Blocking and Other Enhancements for Bottom-Up Model Generation Methods

Peter Baumgartner
National ICT Australia

Renate A. Schmidt
The University of Manchester

Motivation: Disproving

- **Disproving**
  - Show that a given (first-order) formula (with equality) is not valid
  - This can be done by computing a model, i.e. a counterexample
- **Applications of Disproving**
  - Mathematics
    - Refute conjectures
    - Finite group existence
  - Verification: disproving verification conditions
  - Knowledge representation
    - Knowledge base is consistent
    - Speculated subsumption relation does not hold

Existing methods? Limits? What's new here?

Disproving Methods (1)

- **Finite model building**
  - Assume a fixed, finite domain { $d_1, \ldots, d_n$ }
  - Decide if there is a model of the given formula over that domain
  - If not, add a new domain element and repeat
- **Methods**
  - MACE-style: by reduction to
    - propositional SAT (Paradox, Mace2) or
    - function-free clause logic (FM-Darwin)
  - SEM-style: guess function tables and check for model
  - (Tableaux) algorithms by Bry&Torge, Bezem, Nivelle&Meng

✓ No syntactic restrictions on input formula
✗ Finite models sometimes not sufficient
✗ Poor scaling

Finite Model Builders - Scaling Problems

- Consider the clause set consisting of the O(n^2) unit clauses

$$\mathsf{p}(c_1, \ldots, c_n)$$
$$\neg\mathsf{p}(x_1, \ldots, x_{i-1},\ x,\ x_{i+1}, \ldots, x_{j-1},\ x,\ x_{j+1}, \ldots, x_n) \quad \text{for all } 1 \leq i < j \leq n$$

- Second clause says no c_i and c_j can be mapped to the same element
  - Therefore, smallest model has n domain elements
- 10^9 instances for n=10
  - For which n do current finite model finders give up?
- Any resolution method will terminate here

Finite model builders / (our) resolution methods are rather different
Our approach doesn't iterate on domain size
Our approach doesn't identify different constants

Disproving Methods (2)

- **Identify decidable fragment of FOL**
  - Guarded Fragment
  - Description and modal logics
  - Positive-variable dominated clauses
  - Prefix-classes: $\exists^*\forall^*$, $\exists^*\forall\exists^*$, ...
- **Design decision procedure for it**
  - From scratch. E.g. tableaux algorithms for description logics
  - By showing that a certain (resolution) refinement decides it. E.g. with axiomatic translation [Schmidt&Hustadt 2005], ordered resolution + splitting decides many modal logics

✓ **Powerful**

✗ **Resolution not practical for $\exists^*\forall^*$**

Really ?

Problems of Using Resolution for ∃*∀*

- ∃*∀* fragment corresponds to function-free clause logic
  - Important for many database-like applications (Datalog)
- Pathological example for resolution:

$$\text{Res}\ \frac{\mathsf{p}(x,y) \vee \mathsf{p}(y,z) \leftarrow \mathsf{p}(x,z) \qquad \mathsf{p}(\mathsf{a},z) \vee \mathsf{p}(z,\mathsf{a})}{\mathsf{p}(\mathsf{a},y) \vee \mathsf{p}(y,z) \vee \mathsf{p}(z,\mathsf{a})}$$

Derived clauses pattern:

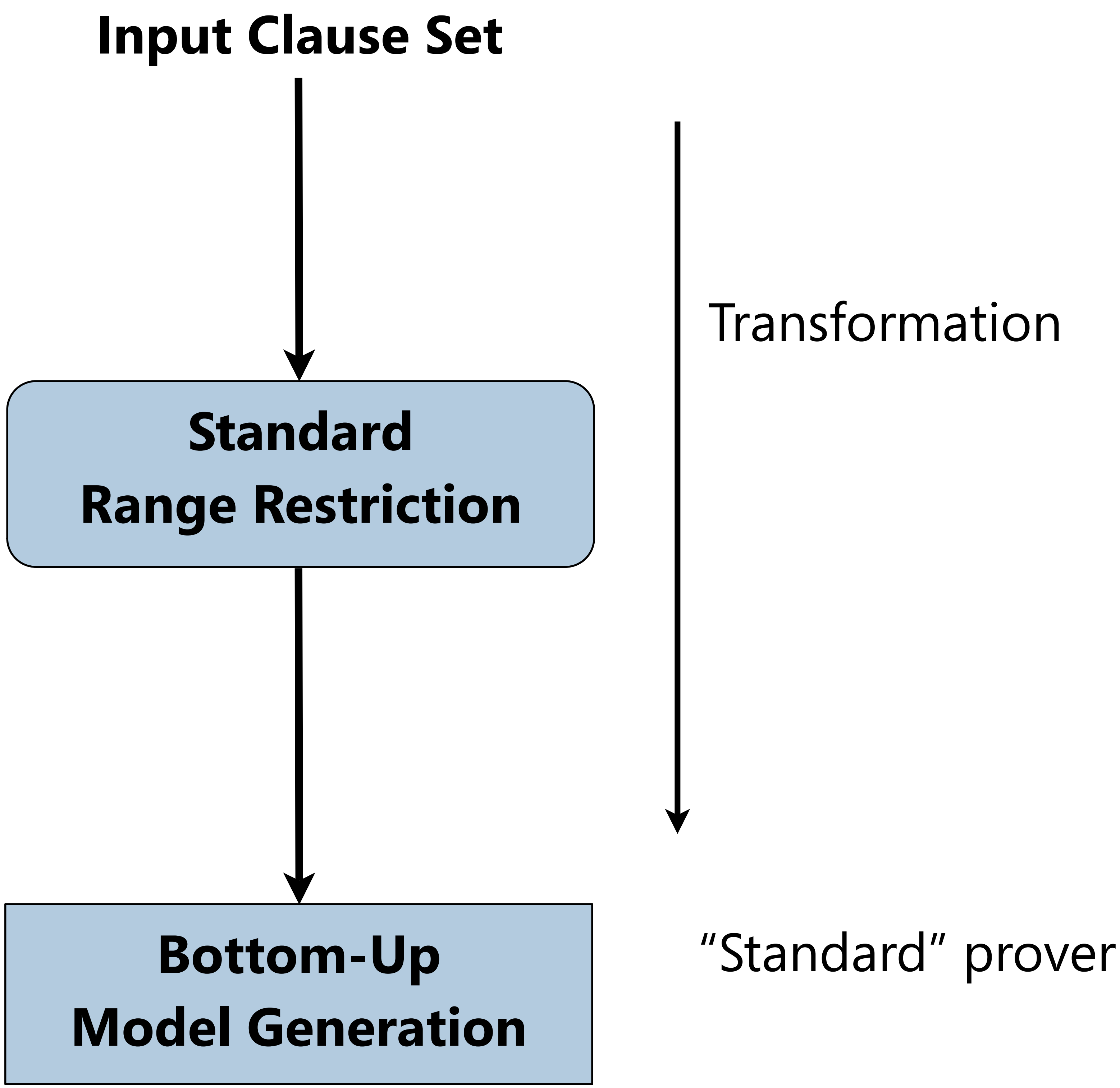
$$\begin{array}{r} \mathsf{p}(\mathsf{a},z) \vee \mathsf{p}(z,\mathsf{a}) \\ \mathsf{p}(\mathsf{a},y) \vee \mathsf{p}(y,z) \vee \mathsf{p}(z,\mathsf{a}) \\ \mathsf{p}(\mathsf{a},x) \vee \mathsf{p}(x,y) \vee \mathsf{p}(y,z) \vee \mathsf{p}(z,\mathsf{a}) \\ \vdots \end{array}$$

