UNIVERSITÀ DELLA CALABRIA

Advances in WASP

Mario Alviano Carmine Dodaro Nicola Leone
Francesco Ricca

University of Calabria, Italy

Lexington, Kentucky
LPNMR 2015

Outline

Outline

Context and motivation

Answer Set Programming (ASP)

- declarative programming paradigm
- strong theoretical basis
- availability of efficient implementations
- ease in representing complex problems

Applications in several fields

- Artificial intelligence
- Linux package configuration
- Bioinformatics
- Industrial applications
- and many more!

Context and motivation

Answer Set Programming (ASP)

- declarative programming paradigm
- strong theoretical basis
- availability of efficient implementations
- ease in representing complex problems

Applications in several fields

- Artificial intelligence
- Linux package configuration
- Bioinformatics
- Industrial applications
- and many more!

Developing effective systems is a crucial research topic

Solving ground ASP programs

Computational tasks and applications

1. Model generation
   - Given a ground ASP program Π, find an answer set of Π

     → [Balduccini et al., LPNMR 2001; Gebser et al, TPLP 2011]

2. Optimum answer set search
   - Given a ground ASP program Π, find an answer set of Π with the minimum cost

     → [Marra et al., JELIA 2014; Koponen et al., TPLP 2015]

3. Cautious reasoning
   - Given a ground ASP program Π and a ground atom a, check whether a is true in all answer sets of Π

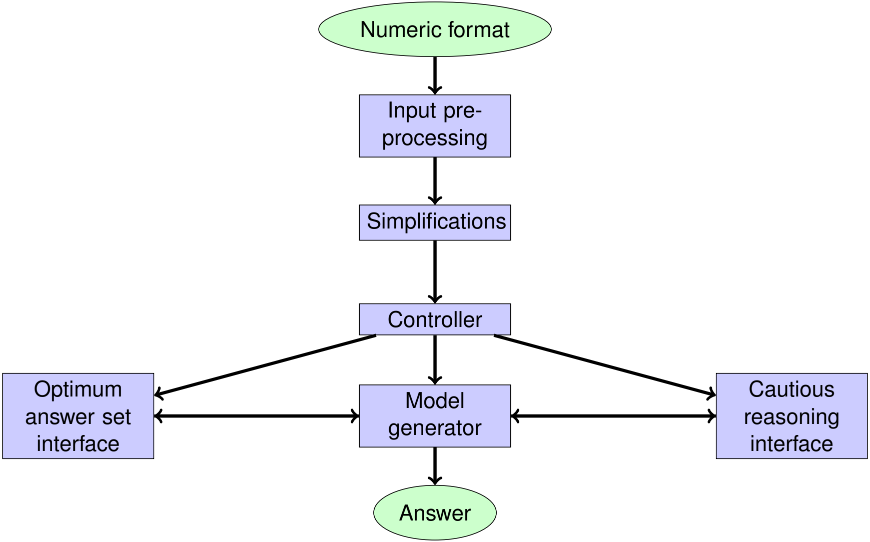
     → [Arenas et al., TPLP 2003; Eiter, LPNMR 2005]

Contribution

1. **Algorithms for the main computational tasks of ASP solving**
   - Model generation
     - Preprocessing, CDCL-like algorithm, incremental interface
   - Optimum answer set search
     - Model and core-guided algorithms
   - Cautious reasoning
     - New framework of anytime algorithms
2. **Implementation of WASP 2.1**
3. **Experimental analysis**

Outline

Architecture of WASP 2.1



Input preprocessing and simplifications

Preprocessing of the input program

- Deletion of duplicate rules
  - -> Even more than 80% in some benchmarks
- Deterministic inferences
  - -> Deletion of satisfied rules
- Clark's completion
  - -> Constraints for discarding unsupported models

Simplifications

- In the style of SATELITE [Eén and Biere, 2005]
  - -> Subsumption, self-subsumption, literals elimination

Model generator

- Based on a CDCL-like algorithm
  - Backtracking search algorithm
  - Introduced for SAT solving
  - Modified for taking in account ASP properties

    $\rightarrow$ more propagation rules

Model generator: propagation

Compute the deterministic consequences of an interpretation

Model generator: propagation

Compute the deterministic consequences of an interpretation

Derivation Rules

1. Unit propagation (from SAT)
2. Aggregates propagation (from Pseudo-Boolean)
3. Unfounded set propagation (ASP specific)
   - Detect unfounded sets
   - Algorithm based on source pointers [Simons et al., 2002]

