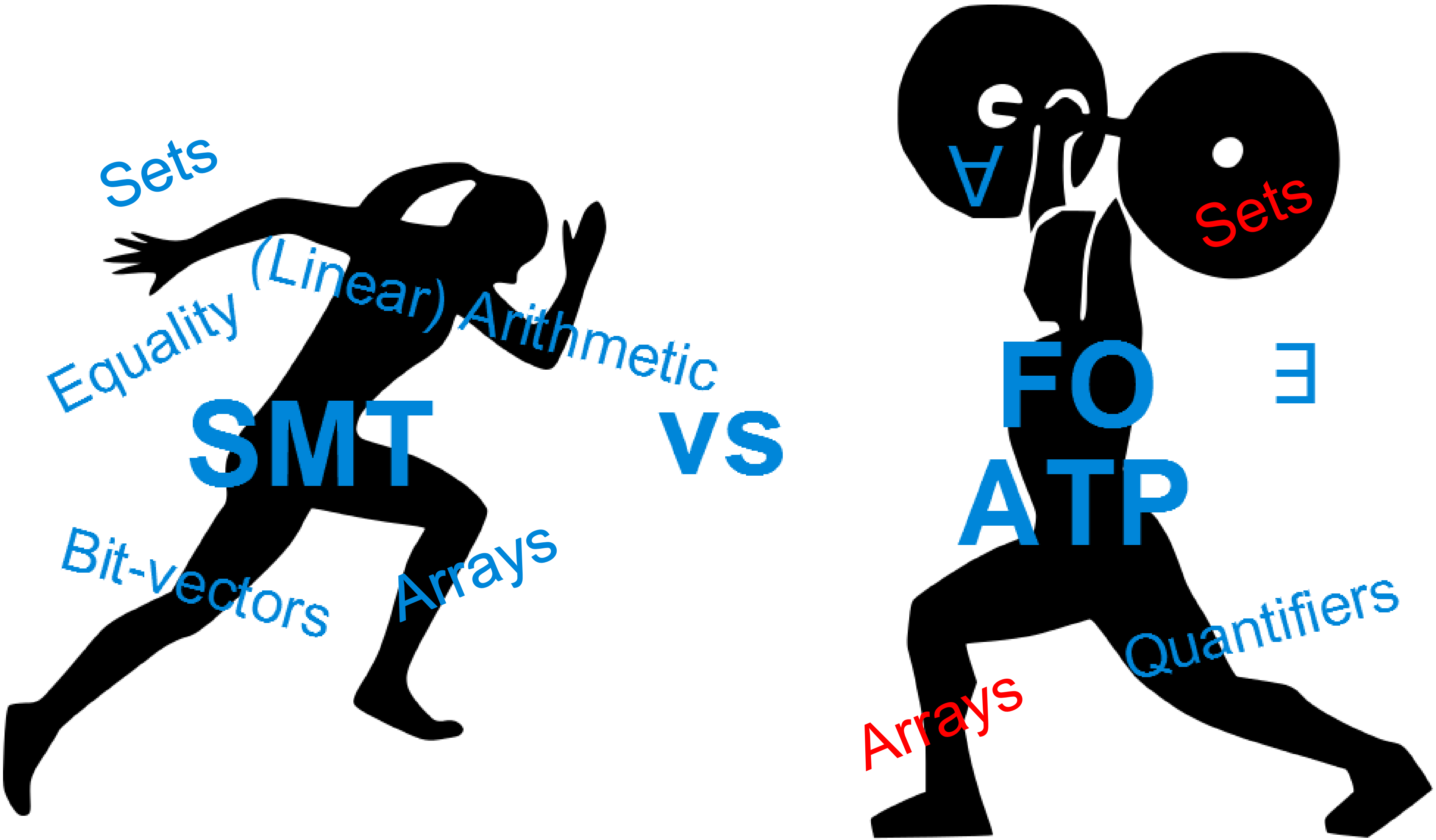
- Major challenge in automated reasoning



Efforts to combine both techniques:

E-matching [DNS,J.ACM'05][R,LPAR'12]
Array fragments [BMS,VMCAI'06][HIV,FoSSaCS'08]
Model based quantifier instantiation [GdM,CAV'09]

Hierarchic Superposition [BGW,AAECC'94][BW,CADE'13]
Instantiation-based TP [GK,LICS'03][GK,LPAR'06]
…



Efforts to combine both techniques:

E-matching [DNS,J.ACM’05][R,LPAR’12]
Array fragments [BMS,VMCAI’06][HIV,FoSSaCS’08]
Model based quantifier instantiation [GdM,CAV’09]

Hierarchic Superposition [BGW,AAECC‘94][BW,CADE‘13]
Instantiation-based TP [GK,LICS’03][GK,LPAR’06]
…

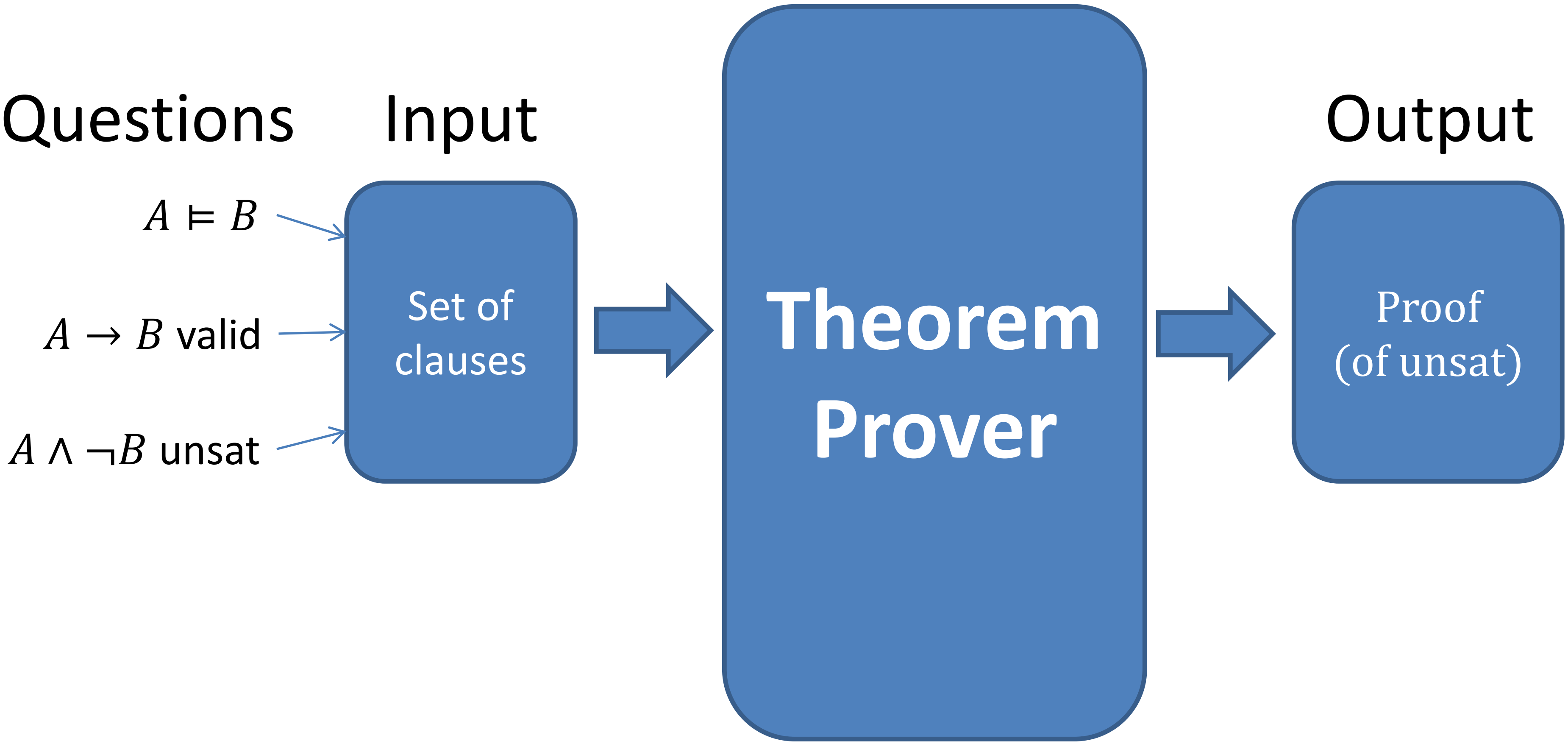
Contribution

1. Observation: state-of-the-art theorem provers can not handle problems with extensionality axioms

2. Solution: new inference rule extensionality resolution

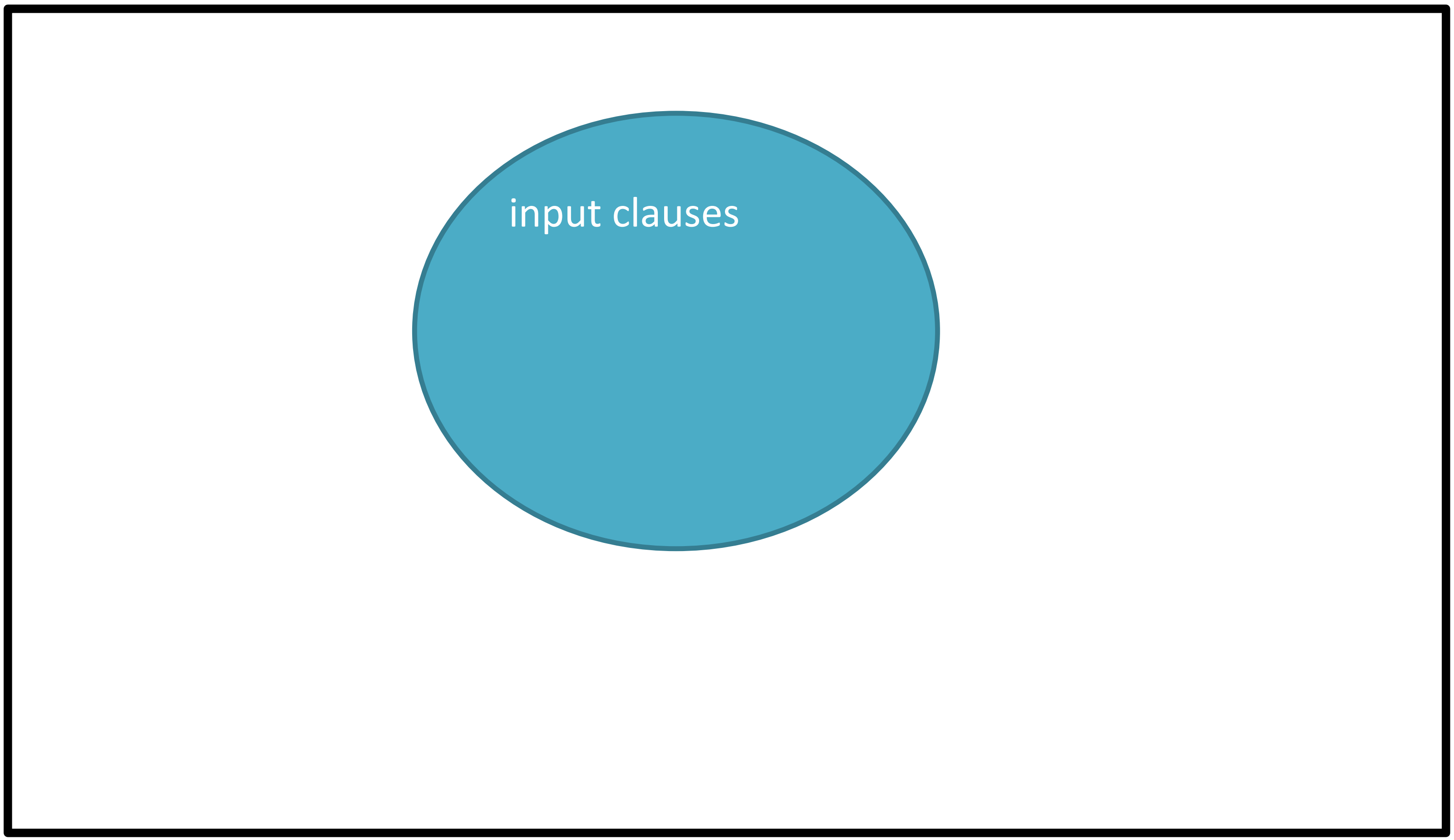
3. Implementation in the Vampire theorem prover