✗ **Refinements like subsumption, condensing, splitting don't help**

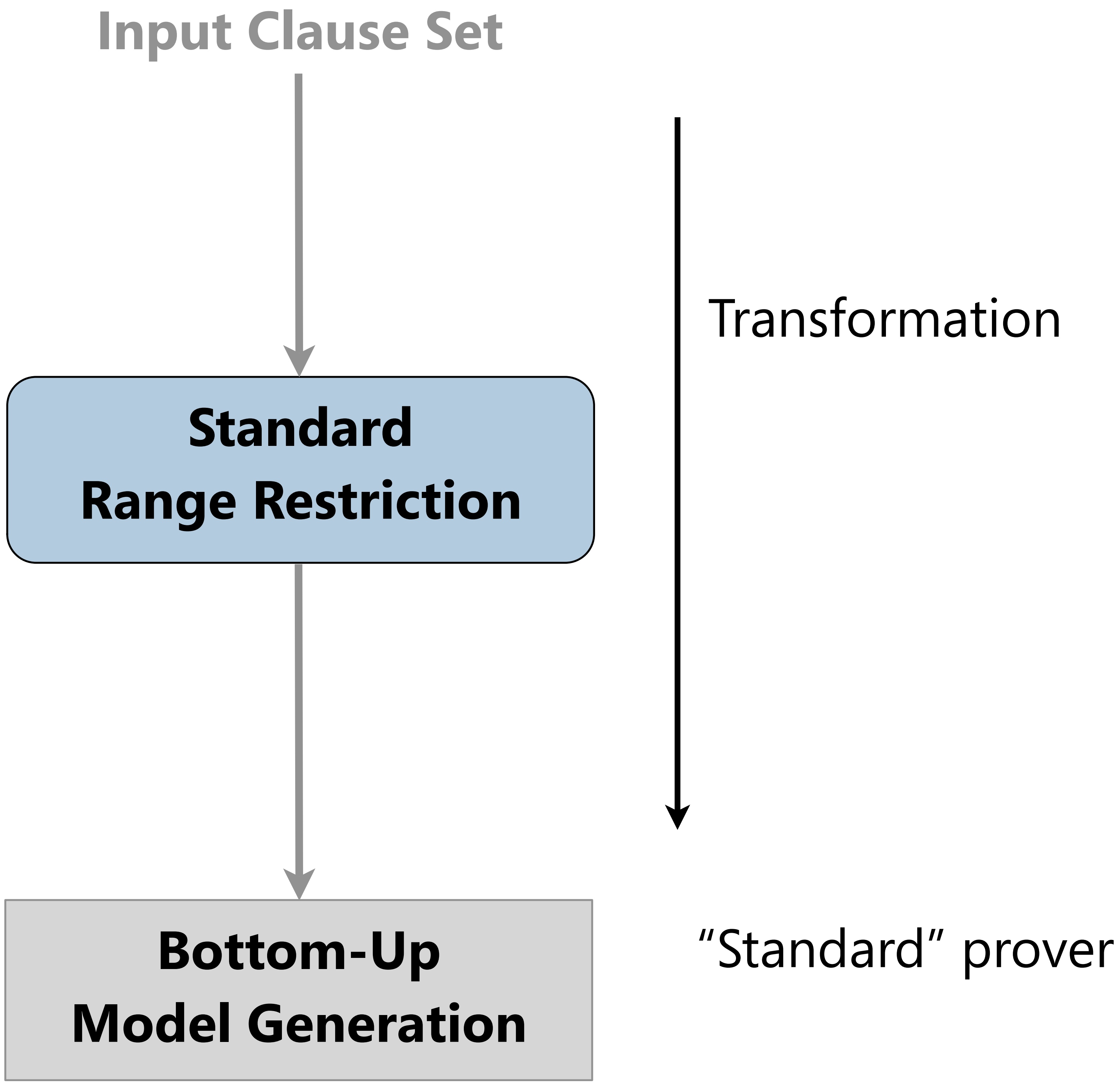
✗ **Hyperresolution + range restriction works, but inefficiently**

✓ **(One) contribution here: improved range restriction**

Classical Approach



Classical Approach



Standard Range Restriction [Manthey & Bry 88]

- A clause is **range restricted** iff each variable in its head also occurs in its body, as in $\mathsf{p}(x, z) \vee \mathsf{p}(z, \mathsf{a}) \leftarrow \mathsf{q}(x, z)$
- Every clause (set) can be made range restricted:
  - Restrict all extra variables in head in all input clauses to dom, e.g.