Model generator: heuristics and learning

Learning

- Detect the reason of a conflict
- Learn constraints using 1-UIP schema

Deletion Policy

- Exponentially many constraints $\rightarrow$ forget something
- Less “useful” constraints are removed

Search Restarts

- Avoid unfruitful branches by restarting the search
- Based on some heuristic sequence

Branching Heuristics

- Look back MINISAT heuristic

Optimum answer set search

- Find the answer set with the minimum cost
  - **Input:** a propositional program Π with weak constraints
  - **Output:** an optimum answer set of Π

- Three family of algorithms implemented in WASP 2.1
  - Model-guided
    - OPT, BASIC, MGD
  - Core-guided
    - OLL, PMRES
    - Stratification: force the ASP solver to concentrate on weak constraints with higher weights
  - Mixed approaches
    - INTERLEAVING

Cautious reasoning

- Formally, an atom a is a cautious consequence of a program Π if a belongs to all answer sets of Π

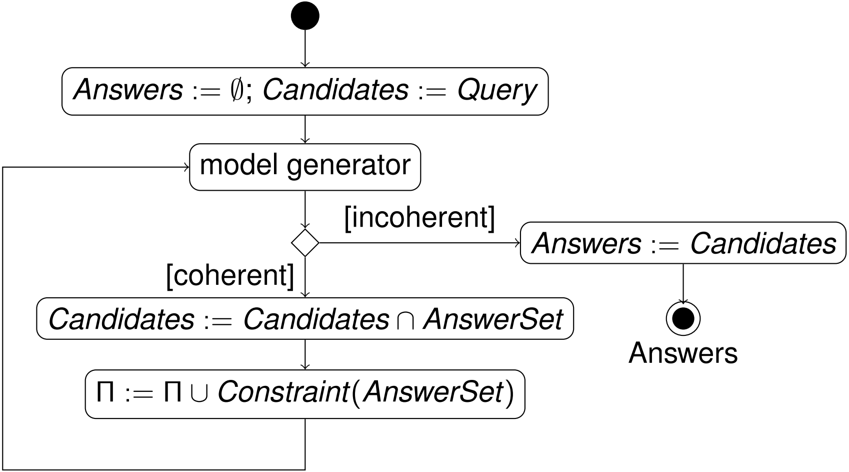
Compute cautious consequences

- **Input:** a propositional program Π and a query Q
- **Output:** atoms in Q which are cautious consequences of Π

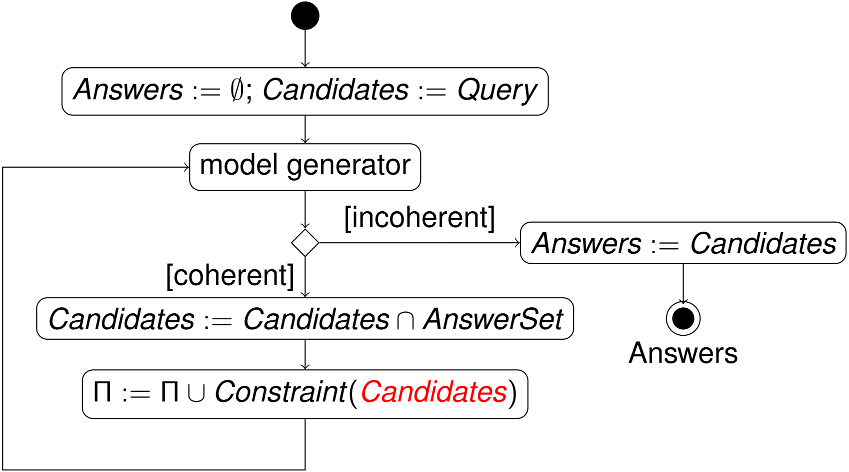
Cautious reasoning: algorithms

- Enumeration of models (DLV)
- Overestimate reduction (CLASP)
- Iterative coherence testing

