



---

# Superposition for Finite Domains

Thomas Hillenbrand · Christoph Weidenbach

Max-Planck-Institut für Informatik  
Saarbrücken



# Introduction (1)

- Setting: domain size *explicitly* bounded from above
- Domain elements represented by *digits*  $1, 2, \dots, n$   
these constants *not necessarily distinct*
- Detect (un)satisfiability of clause sets modulo theory  $\mathcal{T}$ :

$$\forall x. x \simeq 1 \vee \dots \vee x \simeq n$$

- Motivation for this study:
  - *not*: competing with finite-domain model generators
  - *rather*: finite domains in combination with infinite ones:  
enumeration types, components with finite state space
  - besides: clarify what can be achieved with superposition



## Introduction (2)

- Finite domains a *weak point* of superposition-based provers
- Example: Saturate nothing but  $x \simeq 1 \vee x \simeq \underline{2}$ :

Produces  $x \simeq 1 \vee x \simeq y \vee y \simeq \underline{1}$

from there  $x \simeq 1 \vee x \simeq y \vee y \simeq v \vee u \simeq 1 \vee u \simeq v$

⋮

in general  $x_1 \simeq 1 \vee \left( \bigvee_{i=1}^{2^k-1} x_i \simeq x_{i+1} \right) \vee x_{2^k} \simeq 1$

⋮

*Infinite* band of clauses, in spite of principal *decidability!*



# Standard Lifting Revisited (1)

- On ground level:

- ground *inference* rules like:

$$\text{SpL} \quad \frac{C \vee l \simeq r \quad s[l] \not\approx t \vee D}{C \vee s[r] \not\approx t \vee D} \quad \text{if} \cdot \begin{array}{l} l, l \simeq r, s \text{ and } s \not\approx t \text{ are} \\ \text{(strictly) maximal} \end{array}$$

- *redundancy* notion for inferences and clauses
- *model functor* ensures refutational *completeness*

- On non-ground level:

- *lifted* inference rules, say:

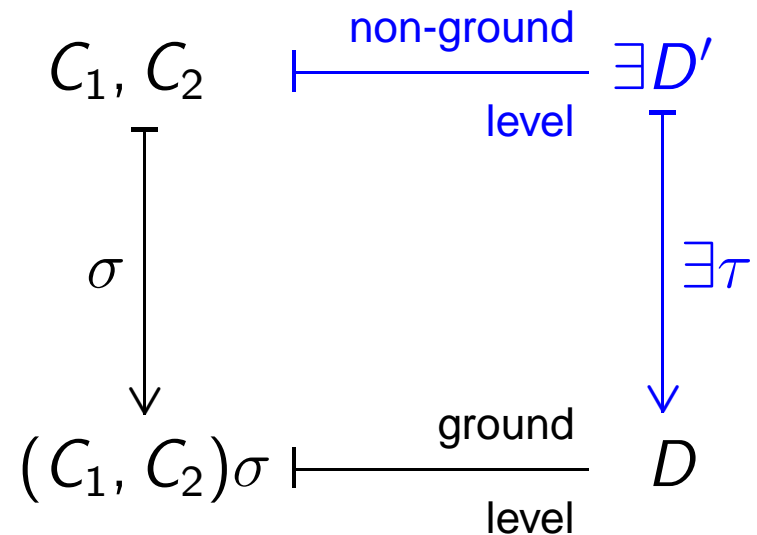
$$\text{SpL} \quad \frac{C \vee l \simeq r \quad s[l'] \not\approx t \vee D}{(C \vee s[r] \not\approx t \vee D)\sigma} \quad \text{if} \cdot \begin{array}{l} \sigma = \text{mgu}(l, l') \text{ and } l' \notin \mathcal{V} \\ l, l \simeq r, s \text{ and } s \not\approx t \text{ are} \\ \text{(strictly) maximal under } \sigma \end{array}$$

- redundancy: via *ground instances* reduced to ground level



## Standard Lifting Revisited (2)

- Completeness via *lifting lemma*:  
Every non-redundant inference from ground instances  $\vec{C}\sigma$  is an instance of an inference from the non-ground clauses  $\vec{C}$ .



- Thereby*: If non-ground clause set is saturated, then set of its ground instances is also saturated.
- Propose *refinement* in case of *finite domains*:  
Instead of instantiating with *all ground terms*, restrict to *digits*  $1, 2, \dots, n$



# Superposition for Finite Domains (1)

- Must exchange cardinality bound  $x \simeq 1 \vee \dots \vee x \simeq n$  for range axioms  $f(\vec{x}) \simeq 1 \vee \dots \vee f(\vec{x}) \simeq n$ .  
Furthermore: Digits as smallest ground terms.

- May *restrict inferences dramatically*:

*No complex unifiers* needed!

$$\text{SpL} \quad \frac{C \vee l \simeq r \quad s[l'] \not\approx t \vee D}{(C \vee s[r] \not\approx t \vee D)\sigma}$$

if

- $\sigma = \text{mgu}(l, l')$  and  $l' \notin \mathcal{V}$
- $\text{ran } \sigma \subseteq \mathcal{V} \cup [1; n]$
- $l, l \simeq r, s$  and  $s \not\approx t$  are (strictly) maximal under  $\sigma$

- Can establish lifting lemma for modified calculus.