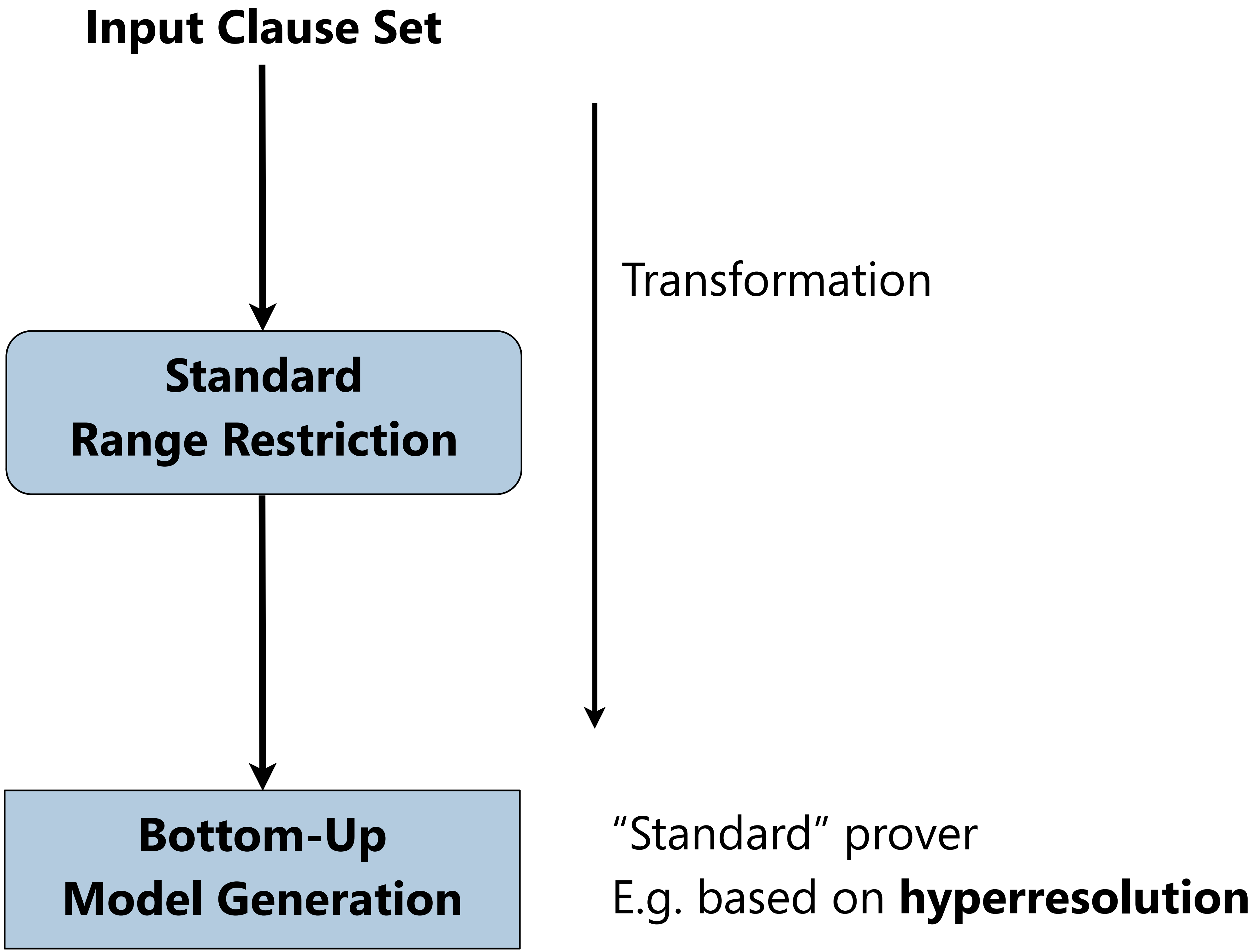
$$\mathsf{p}(x, z) \vee \mathsf{p}(z, \mathsf{a}) \quad \text{becomes} \quad \mathsf{p}(x, z) \vee \mathsf{p}(z, \mathsf{a}) \leftarrow \mathsf{dom}(x) \wedge \mathsf{dom}(z)$$

  - Add "dom"-clauses to enumerate Herbrand universe:

$$\begin{aligned} \mathsf{dom}(\mathsf{a}) & \\ \mathsf{dom}(\mathsf{b}) & \\ \mathsf{dom}(f(x_1, \ldots, x_n)) & \leftarrow \mathsf{dom}(x_1) \wedge \cdots \wedge \mathsf{dom}(x_n) \end{aligned}$$

All positive clauses derived by hyperresolution are ground

Classical Approach



Hyperresolution + Range Restriction + Splitting

Given clause set:

$$\mathsf{p}(\mathsf{a}, z) \lor \mathsf{p}(z, \mathsf{a})$$

$$\mathsf{p}(x, y) \lor \mathsf{p}(y, z) \quad \leftarrow \quad \mathsf{p}(x, z)$$