First-Order Theorem Proving



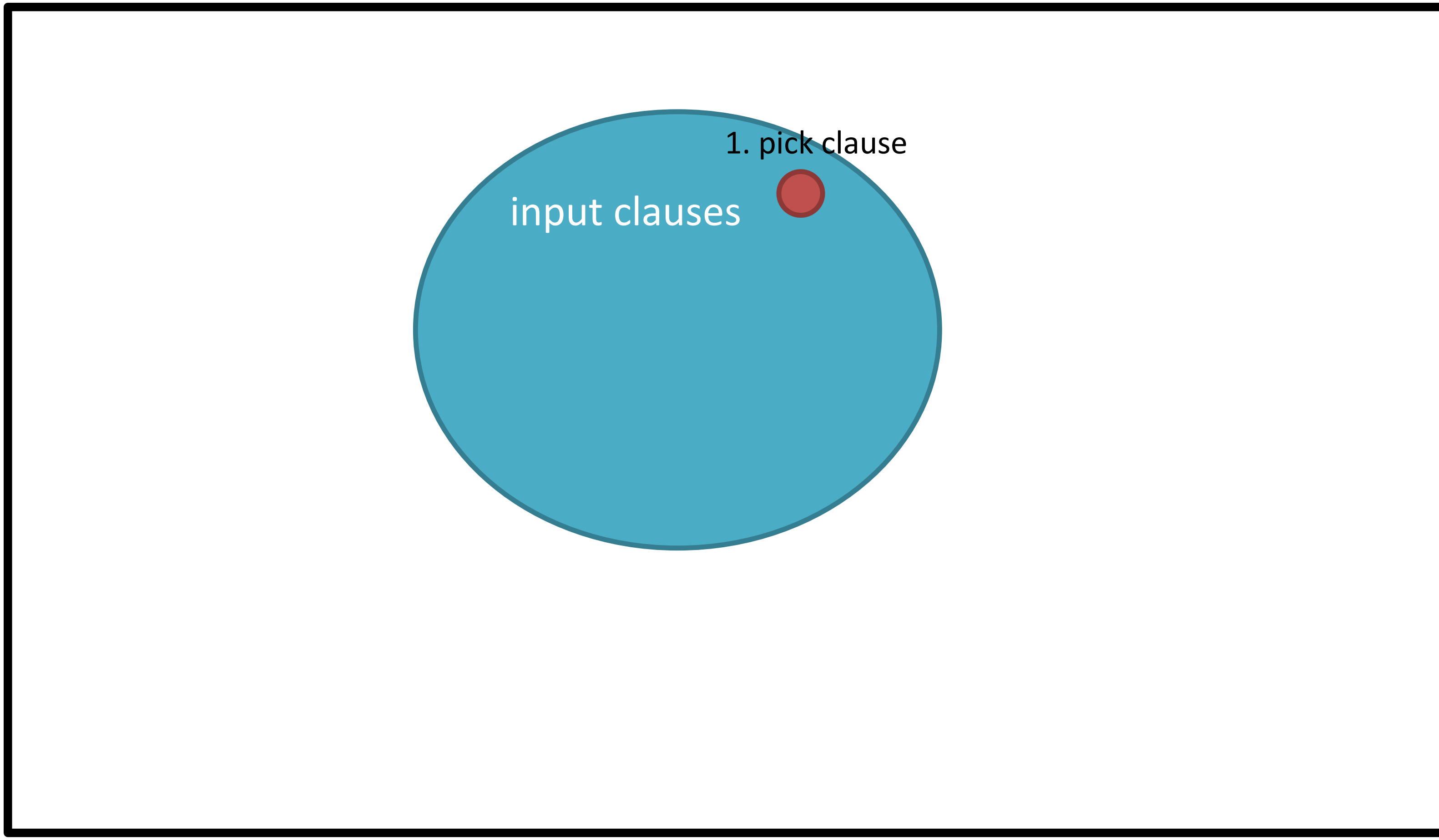
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



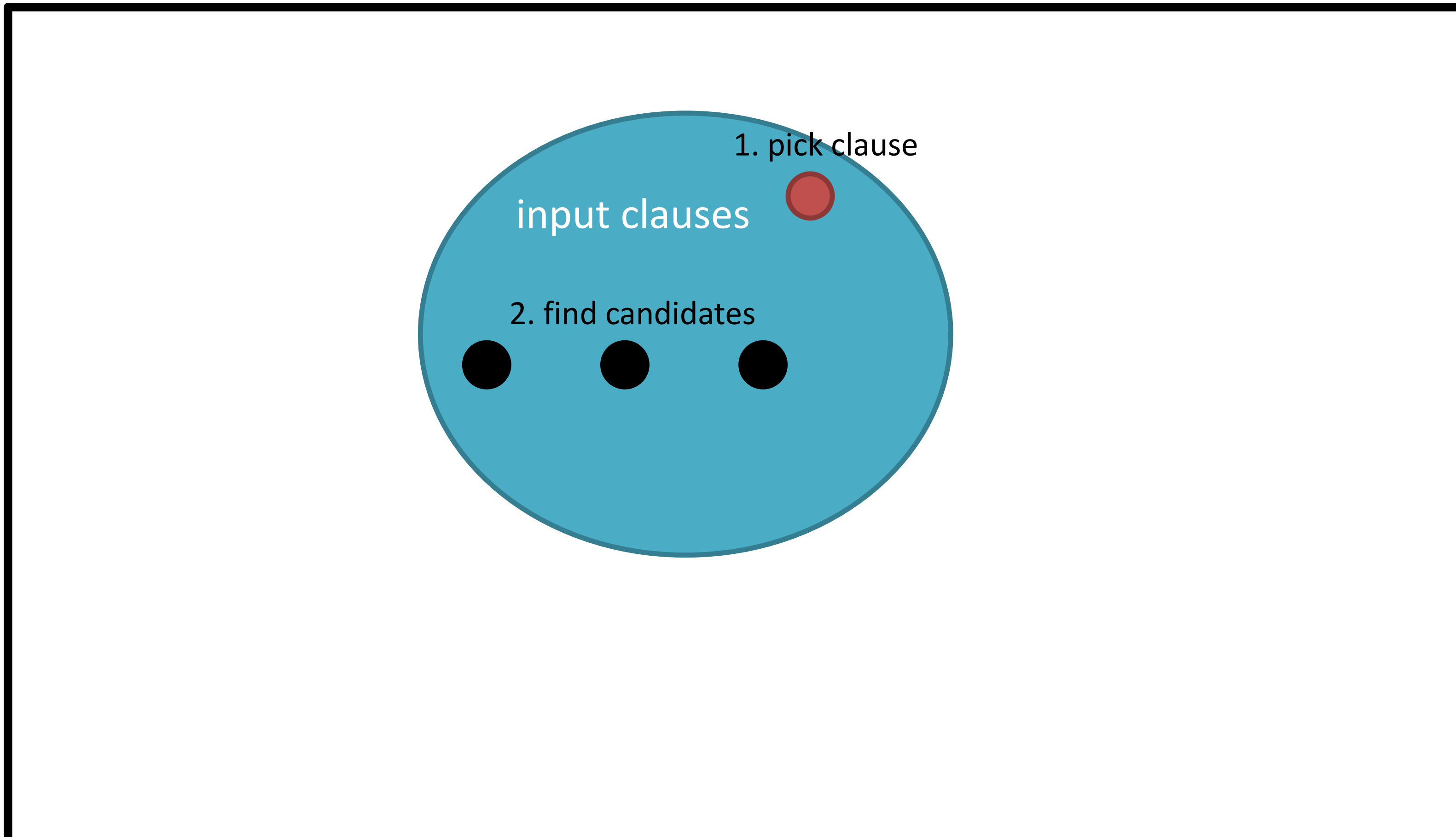
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



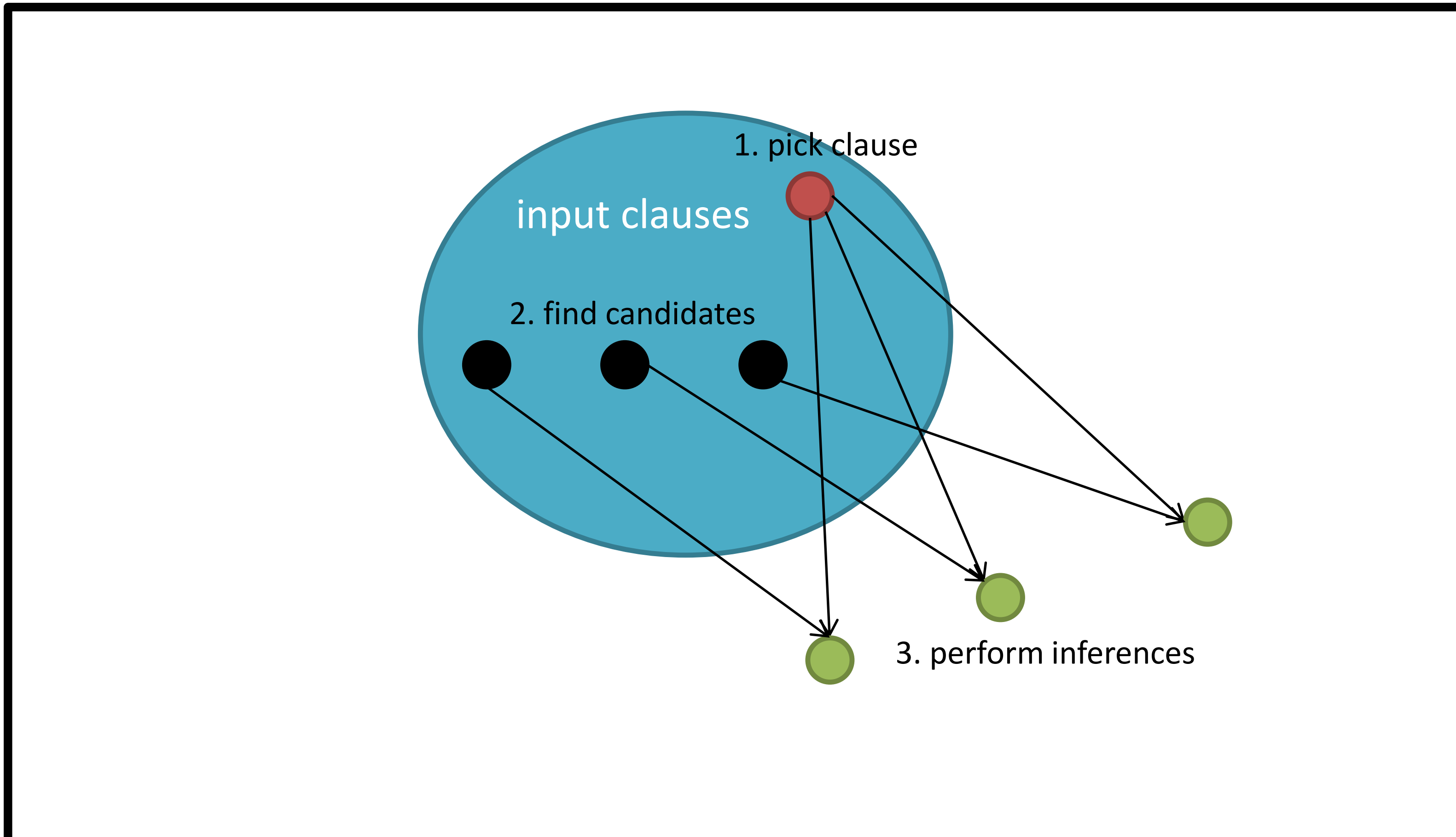
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