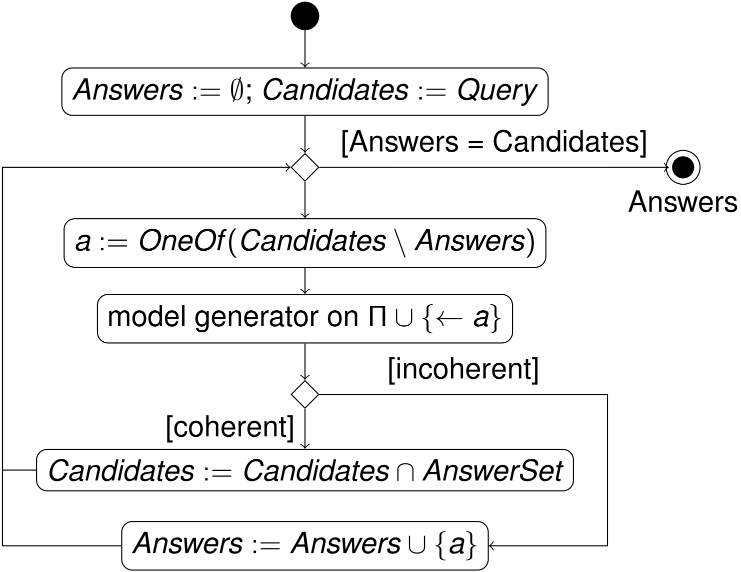
Cautious reasoning by enumeration of models



Cautious reasoning by overestimate reduction



Cautious reasoning by iterative coherence testing



Example

Program Π

$a \leftarrow not\ b$ $b \leftarrow not\ a$ % either a or b
$c \leftarrow a$ $c \leftarrow b$
$d \leftarrow not\ e$ $e \leftarrow not\ d$ % either d or e

Query Q

a, b, c

Execution

Step	**OneOf**	**answer set**	**Answers**	**Candidates**
0			$\emptyset$	$\{a, b, c\}$

Example

Program Π

$a \leftarrow not\ b \quad b \leftarrow not\ a \quad$ % either a or b
$c \leftarrow a \quad c \leftarrow b$
$d \leftarrow not\ e \quad e \leftarrow not\ d \quad$ % either d or e

Query Q

a, b, c

Execution

Step	OneOf	answer set	Answers	Candidates
0			$\emptyset$	$\{a, b, c\}$
1	c	*Incoherent*	$\{c\}$	$\{a, b, c\}$

Example

Program Π

$a \leftarrow not\ b \quad b \leftarrow not\ a \quad$ % either a or b

$c \leftarrow a \quad c \leftarrow b$

$d \leftarrow not\ e \quad e \leftarrow not\ d \quad$ % either d or e

Query Q

a, b, c

Execution

Step	OneOf	answer set	Answers	Candidates
0			$\emptyset$	$\{a, b, c\}$
1	c	*Incoherent*	$\{c\}$	$\{a, b, c\}$
2	a	$\{b, c, d\}$	$\{c\}$	$\{b, c\}$

Example

Program Π

$a \leftarrow not\ b \quad b \leftarrow not\ a \quad$ % either a or b

$c \leftarrow a \quad c \leftarrow b$

$d \leftarrow not\ e \quad e \leftarrow not\ d \quad$ % either d or e

Query Q

a, b, c

Execution

Step	**OneOf**	**answer set**	**Answers**	**Candidates**
0			$\emptyset$	$\{a, b, c\}$
1	c	*Incoherent*	$\{c\}$	$\{a, b, c\}$
2	a	$\{b, c, d\}$	$\{c\}$	$\{b, c\}$
3	b	$\{a, c, d\}$	$\{c\}$	$\{c\}$

Example

Program Π

$a \leftarrow not\ b$	$b \leftarrow not\ a$	% either a or b
$c \leftarrow a$	$c \leftarrow b$	
$d \leftarrow not\ e$	$e \leftarrow not\ d$	% either d or e

Query Q

a, b, c

Execution

Step	OneOf	answer set	Answers	Candidates
0			$\emptyset$	$\{a, b, c\}$
1	c	*Incoherent*	$\{c\}$	$\{a, b, c\}$
2	a	$\{b, c, d\}$	$\{c\}$	$\{b, c\}$
3	b	$\{a, c, d\}$	$\{c\}$	$\{c\}$

Anytime variants

- Often termination cannot be achieved in reasonable time
- Anytime algorithms are crucial for such cases to produce some sound answers
- Iterative coherence testing has the property to be anytime

Anytime variants

- Often termination cannot be achieved in reasonable time
- Anytime algorithms are crucial for such cases to produce some sound answers
- Iterative coherence testing has the property to be anytime

Good news!