Derivation by hyperresolution + splitting after range restriction

$$\text{H-Res}\ \dfrac{\mathsf{p}(\mathsf{a}, z) \lor \mathsf{p}(z, \mathsf{a}) \leftarrow \mathsf{dom}(z) \qquad \mathsf{dom}(b)}{\text{Split}\ \dfrac{\mathsf{p}(\mathsf{a}, \mathsf{b}) \lor \mathsf{p}(\mathsf{b}, \mathsf{a})}{\boxed{\mathsf{p}(\mathsf{a}, \mathsf{b})} \quad \mid \quad \mathsf{p}(\mathsf{b}, \mathsf{a})}}$$

$$\text{H-Res}\ \dfrac{\mathsf{p}(x, y) \lor \mathsf{p}(y, z) \leftarrow \mathsf{dom}(y) \land \mathsf{p}(x, z) \qquad \mathsf{dom}(a) \qquad \mathsf{p}(\mathsf{a}, \mathsf{b})}{\text{Split}\ \dfrac{\mathsf{p}(\mathsf{a}, \mathsf{a}) \lor \mathsf{p}(\mathsf{a}, \mathsf{b})}{\boxed{\mathsf{p}(\mathsf{a}, \mathsf{a})} \quad \mid \quad \mathsf{p}(\mathsf{a}, \mathsf{b})}}$$

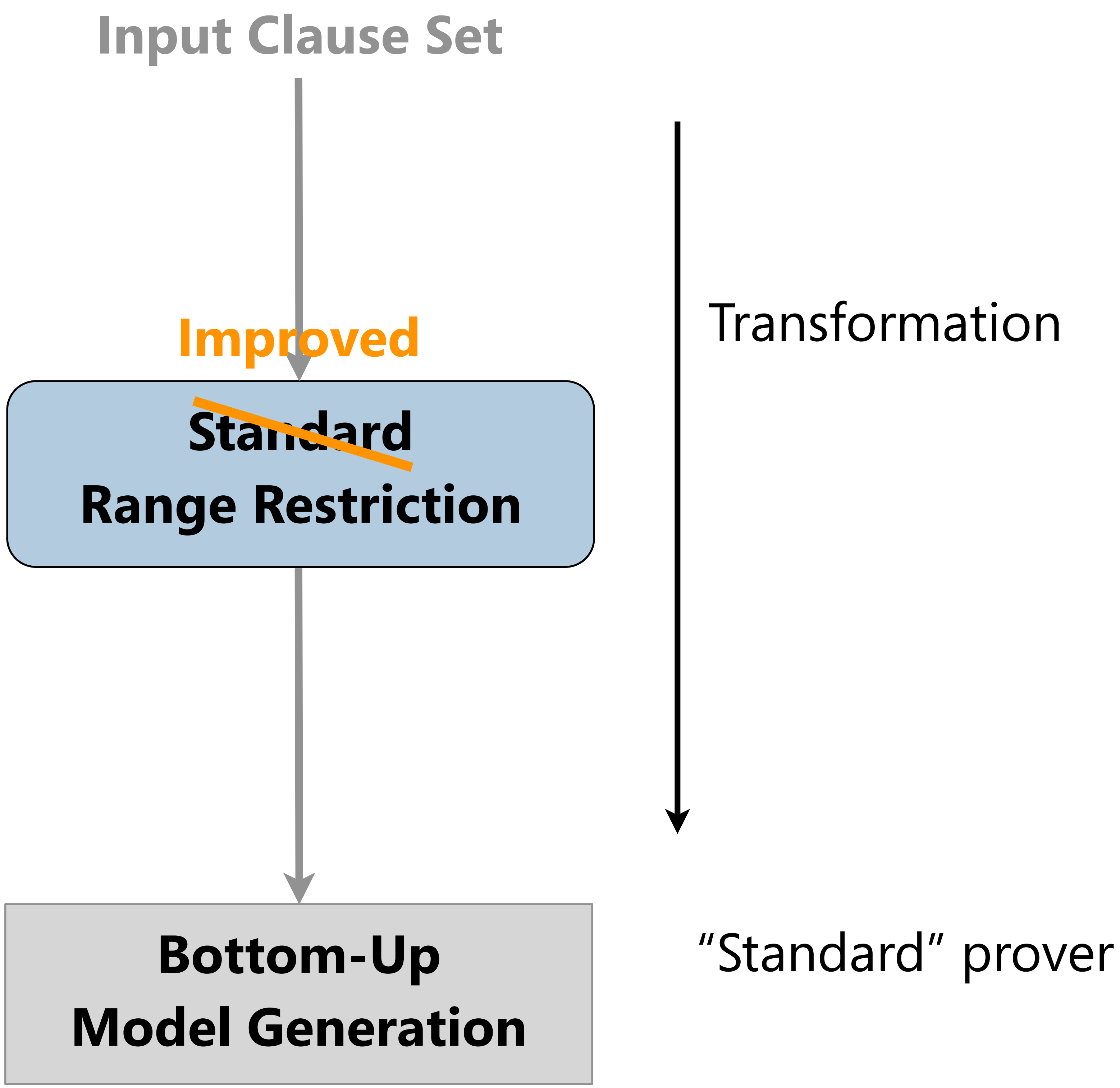
$\boxed{\text{Model}}$

✓ **Decides function-free clause logic (BS class)**

✗ **Search space too big**

Improvement?