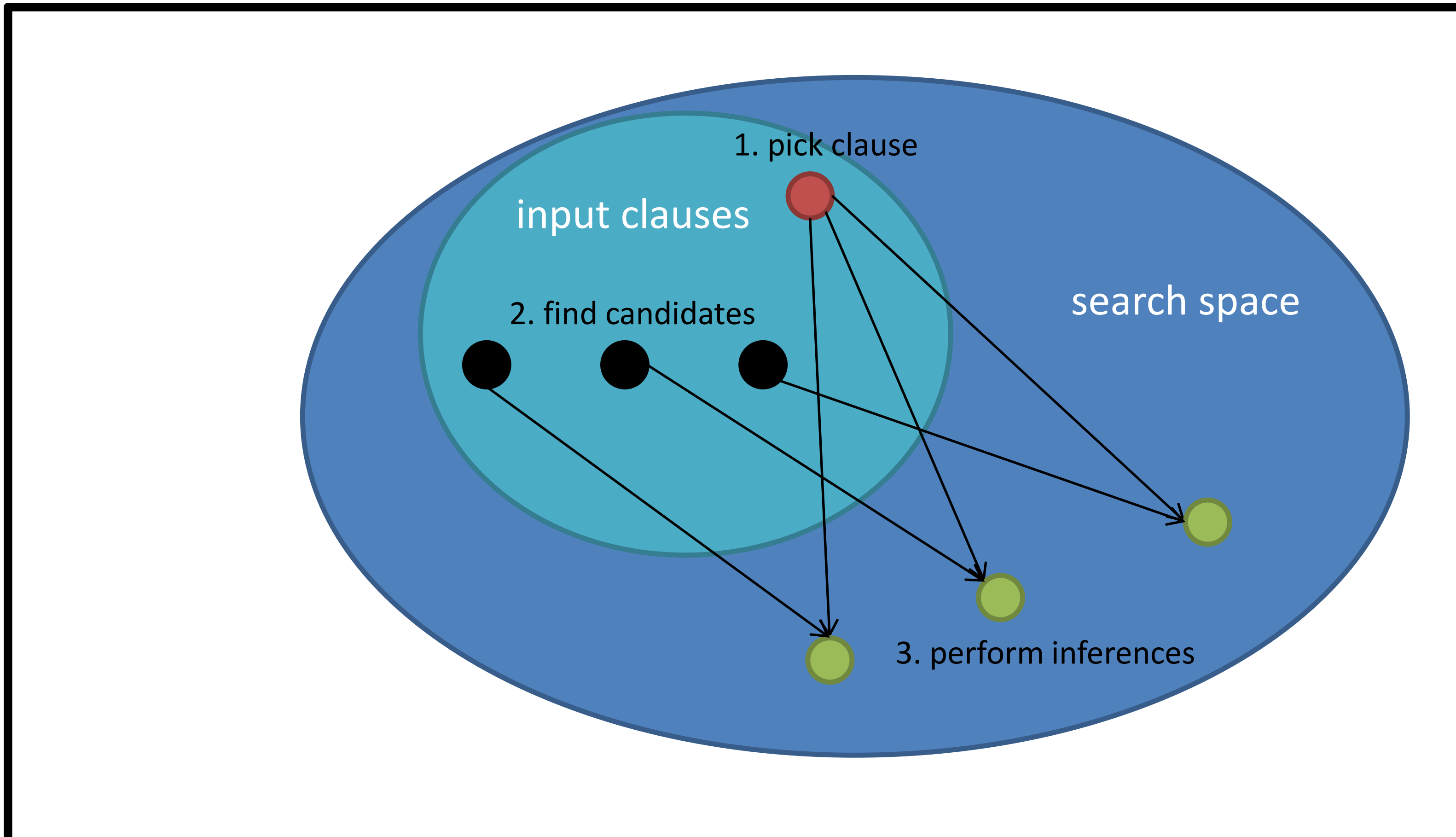
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



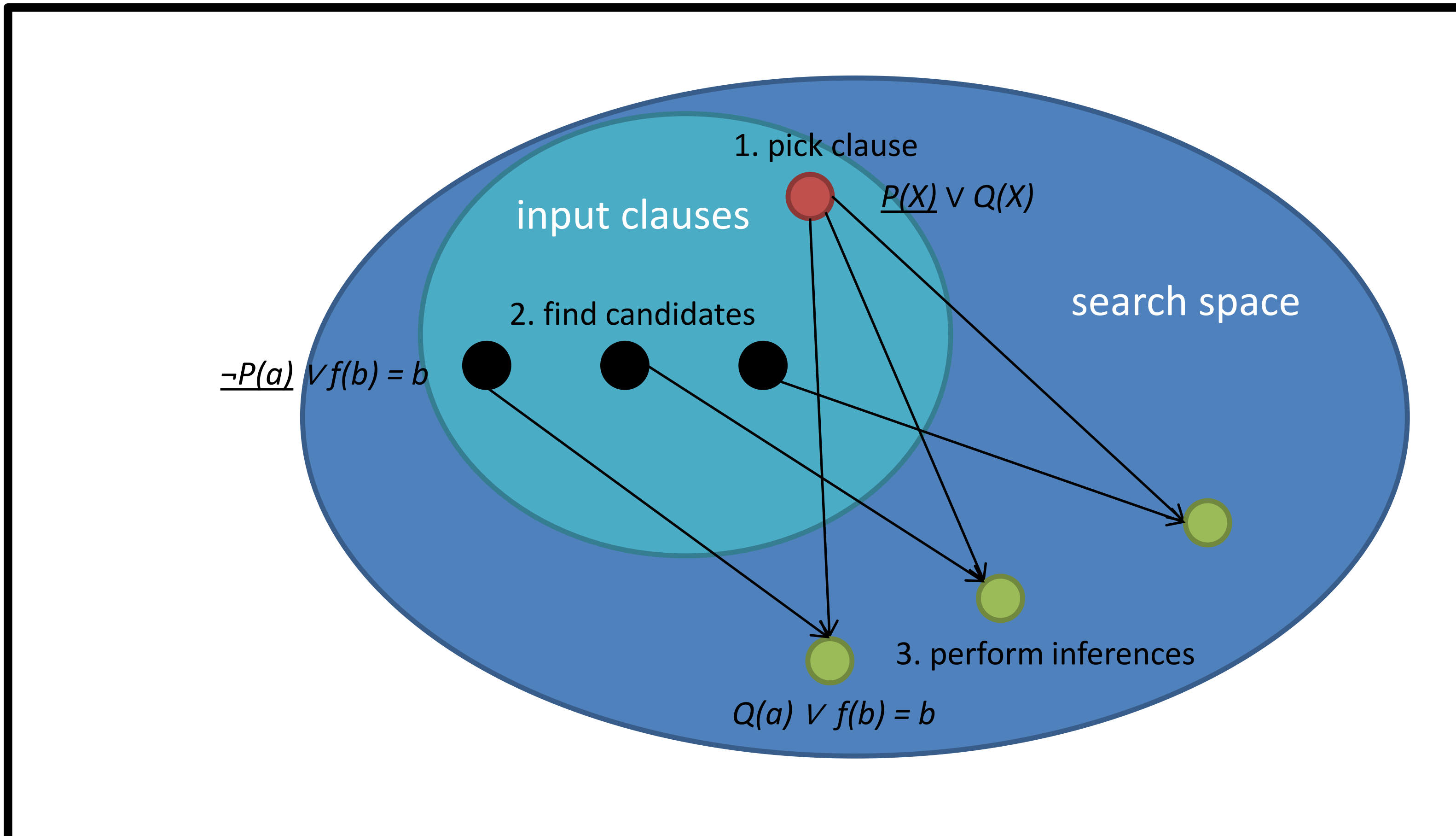
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



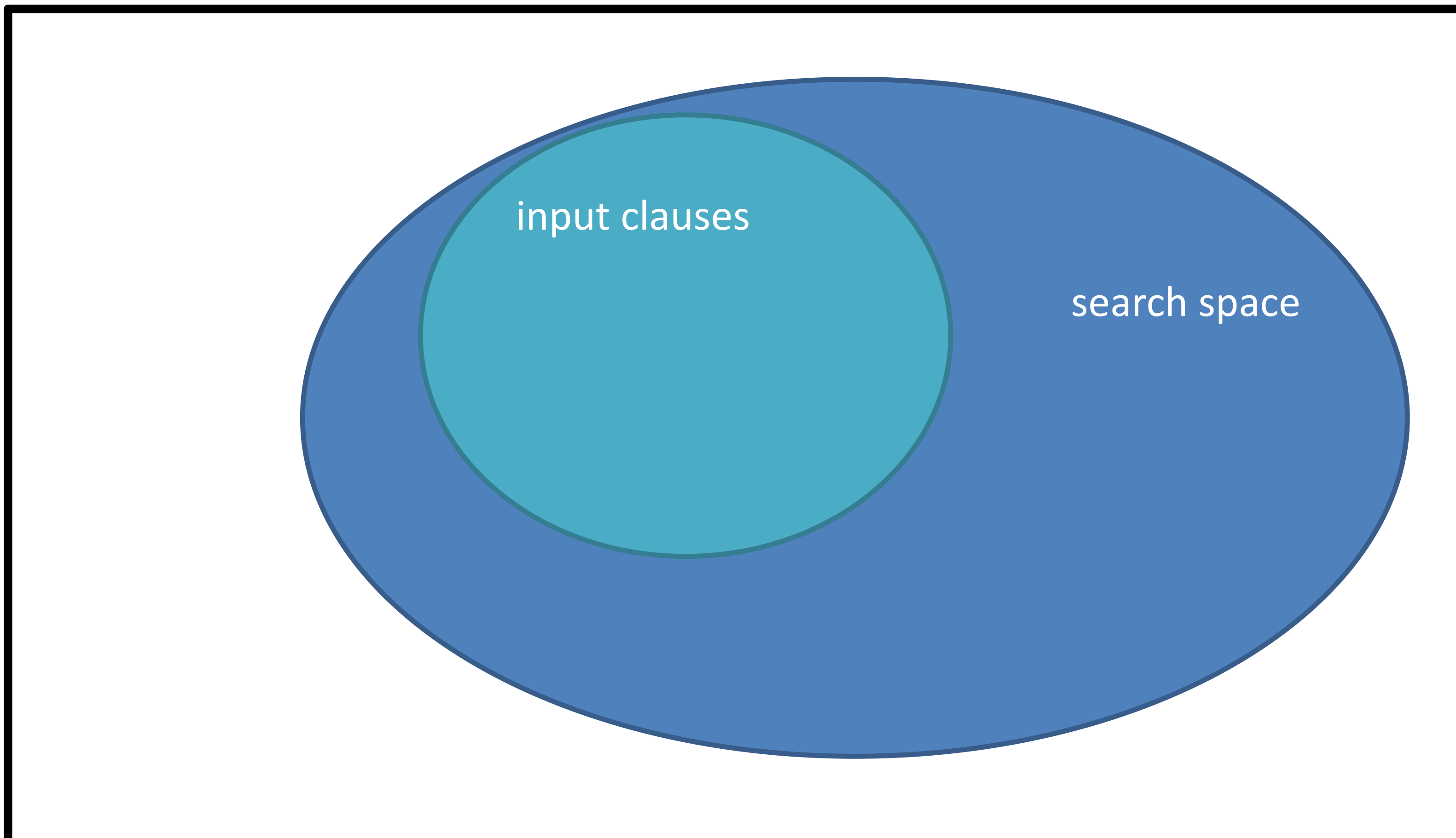
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