Any algorithm for cautious reasoning can be anytime

- Hint: just check for new sound answers after each restart

Outline

Experiments environment

Benchmarks

- Comparison with CLASP and WASP 1
- Time and memory limit: 600 seconds and 15 GB

Model generation

- Benchmarks used in the 4th ASP Competition

Optimum answer set search

- Benchmarks used in the 4th ASP Competition and in [Andres et al., 2012]

Cautious reasoning

- Consistent query answering (CQA)
- Queries of [Kolaitis et al., 2013] on 10 different databases
- For each query two different encodings [Barceló and Bertossi, 2003;Manna et al., 2013]

Model generation

Problem	#	CLASP	WASP 1	WASP 2.1
BottleFillingProblem	30	30	30	30
GracefulGraphs	30	15	9	10
GraphColouring	30	13	8	8
HanoiTower	30	28	15	30
IncrementalScheduling	30	3	0	4
Labyrinth	30	26	21	25
NoMystery	30	9	5	7
PermutationPatternMatching	30	22	20	26
QualitativeSpatialReasoning	30	30	27	29
RicochetRobot	30	30	7	30
Sokoban	30	11	8	12
Solitaire	27	22	20	22
StableMarriage	30	29	27	29
VisitAll	30	19	11	19
Weighted-Sequence Problem	30	25	14	25
Total	447	312	222	306

Optimum answer set search

Problem	#	CLASP	WASP 2.1
CrossingMinim	30	26	22
Labyrinth	29	19	12
MaximalClique	30	30	29
MPSP	6	5	5
Sokoban	28	28	28
StillLife	10	7	5
WBDSS	29	12	12
Fastfood	29	15	19
OpenDoors	31	31	31
ValvesLocation	30	2	4
MaxSAT	42	18	28
Total	294	193	195

Cautious reasoning

Problem	#	CLASP	WASP 2.1
CQA-Q1	20	20	20
CQA-Q2	20	20	20
CQA-Q3	20	15	20
CQA-Q4	20	20	20
CQA-Q5	20	20	20
CQA-Q6	20	14	20
CQA-Q7	20	10	12
Total	140	119	132

Outline

Ongoing and future work

Ongoing work

- Support of non-HCF disjunction

Future work

- Implementation of different heuristics
- Implementation of brave reasoning

Conclusion

Implemented solutions for several computational tasks

- Model generation
  - Preprocessing, CDCL-like algorithm, incremental interface
- Optimum answer set search
  - Model-guided and core-guided algorithms
- Cautious reasoning
  - Algorithm iterative coherence testing
  - A framework of anytime algorithms
- Implementation of WASP 2.1
  - Comparable to the state-of-the-art solvers
  - Friendly Open Source License: Apache 2.0
  - `http://alviano.github.io/wasp/`

Conclusion

Implemented solutions for several computational tasks

- Model generation
  - Preprocessing, CDCL-like algorithm, incremental interface
- Optimum answer set search
  - Model-guided and core-guided algorithms
- Cautious reasoning
  - Algorithm iterative coherence testing
  - A framework of anytime algorithms
- Implementation of WASP 2.1
  - Comparable to the state-of-the-art solvers
  - Friendly Open Source License: Apache 2.0
  - `http://alviano.github.io/wasp/`