Our Approach



Improved Range Restriction

Here: idea by means of an example, see paper for details

(1) Domain elements from clause heads

Clause	Transformation	
	$\mathsf{dom}(\mathsf{a})$	for some constant a
$\mathsf{P}(x) \vee \mathsf{Q}(\mathsf{b})$	$\mathsf{P}(x) \vee \mathsf{Q}(\mathsf{b}) \leftarrow \mathsf{dom}(x)$	
	$\mathsf{dom}(\mathsf{x}) \leftarrow \mathsf{P}(x)$	for each head
	$\mathsf{dom}(\mathsf{x}) \leftarrow \mathsf{Q}(x)$	predicate symbol

✓ **May yield smaller domain, depending on splits chosen**

Improved Range Restriction

(2) Domain elements from clause bodies

Clause	**Transformation**	
	$\mathsf{dom}(\mathsf{a})$	for some constant a
$\mathsf{P}(x)$	$\mathsf{P}(x) \leftarrow \mathsf{dom}(x)$	
	$\mathsf{dom}(x) \leftarrow \mathsf{P}(x)$	
$\perp \leftarrow \mathsf{P}(\mathsf{a}) \wedge \mathsf{P}(\mathsf{b})$	$\perp \leftarrow \mathsf{P}(\mathsf{a}) \wedge \mathsf{P}(\mathsf{b})$	
	$\mathsf{dom}(\mathsf{a}) \leftarrow \mathsf{P}(x)$	for each body
	$\mathsf{dom}(\mathsf{b}) \leftarrow \mathsf{P}(x)$	predicate symbol

✓ **May yield smaller domain, depending on satisfied literals**

Improved Range Restriction

(2) Domain elements from clause bodies

Clause	**Transformation**	
	$\mathsf{dom}(\mathsf{a})$	for some constant a
$\mathsf{P}(x)$	$\mathsf{P}(x) \leftarrow \mathsf{dom}(x)$	
	$\mathsf{dom}(x) \leftarrow \mathsf{P}(x)$	
$\bot \leftarrow \mathsf{Q}(\mathsf{a}) \land \mathsf{Q}(\mathsf{b})$	$\bot \leftarrow \mathsf{Q}(\mathsf{a}) \land \mathsf{Q}(\mathsf{b})$	
	$\mathsf{dom}(\mathsf{a}) \leftarrow \mathsf{Q}(x)$	for each body
	$\mathsf{dom}(\mathsf{b}) \leftarrow \mathsf{Q}(x)$	predicate symbol

✓ **May yield smaller domain, depending on satisfied literals**

Soundness and Completeness

- rr(M) := transformation of clause set M into range-restricted form
- **Proposition**
  A clause set M is satisfiable iff rr(M) is satisfiable
  Proof (completeness):
  - Given a Herbrand model I_{rr} of rr(M).
  - Define Interpretation I for M:
    - Domain $|I|$ = { t | $I_{\text{rr}} \models$ dom(t) }
    - Terms in $|I|$ evaluate to themselves ("Quasi-Herbrand")
  - Show that I is a model of M : ...
- **Corollary**
  A clause set M is E-satisfiable iff
  rr(M) $\cup$ { $x \approx x \leftarrow$ dom(x) } is E-satisfiable
  Proof: Use equality axioms. Only equality axiom affected is reflexivity

Problem with Improved Range Restriction

- **Problem:** function symbols in clause bodies may lead to non-termination of BUMG
- **Example:**
  From $\mathsf{r}(x) \leftarrow \mathsf{q}(x) \wedge \mathsf{p}(\mathsf{f}(x))$

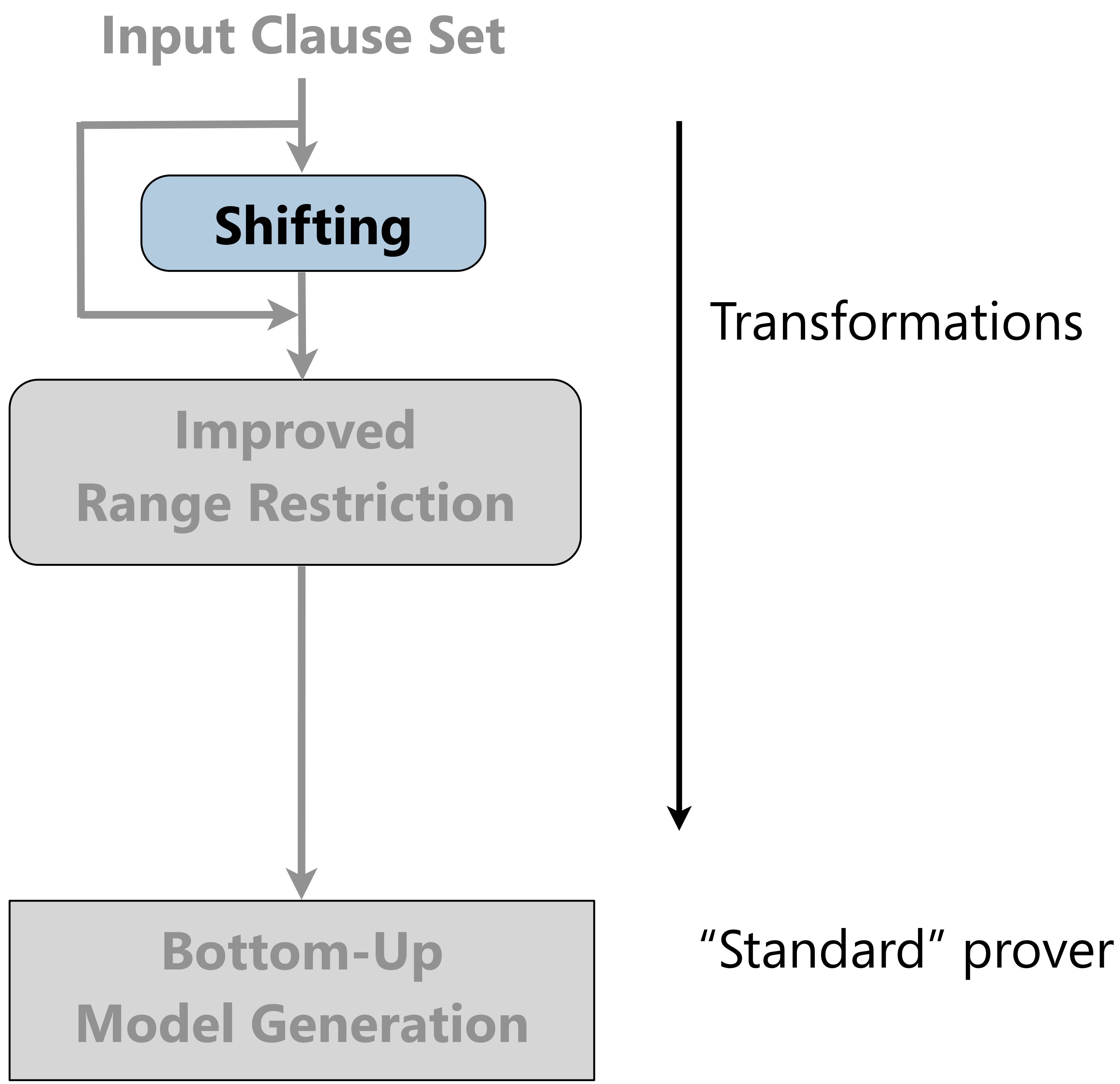
  obtain $\mathsf{dom}(\mathsf{f}(x)) \leftarrow \mathsf{p}(y)$