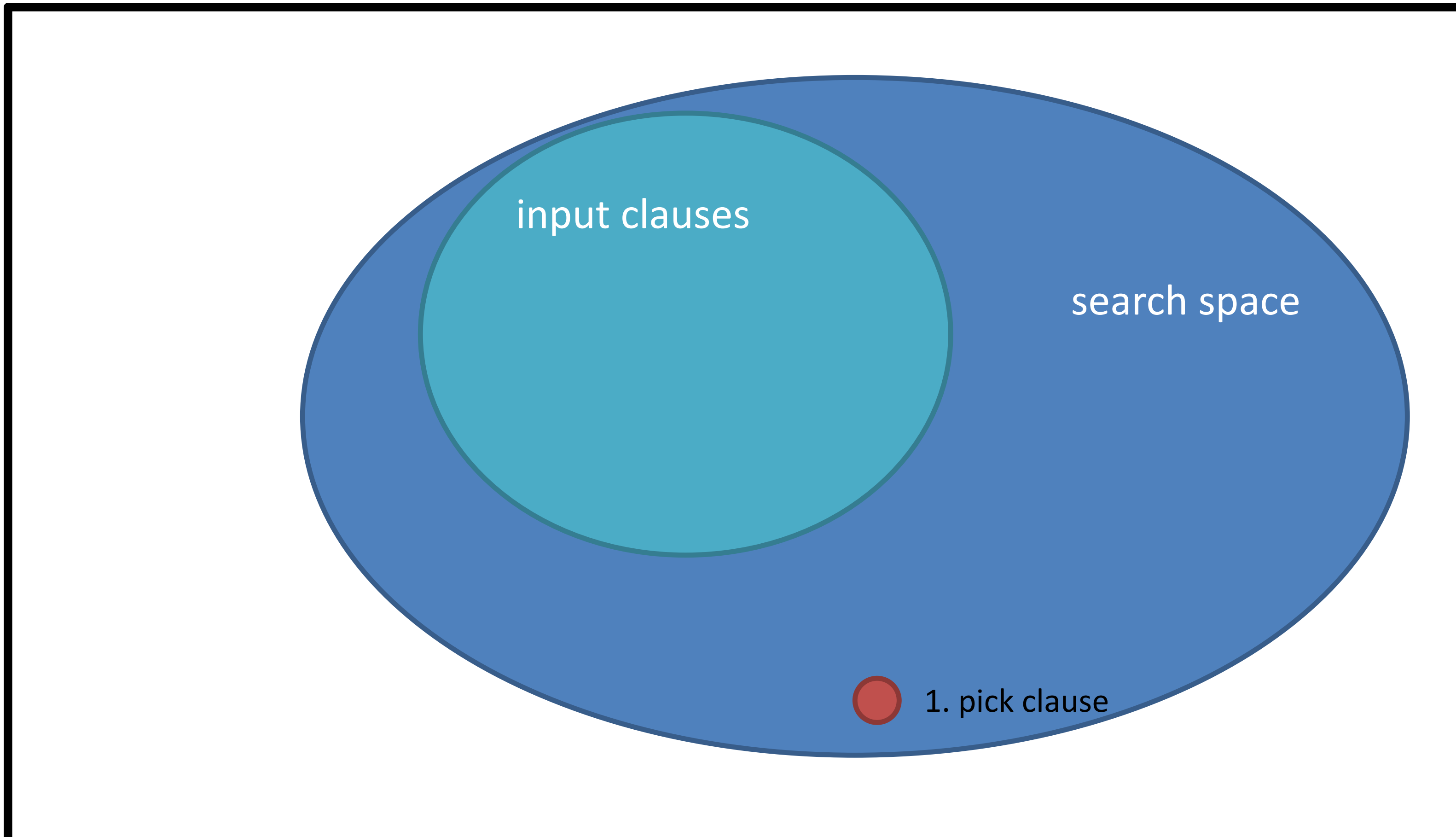
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



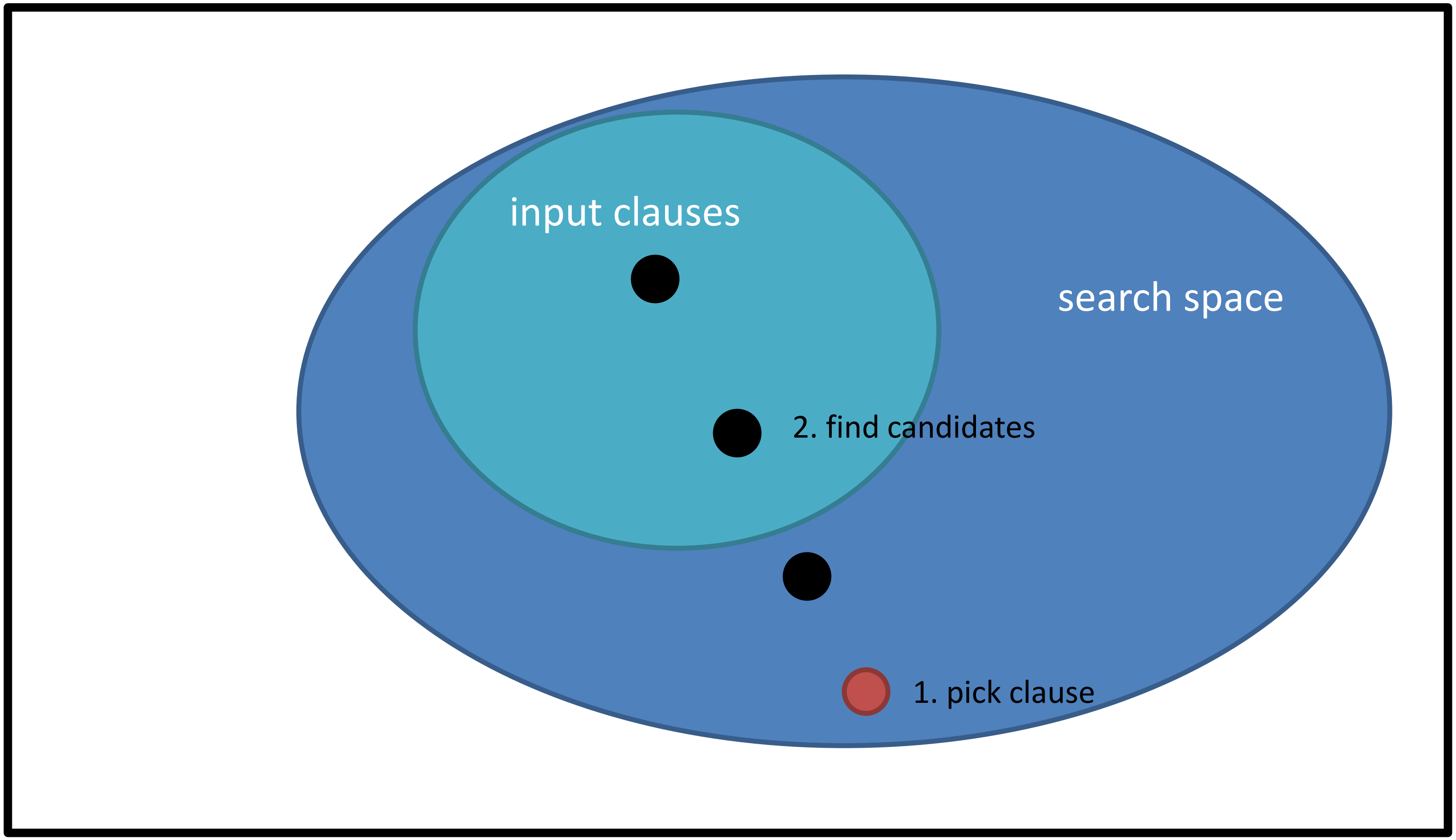
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



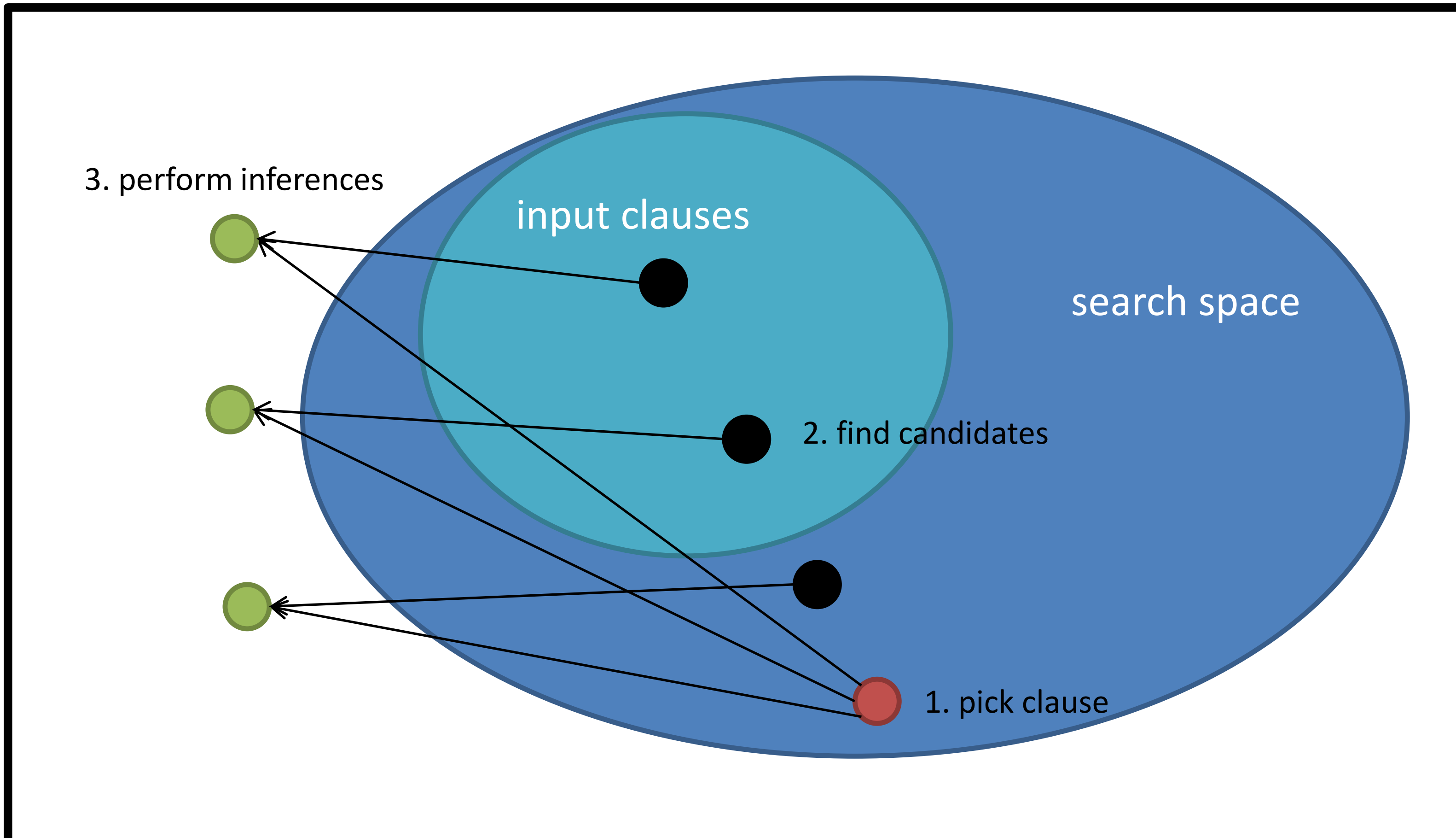
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