Thanks

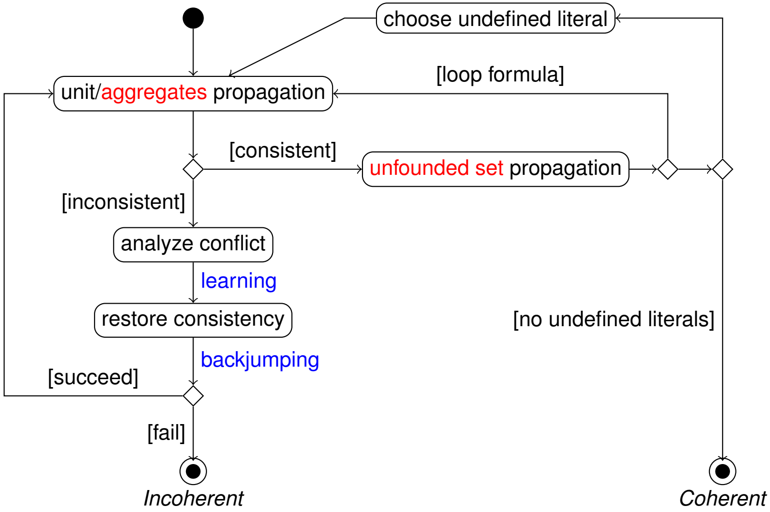
Bonus slides

Simplification of the program in the style of SATELITE

- Subsumption
  - -> A rule r_1 subsumes a rule r_2 if $r_1 \subseteq r_2$
  - -> r_2 can be deleted
- Self-subsumption
  - -> A rule r_1 self-subsumes a rule r_2 if there is a literal ℓ such that $\ell \in r_1$, $\neg\ell \in r_2$ and $r_1 \setminus \{\ell\} \subseteq r_2 \setminus \{\neg\ell\}$
  - -> r_2 can be strengthened by removing $\neg\ell$
- Literals elimination
  - -> Eliminates literals through rule distribution
  - -> Detects if there exists such situation $\ell \iff \ell_1 \wedge \ldots \wedge \ell_n$
  - -> Each occurrence of ℓ is substituted by $\ell_1 \wedge \ldots \wedge \ell_n$

Model Generator

CDCL-like backtracking algorithm

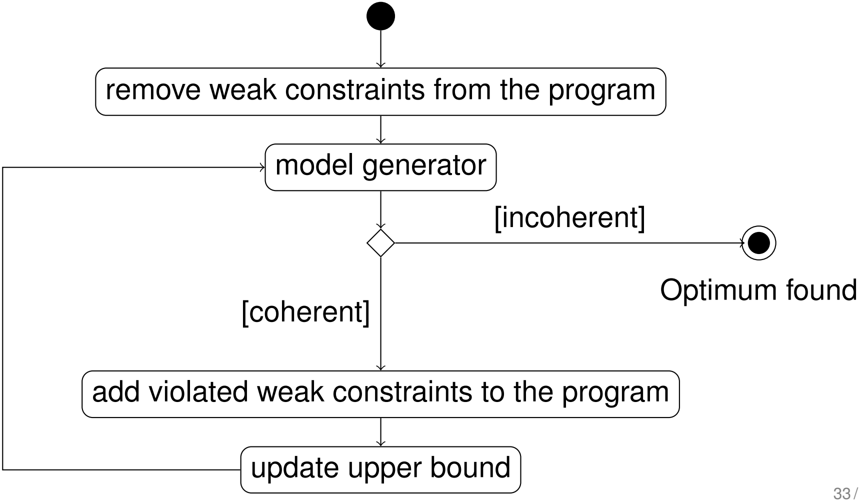


Optimum answer set search

- Model-guided algorithms: OPT, BASIC and MGD
  - \+ Easy to implement
  - \+ Work well on particular domains
  - \+ Produce non-optimum solutions during the search
  - \- Poor performances on industrial instances
- Core-guided algorithms: PMRES and OLL
  - \+ Good performances on industrial instances
  - \- Do not produce non-optimum solutions (in general)
  - \- The implementation is usually nontrivial
- Mixed algorithms: INTERLEAVING
  - \+ Exploit the properties of both strategies
  - \+ Produce non-optimum solutions during the search
  - \- Deterioration of the performance on some instances