  and finally $\mathsf{dom}(\mathsf{f}(x)) \leftarrow \mathsf{dom}(x) \wedge \mathsf{p}(y)$
- Together with p(b), q(a),
  dom(a)

  derive dom(f(a))
  dom(f(f(a)))
  dom(f(f(f(a))))
  …

The "shifting" transformation avoids this problem

Our Approach



Shifting

- Moves body literals with function terms into the head:

$$\mathrm{r}(x) \leftarrow \mathrm{q}(x) \wedge \mathrm{p}(\mathrm{f}(x))$$

Shifting:

$$\mathrm{r}(x) \vee \mathrm{not\_p}(\mathrm{f}(x)) \leftarrow \mathrm{q}(x)$$

$$\bot \leftarrow \mathrm{p}(\mathrm{x}) \wedge \mathrm{not\_p}(\mathrm{x})$$

- Advantage: criticial head atom not_p(f(x)) possibly avoided now:
- With, say p(b), q(a)

  dom(a)

  derive r(a) $\vee$ not_p(f(a))

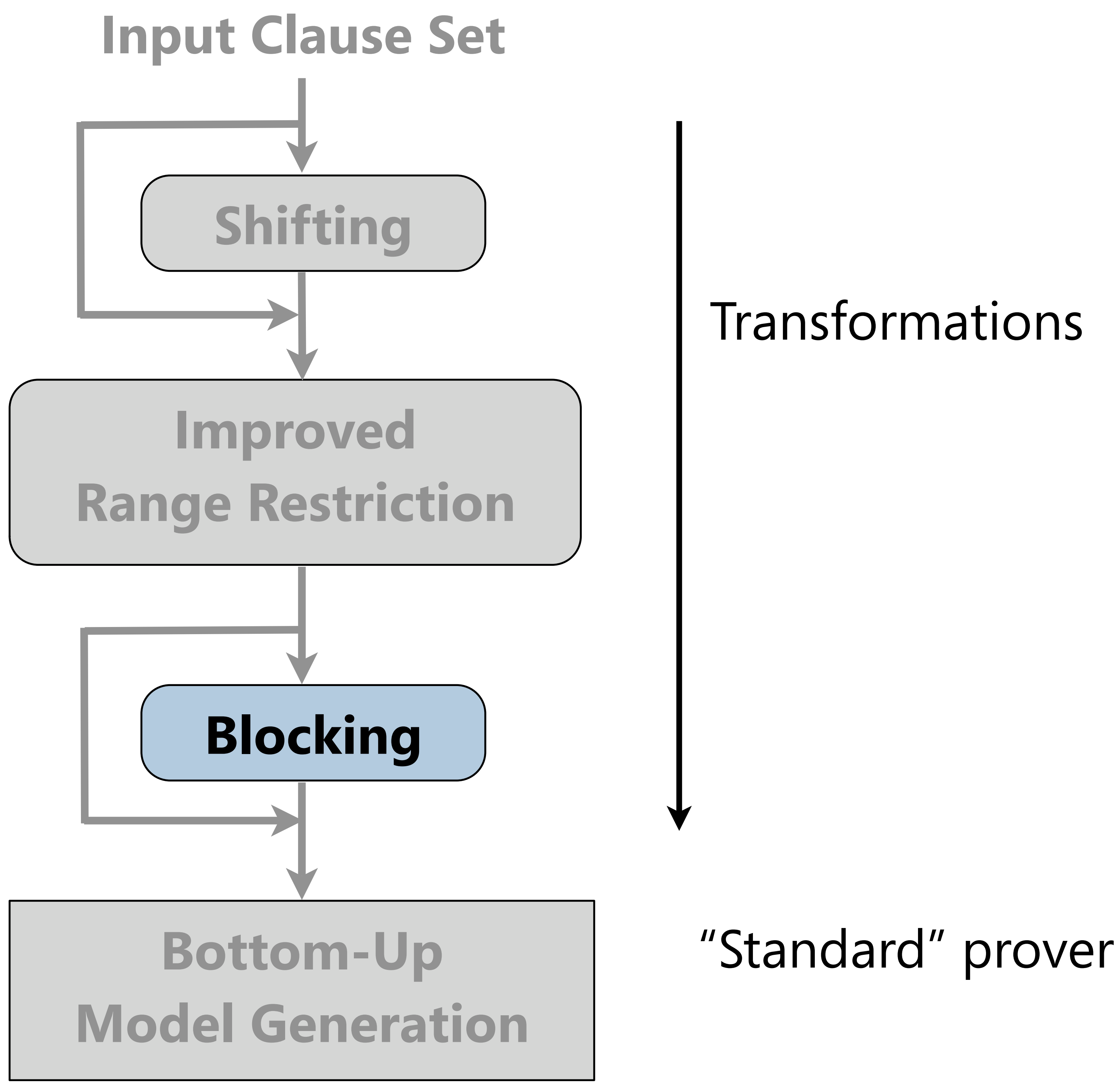
  then split r(a)