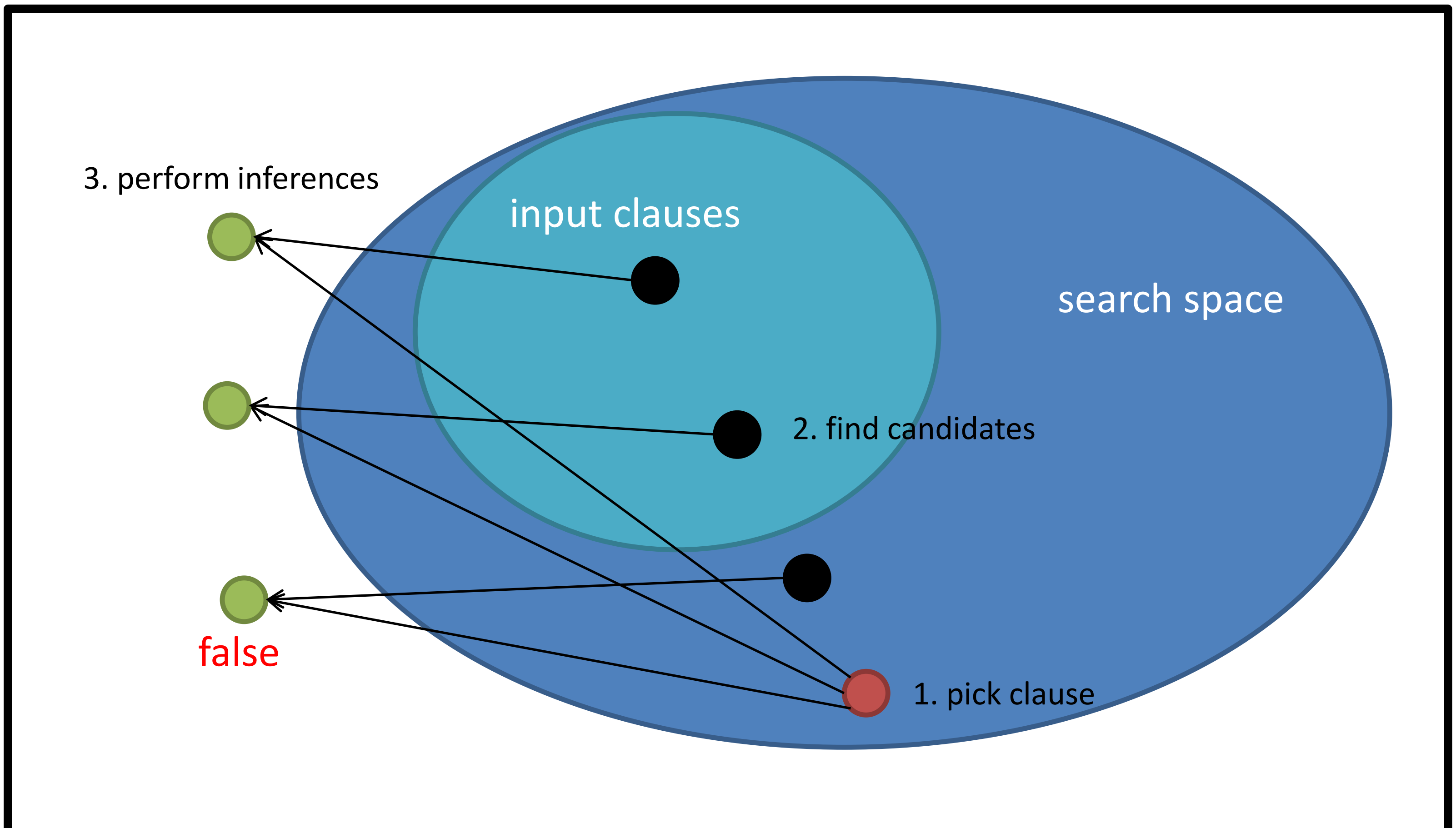
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



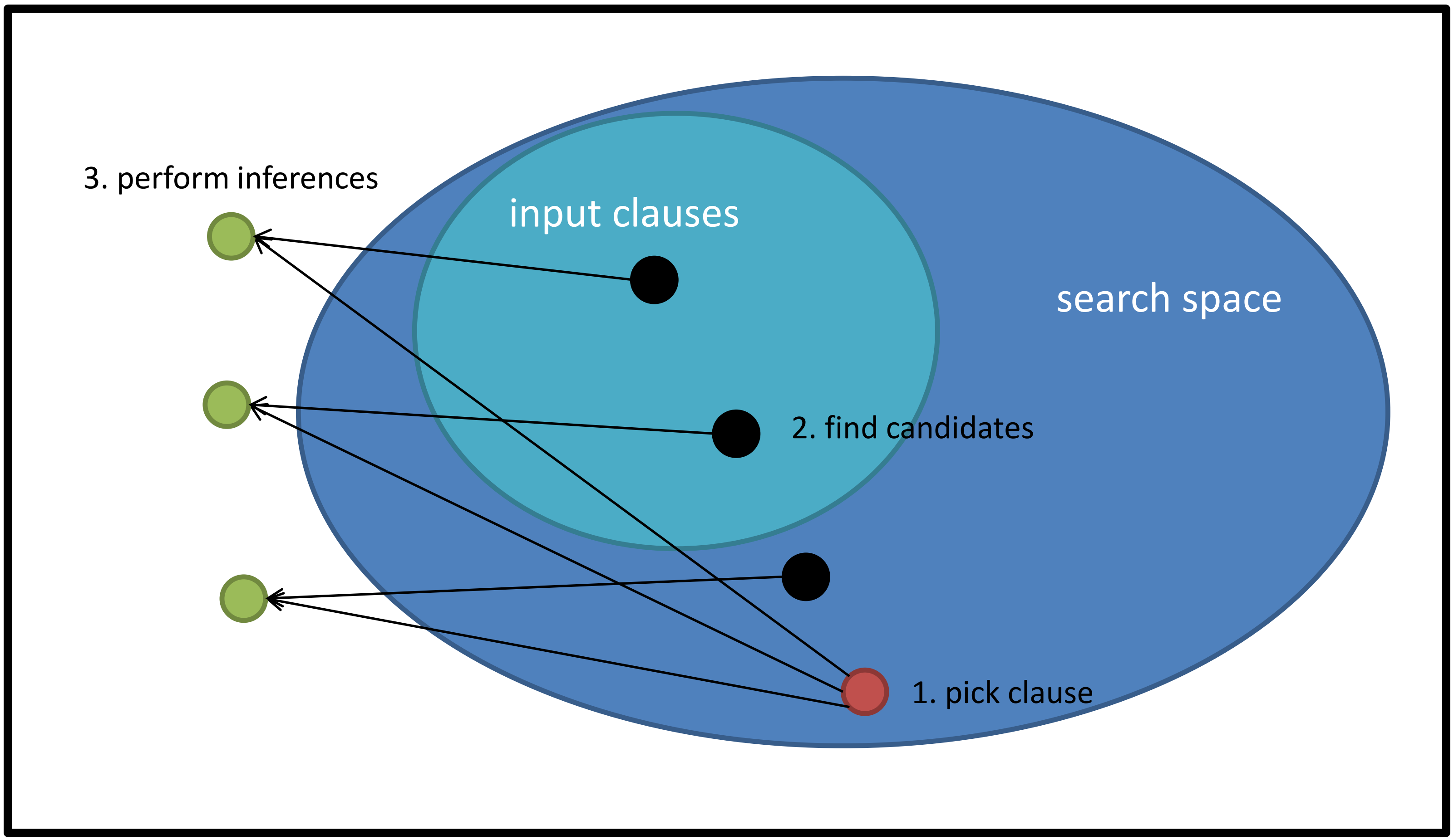
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