**Theorem 1:** Restriction preserves *refutational completeness*.

# Superposition for Finite Domains (2)



- *Additional* calculus modification possible:  
*Lift exactly decreasing* inferences!

$$\text{SpL} \quad \frac{C \vee l \simeq r \quad s[l'] \not\approx t \vee D}{(C \vee s[r] \not\approx t \vee D)\sigma\tau} \quad \text{if} \quad \begin{array}{l} \cdot \sigma = \text{mgu}(l, l') \text{ and } l' \notin \mathcal{V} \\ \cdot \text{ran } \sigma \subseteq \mathcal{V} \cup [1; n], \text{ ran } \tau = [1; n] \\ \cdot \tau \text{ minimal such that} \\ \quad l, l \simeq r, s \text{ and } s \not\approx t \text{ are} \\ \quad \text{(strictly) greatest under } \sigma\tau \end{array}$$

- Number of inference conclusions can increase, but lifting in a sense *more economical*.

*Theorem 2:* Modification preserves *refutational completeness*.

# Superposition for Finite Domains (3)



- Non-ground redundancy notion *different* from standard one: Instances in terms of *digits* instead of *all ground terms*.  
Is  $f(g(1)) \simeq 1$  redundant wrt.  $f(x) \simeq 1$ ? **No!**  
But **Yes!** in presence of  $g(x) \simeq 1 \vee \dots \vee g(x) \simeq n$ .

**Theorem 3:** In order to show all digit instances of  $C$  redundant wrt.  $N$ , one may use:

– all ground instances from  $N$  unless ...

some  $C_\rho$  has greatest term  $f(\vec{i})$  such that  $C_\rho \preceq \bigvee_{j=1}^n f(\vec{i}) \simeq j$

– all ground instances from  $N$  excluding ...

$(x \simeq t \vee D)_\sigma$  where  $x\sigma \equiv f(\vec{i})$  and  $(x \simeq t \vee D)_\sigma \preceq \bigvee_{j=1}^n f(\vec{i}) \simeq j$

- Redundancy notion is *almost* compatible with standard one.  
*Rare* deviations exist *both ways*.



# Termination Aspects (1)

- *Goal*: “Deduction *as* decision procedure”  
Present *one* possible solution.
- Back to standard superposition and ground clause sets:  
*infinite derivations* like  $\underline{a} \simeq f^i(b)$ ,  $f(\underline{a}) \simeq b \vdash f^{i+1}(b) \simeq a$
- Remedy on Horn fragment: *positive unit literal strategy*  
Beyond Horn: add *splitting rule*

- Strategy promising in practice:  
*lift it!*

						1	
4							
	2						
			5	4	7		
		8		3			
		1	9				
3		4		2			
	5	1					
		8	6				



0.3 sec



6	9	3	7	8	4	5	1	2
4	8	7	5	1	2	9	3	6
1	2	5	9	6	3	8	7	4
9	3	2	6	5	1	4	8	7
5	6	8	2	4	7	3	9	1
7	4	1	3	9	8	6	2	5
3	1	9	4	7	5	2	6	8
8	5	6	1	2	9	7	4	3
2	7	4	8	3	6	1	5	9



## Termination Aspects (2)

- Splitting of *non-ground* clauses:
  - in *general* case *variable-disjoint parts* only  
hence additional inferences like equality factoring
  - In case of *finite* domains:  
enforce variable disjointness via *instantiation with digits*

$$\text{Split} \quad \frac{C \vee s \simeq t \vee l \simeq r \vee D}{(C \vee s \simeq t)_\tau \mid (l \simeq r \vee D)_\tau}$$

- the partitioning is designated if ·  $\tau$  minimally numbers such that conclusions share no variables

- Observations:
  - Inferences of Thm. 2 *do not increase* the number of variables.
  - If digits are minimal in ordering,  
then eventually every  $f(i_1, \dots, i_m)$  *reduces to a digit*.
  - With some instantiation, only shallow clauses remain.



## Termination Aspects (3)

*Theorem 4:* The presented calculus configuration  
*decides  $\mathcal{T}$ -satisfiability* of finite clause sets.

*Corollary 5:* Superposition for finite domains  
*decides the Bernays-Schönfinkel class.*

$\forall \dots \forall$



$\exists \dots \exists$

For the latter, *no codomain axioms* needed,  
and *no instance rewriting* in decision procedure.



# Combination with Arbitrary Sorts

- Setting: (sub)sorts encoded via *monadic predicates*.  
*Finite* ones:  $\neg S(x) \vee x \simeq 1 \vee \dots \vee x \simeq n$   $S(1)$   $S(2)$   $\dots$   $S(n)$   
*Declarations*: for example  $\neg S(x) \vee S(f(x))$

- *Inference restrictions*:

$$\text{SpL} \quad \frac{C \vee l \simeq r \quad s[l'] \not\simeq t \vee D}{(C \vee s[r] \simeq t \vee D)\sigma} \quad \text{if}$$

- $\sigma = \text{mgu}(l, l')$  and  $l' \notin \mathcal{V}$
- $\text{ran } \sigma|_S \subseteq \mathcal{V} \cup [1; n]$
- *there exists a minimally numbering  $\tau$  of sort  $S$  such that  $l, l \simeq r, s$  and  $s \not\simeq t$  are (strictly) maximal under  $\sigma\tau$*

**Theorem 6:** Restrictions preserves *refutational completeness*.



# Conclusions and Ongoing Work

---

- Drastic *restriction of inferences* possible for finite domains.  
Superposition an alternative to *a-priori grounding* approaches?
- Compatible with *sort concept* as implemented in SPASS
- Can be fine-tuned into *decision procedure* for  $\mathcal{T}$ -satisfiability
- *Experimental evaluation:*  
implementation within SPASS under way  
preliminary results from ground-level Sudoku encodings
- Look for promising calculus *variations*