  and done

✓ Improved RR + Shifting already quite effective

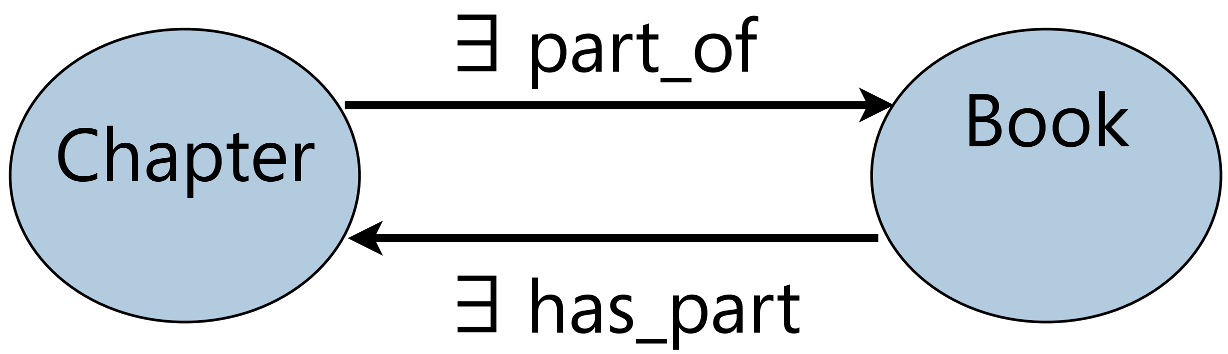
Next: going beyond ∃*∀* by "blocking"

Our Approach



Blocking: Idea

- Detect periodicity in models and achieve termination by exploiting standard redundancy criteria
- Example from Tambis KB



$$
\begin{aligned}
\mathsf{Chapter}(\mathsf{a}) & \\
\mathsf{Book}(\mathsf{f}_{\mathsf{Book}}(x)) &\leftarrow \mathsf{Chapter}(x) \\
\mathsf{Chapter}(\mathsf{f}_{\mathsf{Chapter}}(x)) &\leftarrow \mathsf{Book}(x) \\
\bot &\leftarrow \mathsf{Chapter}(x) \wedge \mathsf{Book}(x)
\end{aligned}
$$

- BUMG without blocking derives infinite model:

$$\{\mathsf{Chapter}(\mathsf{a}),\ \mathsf{Book}(\mathsf{f}_{\mathsf{Book}}(\mathsf{a})),\ \mathsf{Chapter}(\mathsf{f}_{\mathsf{Chapter}}(\mathsf{f}_{\mathsf{Book}}(\mathsf{a}))), \ldots\}$$

- But same model represented finitely by

$$\{\mathsf{Chapter}(\mathsf{a}),\ \mathsf{Book}(\mathsf{f}_{\mathsf{Book}}(\mathsf{a}))\} \quad \text{and} \quad \mathsf{f}_{\mathsf{Chapter}}(\mathsf{f}_{\mathsf{Book}}(\mathsf{a})) \approx \mathsf{a}$$

Blocking transformation encodes this search for equations

Blocking

- If y is a subterm of x then speculate $x \approx y$ - or not:

$$x \approx y \lor x \not\approx y \leftarrow \mathsf{sub}(x,y)$$
$$\bot \leftarrow x \approx y \land x \not\approx y$$

  To be effective, BUMG must consider the case $x \approx y$ first

- The subterm relation, restricted to dom elements:

$$\mathsf{sub}(x,x) \leftarrow \mathsf{dom}(x)$$
$$\mathsf{sub}(x, f(x_1,\ldots,x_n)) \leftarrow \mathsf{sub}(x,x_i) \land \mathsf{dom}(x) \land \mathsf{dom}(f(x_1,\ldots,x_n))$$

  for every n-ary function symbol $f \in \Sigma_f$ and all $i \in \{1,\ldots,n\}$

- This way, say, dom(f(a)) will be simplified to dom(a) when equation f(a) $\approx$ a has been speculated

Never equates two constants, search limited to subterms in domain

Experiments

- TPTP Version 3.1.1, tried all 514 clausal satisfiable problems
- Main prover: slightly modified superposition prover MSPASS
- Environment: Linux PC, Intel Pentium 4, 3.8 GHz, 1 GByte
- Timeout 5 minutes
  Memory limit 300 MByte (never a problem for MSPASS and KRHyper)
- Results: MSPASS + our transformations vs. ...
  - ... SPASS auto mode: orthogonal
  - ... Paradox: about 20 problems unsolvable for Paradox that can be solved by our methods
- Next slides:
  - Detailed evaluation of MSPASS + our transformations
  - On non-equational problems also tried KRHyper

MSPASS on Satisfiable TPTP Problems

Category	#	rr $-$sp	rr $+$sp	sh $\circ$ rr $-$sp	sh $\circ$ rr $+$sp	rr $\circ$ bl $+$sp	sh $\circ$ rr $\circ$ bl $+$sp	crr $\circ$ bl $+$sp
ALG	1	0	0	0	0	1	0	0
BOO	13	0	0	0	0	2	3	2
COL	5	0	0	0	0	0	0	0
GEO	1	0	0	0	0	0	0	0
GRP	25	7	7	7	8	15	14	12
KRS	8	1	1	4	8	4	6	4
LAT	1	0	0	0	0	1	1	0
LCL	4	0	1	1	1	1	1	1
MGT	10	1	1	3	4	4	5	0
MSC	1	1	1	1	1	1	1	1
NLP	236	49	79	68	96	87	160	68
NUM	1	1	1	1	1	1	1	1
PUZ	20	6	6	6	6	10	10	9
RNG	4	0	0	0	0	0	0	0
SWV	8	0	0	0	0	1	1	0
SYN	176	20	50	20	52	124	125	120
All	514	86	147	111	177	252	328	218