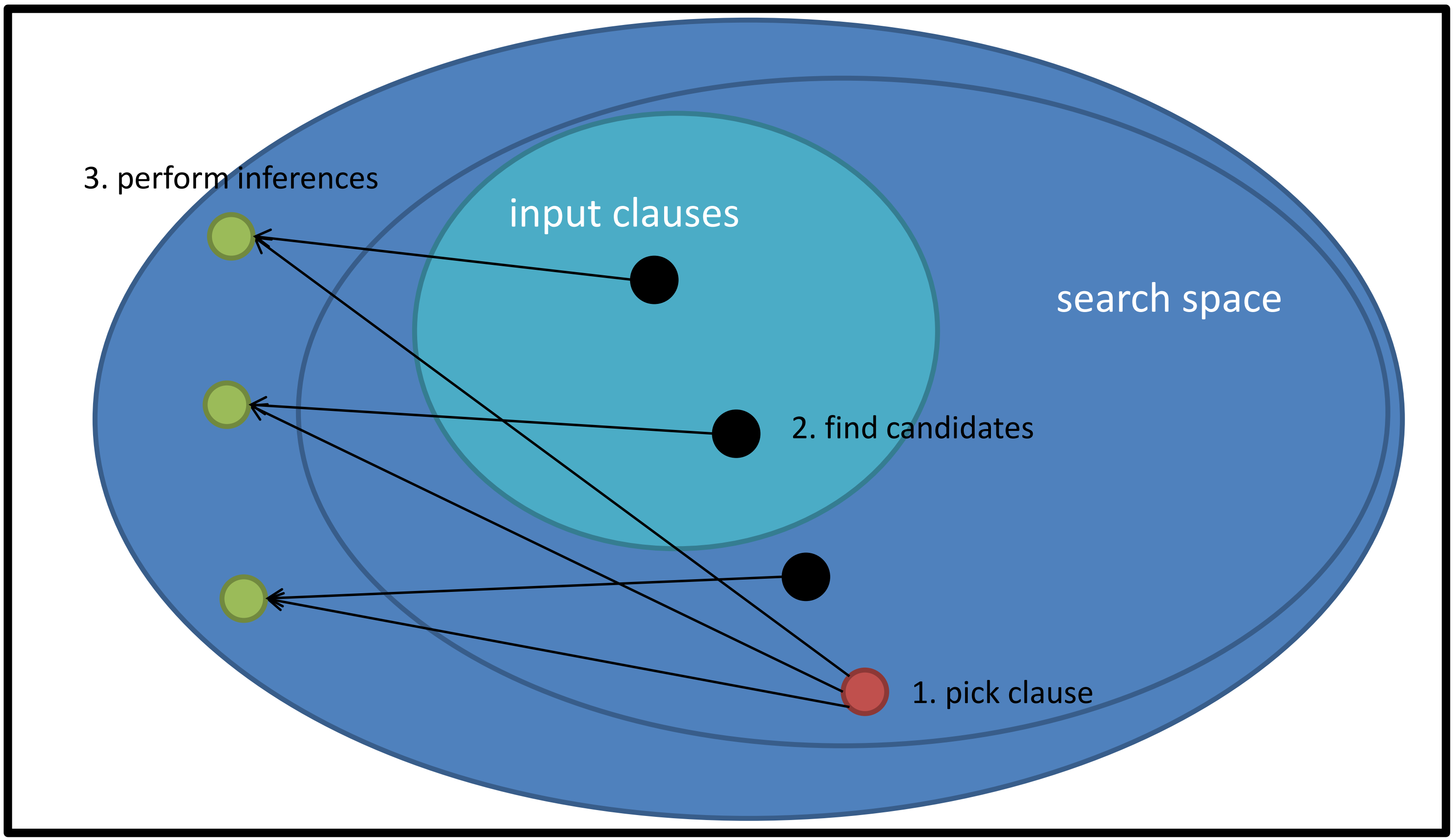
Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



Superposition Theorem Proving

Superposition calculus + Saturation Algorithm



Superposition Theorem Proving

Superposition calculus + Saturation Algorithm

Superposition Theorem Proving

Superposition calculus + Saturation Algorithm

ATP Research

How to organize proof search?

How to organize proof search?

Intuition

"Generally"

pick *"small"* clauses,

select only *"most complex"* literals in picked clause and candidate clauses,

and *"simplify"* them.

How to organize proof search?

Intuition	**Formal concepts**
"Generally"	Fair inference process
pick *"small"* clauses,	Simplification ordering (e.g. KBO)
select only *"most complex"* literals in picked clause and candidate clauses,	Literal selection
and *"simplify"* them.	Constraints on inference rules

How to organize proof search?

Intuition	**Formal concepts**
“Generally”	Fair inference process
pick *“small”* clauses,	Simplification ordering (e.g. KBO)
select only *“most complex”* literals in picked clause and candidate clauses,	Literal selection
and *“simplify”* them.	Const … renc… rules

Not always optimal,
e.g. for theories
with extensionality!

Extensionality

- An extensionality axiom defines the meaning of equality for certain objects

- Examples
  - Set Extensionality Axiom
    $$\forall X \forall Y \left(\forall e \left(e \in X \leftrightarrow e \in Y\right) \rightarrow X = Y\right)$$
  - Array Extensionality Axiom
    $$\forall X \forall Y \left(\forall i \left(X[i] = Y[i]\right) \rightarrow X = Y\right)$$

Reasoning with Extensionality

Prove: $\forall X \forall Y\ (X \cup Y = Y \cup X)$

Take two arbitrary sets a and b.

By extensionality, show for arbitrary element e:

$$e \in a \cup b \leftrightarrow e \in b \cup a$$

- Assume $e \in a \cup b$,
  then $e \in a$ or $e \in b$, (def. of ∪)
  and in both cases $e \in b \cup a$. (commut. of "or") (def. of ∪)
- Assume $e \in b \cup a$; symmetric.

Almost trivial, but …

Extensional Crisis

… hard for FO theorem provers.

Top provers from CASC-24 competition last year:

$$X \cup Y = Y \cup X$$

all tools timeout (1 minute)

$$X \cap Y \subseteq Z \subseteq X \cup Y \rightarrow (X \cup Y) \cap (\bar{X} \cup Z) = Y \cup Z$$

all tools timeout (1 hour)

Why do all top provers fail?

Why do all top provers fail?

Extensionality axioms as clauses

Array: $\forall X \forall Y\ (\forall i\ (X[i] = Y[i]) \rightarrow X = Y)$

$$x[g(x,y)] \neq y[g(x,y)] \vee x = y$$

Clause form

Why do all top provers fail?

Extensionality axioms as clauses

Array: $\forall X \forall Y \; (\forall i \; (X[i] = Y[i]) \rightarrow X = Y)$

$$x[g(x,y)] \neq y[g(x,y)] \vee x = y$$

Clause form

Set: $\forall X \forall Y \; (\forall e \; (e \in X \leftrightarrow e \in Y) \rightarrow X = Y)$

$$f(x,y) \notin x \vee f(x,y) \notin y \vee x = y$$

Why do all top provers fail?

- Extensionality axioms as clauses
  Array: $x[g(x,y)] \neq y[g(x,y)] \lor x = y$
  Set: $f(x,y) \notin x \lor f(x,y) \notin y \lor x = y$

Why do all top provers fail?

- Extensionality axioms as clauses

  Array: $x[g(x,y)] \neq y[g(x,y)] \lor x = y$

  Set: $f(x,y) \notin x \lor f(x,y) \notin y \lor x = y$

Why do all top provers fail?

- Extensionality axioms as clauses
  Array: $x[g(x,y)] \neq y[g(x,y)] \vee x = y$
  Set: $f(x,y) \notin x \vee f(x,y) \notin y \vee x = y$
- $x = y$ is always the smallest literal → will not be selected

Why do all top provers fail?

- Extensionality axioms as clauses
  Array: $x[g(x,y)] \neq y[g(x,y)] \vee x = y$
  Set: $f(x,y) \notin x \vee f(x,y) \notin y \vee x = y$
- $x = y$ is always the smallest literal → will not be selected
- Prover searches in the wrong direction

Why do all top provers fail?

Just select $x = y$!?!

- Extensionality axioms as clauses
  Array: $x[g(x,y)] \neq y[g(x,y)] \vee x = y$
  Set: $f(x,y) \notin x \vee f(x,y) \notin y \vee x = y$
- $x = y$ is always the smallest literal → will not be selected
- Prover searches in the wrong direction

OUR SOLUTION

Extensionality resolution inference rule

$$\underbrace{\boxed{x = y} \vee C}_{\text{Extensionality axiom}} \qquad \underbrace{\underline{s \neq t} \vee D}_{\text{Selected inequality}}$$

OUR SOLUTION

Extensionality resolution inference rule

$$\frac{\overbrace{\boxed{x = y} \vee C}^{\text{Extensionality axiom}} \qquad \overbrace{\underline{s \neq t} \vee D}^{\text{Selected inequality}}}{C\theta \vee D} \qquad \theta = \{x \mapsto s, y \mapsto t\}$$

OUR SOLUTION

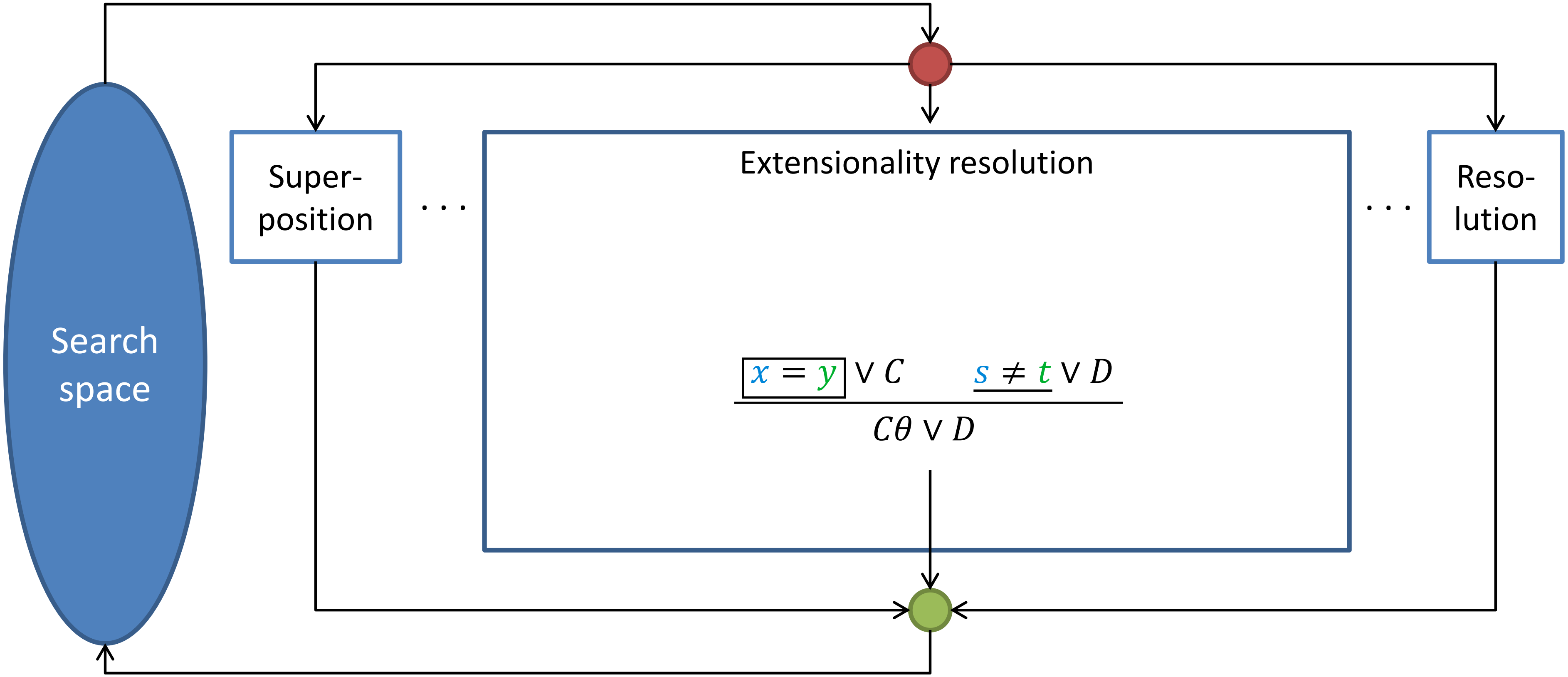
Extensionality resolution inference rule

$$\frac{\overbrace{\boxed{x = y} \vee C}^{\text{Extensionality axiom}} \qquad \overbrace{\underline{s \neq t} \vee D}^{\text{Selected inequality}}}{C\theta \vee D} \qquad \theta = \{x \mapsto s, y \mapsto t\}$$