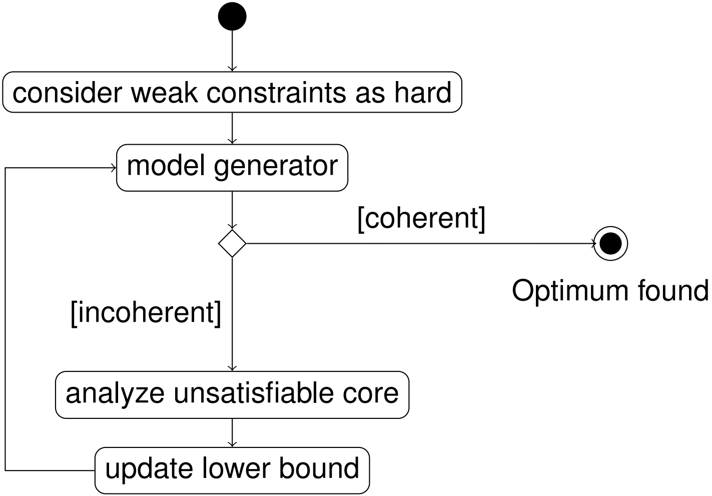
Model-guided algorithms

I need a solution! Give me any answer set

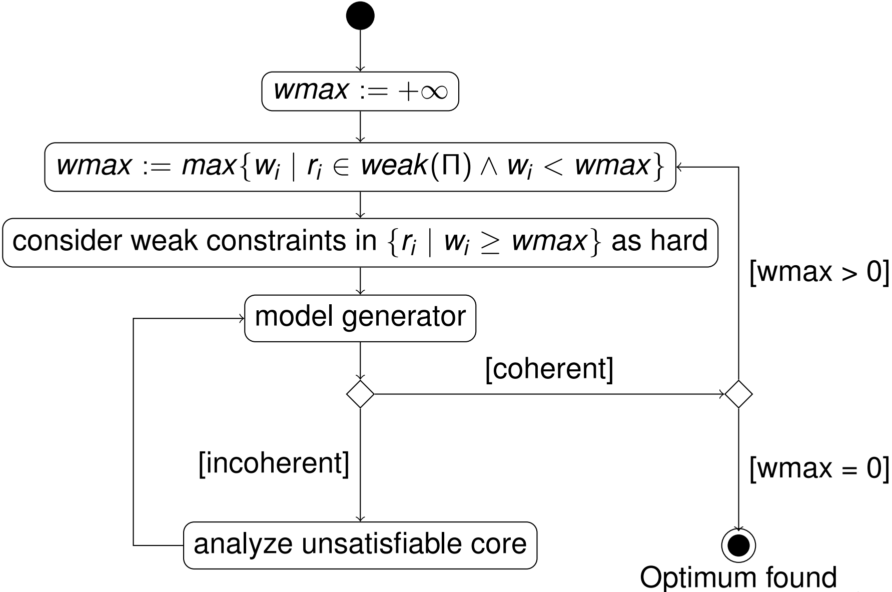


Core-guided algorithms

I feel lucky! Try to satisfy all weak constraints



Stratification



Impact of stratification

Problem	#	OLL	OLL NO_STRAT	PMRES	PMRES NO_STRAT
Fastfood	29	19	10	19	14
OpenDoors	31	31	31	31	31
ValvesLocation	30	4	2	4	2
MaxSAT	42	28	19	28	27
Total	132	82	62	82	74

Competition score: unweighted problems

Problem	OPT	BASIC	MGD	OLL	PMRES	INTERLEAVING
CrossingMinim	70	74	79	110	110	138
Labyrinth	23	116	89	60	40	85
MaximalClique	60	85	94	145	40	83
MPSP	27	26	26	25	25	30
Sokoban	125	125	125	125	125	125
StillLife	43	39	37	25	25	46
WBDSS	16	20	19	60	50	118
Fastfood	110	143	143	115	117	134
OpenDoors	155	155	155	155	155	155
ValvesLocation	10	118	116	101	101	92
MaxSAT	118	74	80	143	143	166
Total	756	973	963	1063	930	1170

CQA: encoding [Barceló and Bertossi, 2003]

Problem	#	CLASP	WASP 2.1
Query1	10	10	10
Query2	10	10	10
Query3	10	6	10
Query4	10	10	10
Query5	10	10	10
Query6	10	5	10
Query7	10	5	6
Total	70	56	66

CQA: encoding [Manna et al., 2013]

Problem	#	CLASP	WASP 2.1
Query1	10	10	10
Query2	10	10	10
Query3	10	9	10
Query4	10	10	10
Query5	10	10	10
Query6	10	9	10
Query7	10	5	6
Total	70	63	66
Total	140	119	132

Related work: WASP 1 vs WASP 2.1

Technique	WASP 1	WASP 2.1
Support propagation	Native	Clark's Completion
Unfounded set propagation	Source pointers	Source pointers
Optimum answer set search	OPT, MGD OLL, BCD	BASIC, OPT, MGD OLL, PMRES, INTERLEAVING Stratification
Cautious Reasoning	No	Overestimate reduction Iterative coherence testing Anytime variants!

Related work: CLASP vs WASP 2.1

Technique	CLASP	WASP 2.1
Support propagation	Clark's Completion	Clark's Completion
Unfounded set propagation	Source pointers	Source pointers
Optimum answer set search	BASIC OLL, PMRES	BASIC, OPT, MGD OLL, PMRES, INTERLEAVING Stratification
Cautious Reasoning	Overestimate reduction	Overestimate reduction Iterative coherence testing Anytime variants!