MSPASS on Satisfiable TPTP Problems

Category	#	rr −sp	rr +sp	sh ∘ rr −sp	sh ∘ rr +sp	rr ∘ bl +sp	sh ∘ rr ∘ bl +sp	crr ∘ bl +sp
ALG	1	0	0	0	0	1	[illegible]	0
BOO	13	0	0	0	0	2	3	2
COL	5	0	0	0	0	0	0	0
GEO	1	0	0	0	0	0	0	0
GRP	25	7	7	7	8	15	14	12
KRS	8	1	1	4	8	4	6	4
LAT	1	0	0	0	0	1	1	0
LCL	4	0	1	1	1	1	1	1
MGT	10	1	1	3	4	4	5	0
MSC	1	1	1	1	1	1	1	1
NLP	236	49	79	68	96	87	160	68
NUM	1	1	1	1	1	1	1	1
PUZ	20	6	6	6	6	10	10	9
RNG	4	0	0	0	0	0	0	0
SWV	8	0	0	0	0	1	1	0
SYN	176	20	50	20	52	124	125	120
All	514	86	147	111	177	252	328	218

Splitting is advisable

MSPASS on Satisfiable TPTP Problems

Category	#	rr −sp	rr +sp	sh ∘ rr −sp	sh ∘ rr +sp	rr ∘ bl +sp	sh ∘ rr ∘ bl +sp	crr ∘ bl +sp
ALG	1	0	0	0	0	1	0	0
BOO	13	0	0	0	0	2	3	2
COL	5	0	0	0	0	0	0	0
GEO	1	0	0	0	0	0	0	0
GRP	25	7	7	7	8	15	14	12
KRS	8	1	1	4	8	4	6	4
LAT	1	0	0	0	0	1	1	0
LCL	4	0	1	1	1	1	1	1
MGT	10	1	1	3	4	4	5	0
MSC	1	1	1	1	1	1	1	1
NLP	236	49	79	68	96	87	160	68
NUM	1	1	1	1	1	1	1	1
PUZ	20	6	6	6	6	10	10	9
RNG	4	0	0	0	0	0	0	0
SWV	8	0	0	0	0	1	1	0
SYN	176	20	50	20	52	124	125	120
All	514	86	147	111	177	252	328	218

sh ∘ rr and rr ∘ bl orthogonal

MSPASS on Satisfiable TPTP Problems

$sh \circ rr \circ bl$ generally best

Category	#	rr −sp	rr +sp	$sh \circ rr$ −sp	$sh \circ rr$ +sp	$rr \circ bl$ +sp	$sh \circ rr \circ bl$ +sp	$crr \circ bl$ +sp
ALG	1	0	0	0	0	1	0	0
BOO	13	0	0	0	0	2	3	2
COL	5	0	0	0	0	0	0	0
GEO	1	0	0	0	0	0	0	0
GRP	25	7	7	7	8	15	14	12
KRS	8	1	1	4	8	4	6	4
LAT	1	0	0	0	0	1	1	0
LCL	4	0	1	1	1	1	1	1
MGT	10	1	1	3	4	4	5	0
MSC	1	1	1	1	1	1	1	1
NLP	236	49	79	68	96	87	160	68
NUM	1	1	1	1	1	1	1	1
PUZ	20	6	6	6	6	10	10	9
RNG	4	0	0	0	0	0	0	0
SWV	8	0	0	0	0	1	1	0
SYN	176	20	50	20	52	124	125	120
All	514	86	147	111	177	252	328	218

MSPASS on Satisfiable TPTP Problems

Category	#	rr −sp	rr +sp	sh ∘ rr −sp	sh ∘ rr +sp	rr ∘ bl +sp	sh ∘ rr ∘ bl +sp	crr ∘ bl +sp
ALG	1	0	0	0	0	1	0	0
BOO	13	0	0	0	0	2	3	2
COL	5	0	0	0	0	0	0	0
GEO	1	0	0	0	0	0	0	0
GRP	25	7	7	7	8	15	14	12
KRS	8	1	1	4	8	4	6	4
LAT	1	0	0	0	0	1	1	0
LCL	4	0	1	1	1	1	1	1
MGT	10	1	1	3	4	4	5	0
MSC	1	1	1	1	1	1	1	1
NLP	236	49	79	68	96	87	160	68
NUM	1	1	1	1	1	1	1	1
PUZ	20	6	6	6	6	10	10	9
RNG	4	0	0	0	0	0	0	0
SWV	8	0	0	0	0	1	1	0
SYN	176	20	50	20	52	124	125	120
All	514	86	147	111	177	252	328	218

sh ∘ rr ∘ bl much better than crr ∘ bl

KRHyper on Satisfiable non-Equational Problems

Rating	#	MSPASS	KRHyper additional	Rating	#	MSPASS	KRHyper additional
1.00	4	0		0.40	47	26	1
0.80	57	24	4	0.33	8	4	1
0.67	26	5		0.20	70	50	
0.60	44	23	10	0.17	31	10	
0.50	5	0		0.00	223	198	1

Conclusions

- Various improvements to BUMG paradigm, based on
  - Shifting, improved range restriction, blocking
  - Hyperresolution + splitting
    - State-of-the-art equality inference rules
    - Standard notion of redundancy
- Improves model building capabilities of standard BUMG provers
  - E.g. MSPASS, KRHyper, but not limited to these
  - Method generates domain elements on a by need basis
  - Never identifies constants (unlike finite model finders)
- Future work
  - Sorts
  - Nonmonotonic reasoning