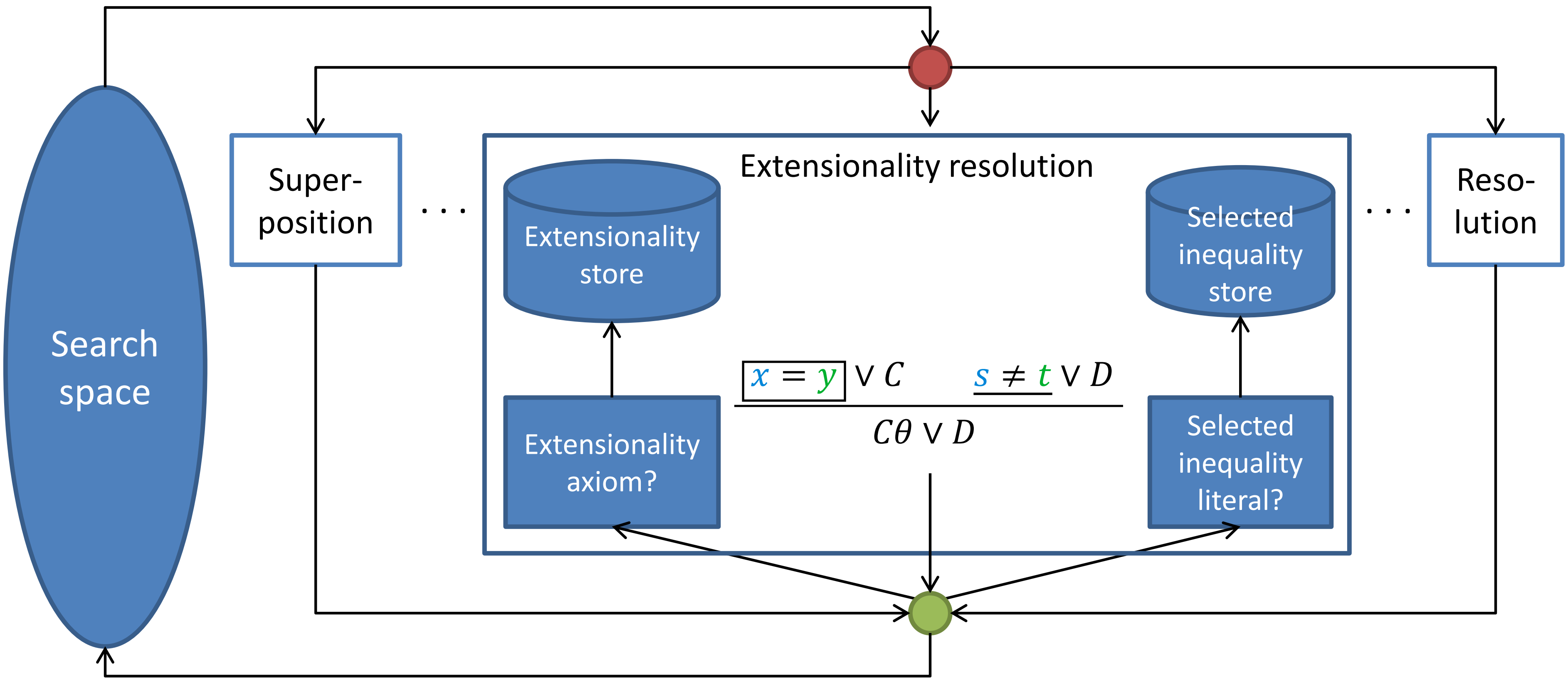
Example:

$$\frac{\boxed{x = y} \vee f(x, y) \notin x \vee f(x, y) \notin y \qquad \underline{a \cup b \neq b \cup a}}{f(a \cup b, b \cup a) \notin a \cup b \vee f(a \cup b, b \cup a) \notin b \cup a}$$

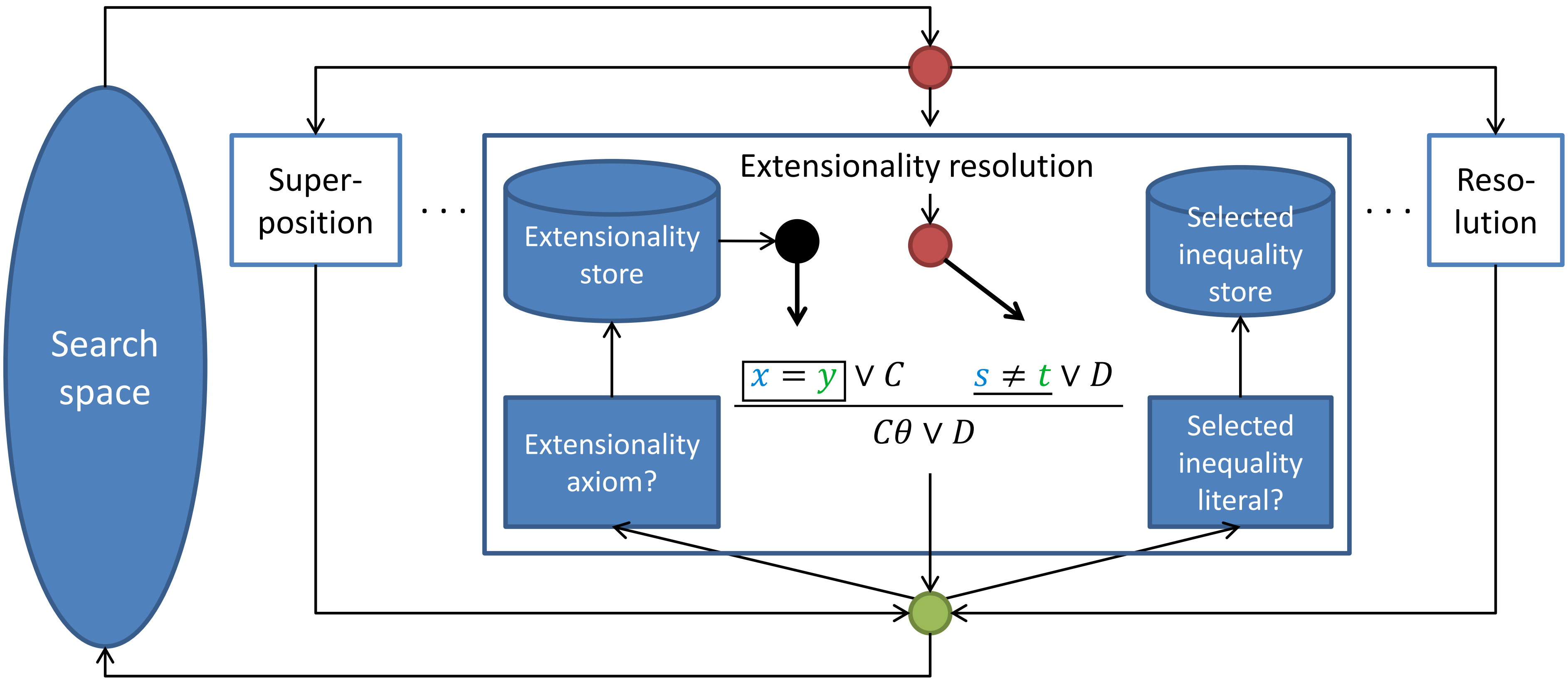
Integration into saturation algorithms



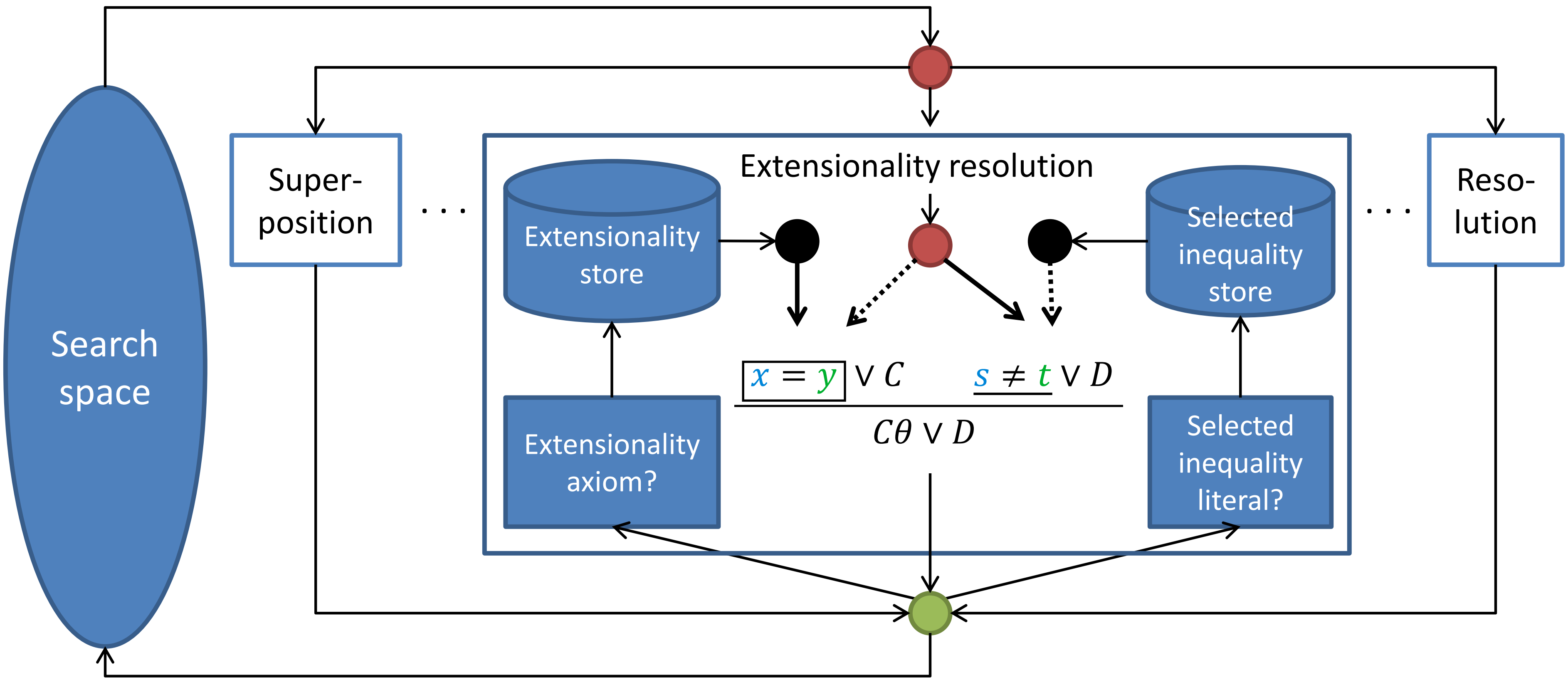
Integration into saturation algorithms



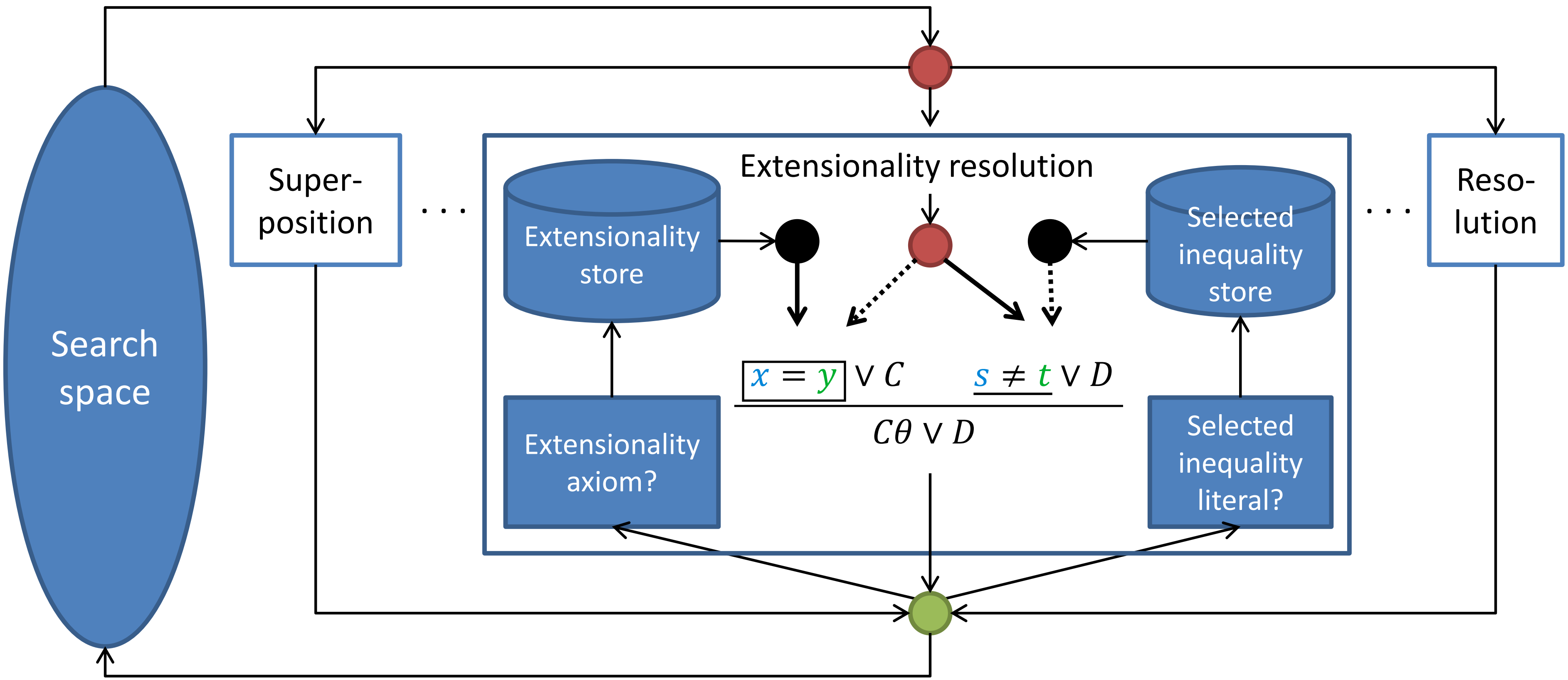
Integration into saturation algorithms



Integration into saturation algorithms



Integration into saturation algorithms



+ Straight forward to implement
+ No special index structures required
+ No changes to the underlying inference mechanism

Recognition of extensionality axioms

Recognition of extensionality axioms

- The Good,
  - Known extensionality axioms (set, array, subset, …)

Recognition of extensionality axioms

- The Good,
  - Known extensionality axioms (set, array, subset, …)
- the Bad,
  - Constructor axioms

$$f(x) \neq f(y) \vee x = y$$

Recognition of extensionality axioms

- The Good,
  - Known extensionality axioms (set, array, subset, …)
- the Bad,
  - Constructor axioms

$$f(x) \neq f(y) \vee x = y$$

- and the Ugly?

$$x_4 = x_6 \vee ssSkC0 \vee \neg in(x_6, x_7) \vee \neg front(x_7) \vee \neg furniture(x_7) \vee \neg seat(x_7) \vee \neg fellow(x_6) \vee \neg man(x_6) \vee \neg young(x_6) \vee \neg seat(x_5) \vee \neg furniture(x_5) \vee \neg front(x_5) \vee \neg in(x_4, x_5) \vee \neg young(x_4) \vee \neg man(x_4) \vee \neg fellow(x_4) \vee \neg in(x_2, x_3) \vee \neg city(x_3) \vee \neg hollywood(x_3) \vee \neg event(x_2) \vee \neg barrel(x_2, x_1) \vee \neg down(x_2, x_0) \vee \neg old(x_1) \vee \neg dirty(x_1) \vee \neg white(x_1) \vee \neg car(x_1) \vee \neg chevy(x_1) \vee \neg street(x_0) \vee \neg way(x_0) \vee \neg lonely(x_0).$$

Implementation and Evaluation

- Implementation VAMPIRE^{EX}
  - extension of the VAMPIRE theorem prover
  - ca. 1,000 lines of code

- Benchmark suits
  - Handcrafted set theory problems
  - SMT-LIB array problems
  - TPTP library

Set Theory Experiments

- 36 handcrafted problems
- VAMPIRE^{EX} solves all problems very fast
  - > 0.1 s: 5
  - > 1 s: 2
- 17 problems only solved by VAMPIRE^{EX}

#	VAMPIRE^{EX}	IPROVER	PRINCESS	VAMPIRE	CVC4	E	MUSCADET	ZIPPER-POSITION	BEAGLE	E-KR-HYPER
1	0.02	13.70	7.78				0.10			
2	0.01		7.92		41.54					
3	0.06									
4	0.02	1.47	9.36	0.21	30.24	1.38	0.65			
5	0.02	0.89	17.19	1.92	56.05	33.98	0.10			
6	0.02	0.29	15.41		54.40					
7	0.03									
8	0.02									
9	0.02									
10	0.04									
11	0.04									
12	0.02	0.58	15.36	0.39	50.52					
13	0.02	1.10	15.23	0.14	30.34	0.35	0.09			
14	0.02	2.44	7.80	0.02		0.07	0.09	10.59	6.85	
15	0.02	13.80	8.55	0.12	32.15	1.55				
16	3.41									
17	0.01			0.02	30.94	24.31		0.44		
18	0.94									
19	0.03									
20	0.02									
21	0.03									
22	1.73									
23	0.24									
24	0.15			0.43						
25	0.05									
26	0.05									
27	0.03	11.80	25.80			52.47				
28	0.06	11.80	33.73	0.80	34.32					
29	0.03	38.63		0.22	31.33	1.64				
30	0.02	3.32	12.36	0.06		27.54	0.11	23.30		
31	0.03									
32	0.04									
33	0.02	23.28	20.92							
34	0.02	0.50	6.71	0.02	30.29	0.03	0.08	0.59	2.22	
35	0.02	8.23	6.87	0.23	30.34	30.23				
36	0.02	1.50		20.86	44.77					
	36	16	15	14	13	11	7	4	2	0

Array Experiments

278 problems from the QF_AX category of SMT-LIB

Prover	solved	runtime
VAMPIRE^{EX}	193	2,255.06
VAMPIRE	110	2,272.17
E	81	600.01
BEAGLE	16	185.44
ZIPPERPOSITION	15	49.27
PRINCESS	10	35.02
iPROVER	9	47.13
CVC4	8	0.36
E-KRHYPER	8	1.26
MUSCADET	4	0.41

Number of solved problems increased from 39.57% to 69.42%.

TPTP Library Experiments

- 7033 problems with potential extensionality axioms
- VAMPIRE^{EX} solves 84 new problems

12 of them have CASC rating 1

Never solved before

Prover	solved	uniquely solved
VAMPIRE	4015	156
VAMPIRE^{EX}	3870	84

- Strategy scheduling

Value of a new technique lies in its complementary impact

Options in Vampire

age_weight_ratio
aig_bdd_sweeping
aig_conditional_rewriting
aig_definition_introduction
aig_definition_introduction_threshold
aig_formula_sharing
aig_inliner
arity_check
backward_demodulation
backward_subsumption
backward_subsumption_resolution
bfnt
binary_resolution
bp_add_collapsing_inequalities
bp_allowed_fm_balance
bp_almost_half_bounding_removal
bp_assignment_selector
bp_bound_improvement_limit
bp_conflict_selector
bp_conservative_assignment_selection
bp_fm_elimination
bp_max_prop_length
bp_propagate_after_conflict
bp_start_with_precise
bp_start_with_rational
bp_variable_selector
color_unblocking
condensation
decode
demodulation_redundancy_check
distinct_processor
epr_preserving_naming
epr_preserving_skolemization
epr_restoring_inlining
equality_propagation
equality_proxy
equality_resolution_with_deletion
extensionality_allow_pos_eq
extensionality_max_length
extensionality_resolution
flatten_top_level_conjunctions
forbidden_options
forced_options
forward_demodulation
forward_literal_rewriting

forward_subsumption
forward_subsumption_resolution
function_definition_elimination
function_number
general_splitting
global_subsumption
horn_revealing
hyper_superposition
ignore_missing
include
increased_numeral_weight
inequality_splitting
input_file
input_syntax
inst_gen_big_restart_ratio
inst_gen_inprocessing
inst_gen_passive_reactivation
inst_gen_resolution_ratio
inst_gen_restart_period
inst_gen_restart_period_quotient
inst_gen_selection
inst_gen_with_resolution
interpreted_simplification
latex_output
lingva_additional_invariants
literal_comparison_mode
log_file
lrs_first_time_check
lrs_weight_limit_only
max_active
max_answers
max_inference_depth
max_passive
max_weight
memory_limit
mode
name_prefix
naming
niceness_option
nongoal_weight_coefficient
nonliterals_in_clause_weight
normalize
output_axiom_names
predicate_definition_inlining
predicate_definition_merging

predicate_equivalence_discovery
predicate_equivalence_discovery_add_implications
predicate_equivalence_discovery_random_simulation
predicate_equivalence_discovery_sat_conflict_limit
predicate_index_introduction
print_clausifier_premises
problem_name
proof
proof_checking
protected_prefix
question_answering
random_seed
row_variable_max_length
sat_clause_activity_decay
sat_clause_disposer
sat_learnt_minimization
sat_learnt_subsumption_resolution
sat_lingeling_incremental
sat_lingeling_similar_models
sat_restart_fixed_count
sat_restart_geometric_increase
sat_restart_geometric_init
sat_restart_luby_factor
sat_restart_minisat_increase
sat_restart_minisat_init
sat_restart_strategy
sat_solver
sat_var_activity_decay
sat_var_selector
saturation_algorithm
selection
show_active
show_blocked
show_definitions
show_interpolant
show_new
show_new_propositional
show_nonconstant_skolem_function_trace
show_options
show_passive
show_preprocessing
show_skolemisations

show_symbol_elimination
show_theory_axioms
simulated_time_limit
sine_depth
sine_generality_threshold
sine_selection
sine_tolerance
smtlib_consider_ints_real
smtlib_flet_as_definition
smtlib_introduce_aig_names
sos
split_at_activation
splitting
ssplitting_add_complementary
ssplitting_component_sweeping
ssplitting_congruence_closure
ssplitting_eager_removal
ssplitting_flush_period
ssplitting_flush_quotient
ssplitting_nonsplittable_components
statistics
superposition_from_variables
symbol_precedence
tabulation_bw_rule_subsumption_resolution_by_lemmas
tabulation_fw_rule_subsumption_resolution_by_lemmas
tabulation_goal_awr
tabulation_goal_lemma_ratio
tabulation_instantiate_producing_rules
tabulation_lemma_awr
test_id
thanks
theory_axioms
time_limit
time_statistics
trivial_predicate_removal
unit_resulting_resolution
unused_predicate_definition_removal
use_dismatching
weight_increment
while_number
xml_output

Conclusion

- Extensional crisis in the life of theorem provers
- Extensionality resolution: the right medication to overcome the crisis
- Future
  - Strategy synthesis
  - Combination of theories (esp. arithmetic)