Superposition and Model Evolution Combined

Peter Baumgartner
NICTA and
Australian National University

Uwe Waldmann
Max Planck Institute
for Informatics

Motivation

- Both Superposition and Model Evolution are calculi for $\mathrm{FOL}_=$
- **Superposition**
  - Equality, redundancy elimination
  - Decides Guarded Fragment, Monadic class, ...
  - Wins FOF CASC division
- **Model Evolution, more generally "Instance Based Methods"**
  - Conceptually different to resolution/superposition
  - Method of choice for Bernays-Schönfinkel class (EPR)
  - Wins EPR CASC division

Combine Superposition and Instance Based Methods?

ME+Sup = Model Evolution + Superposition

Motivating Example

Ordered arrays

$$
\begin{array}{ll}
(1) & x \le z \vee \neg(x \le y) \vee \neg(y \le z) \\
(2) & x \le y \vee y \le x \\
(3) & x \approx y \vee \neg(x \le y) \vee \neg(y \le x) \\
(4) & \mathrm{select}(\mathrm{store}(a, i, e), i) \approx e \\
(5) & \mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \vee i \approx j \\
(6) & i \le j \vee \neg(\mathrm{select}(\mathrm{a0}, i) \le \mathrm{select}(\mathrm{a0}, j))
\end{array}
$$

- Termination on (1)-(3): ME: **yes** Superposition: **no**
- Termination on (4)-(6): ME: **no** Superposition: **yes**
- Termination on (1)-(6): ME+Sup: **yes**
  - use ME for $\le$-literals
  - use Superposition for $\approx$-literals

Propositional Resolution $\rightarrow$ Superposition

Ordered resolution

$$\frac{A \vee C \qquad \neg A \vee D}{C \vee D}$$

if

(i) A is strictly maximal in $A \vee C$

(ii) $\neg A$ is maximal in $\neg A \vee D$

Superposition - ground level

$$\frac{l \approx r \vee C \qquad s[l]_p \not\approx t \vee D}{s[r]_p \not\approx t \vee C \vee D}$$

if

(iii) $l \succ r$,

(iv) $l \approx r$ is strictly maximal in $l \approx r \vee C$,

(v) $s \succ t$, and

(vi) $s \not\approx t$ is maximal in $s \not\approx t \vee C$.

Propositional Resolution $\to$ Superposition

Ordered resolution

$$\frac{A \vee C \qquad \neg A \vee D}{C \vee D}$$

if

(i) A is strictly maximal in $A \vee C$

(ii) $\neg A$ is maximal in $\neg A \vee D$

Superposition

$$\frac{l \approx r \vee C \qquad s[u]_p \not\approx t \vee D}{(s[r]_p \not\approx t \vee C \vee D)\sigma}$$

if

(i) σ is a mgu of l and u,

(ii) u is not a variable,

(iii) $r\sigma \not\succeq l\sigma$,

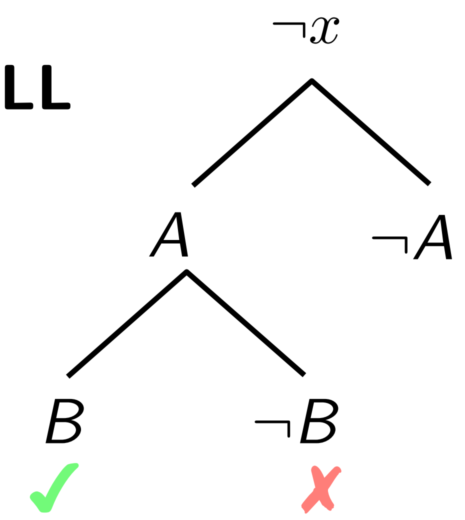
(iv) $(l \approx r)\sigma$ is strictly maximal in $(l \approx r \vee C)\sigma$,

(v) $t\sigma \not\succeq s\sigma$, and

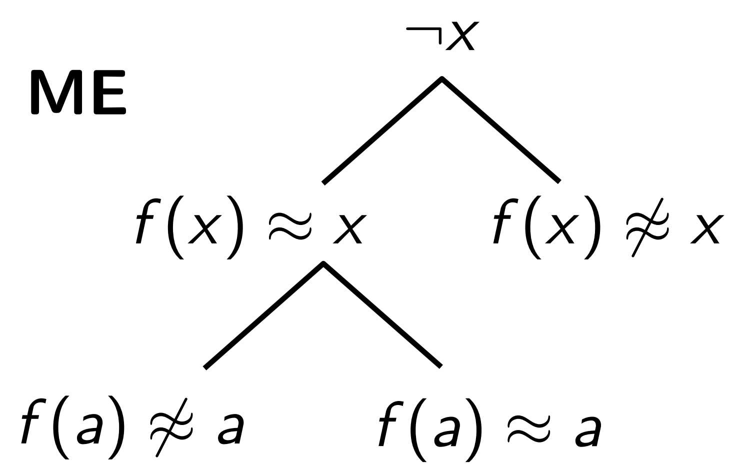
(vi) $(s \not\approx t)\sigma$ is maximal in $(s \not\approx t \vee C)\sigma$.

DPLL $\longrightarrow$ Model Evolution (ME)

DPLL



ME



$\{A\} \overset{?}{\models} \neg A \vee B$ no - Split

$\{A, B\} \overset{?}{\models} \neg A \vee B$ ✓

$\{A, \neg B\} \overset{?}{\models} \neg A \vee B$ ✗ Close

- Branches are called "contexts"
- Context induces interpretation
- **Split** to repair interpretation
- **Close** to abandon interpretation

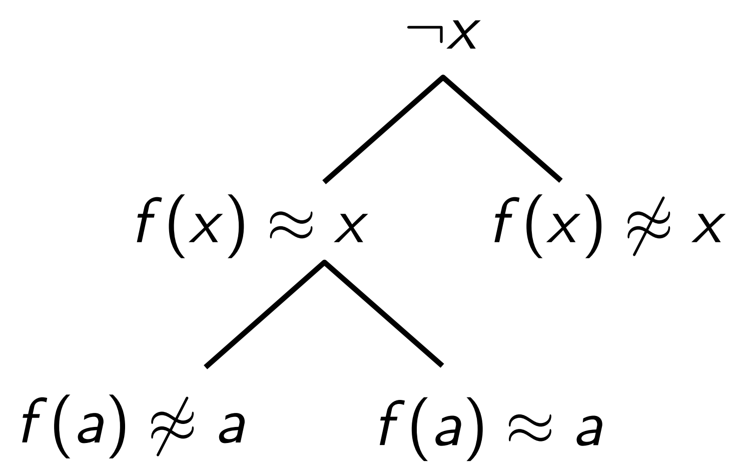
Induced Interpretation via *Productivity*

Productivity

A context literal $K \in \Lambda$ **produces** L iff

(i) L is an instance of K and

(ii) there is no more specific literal in Λ that produces $\overline{L}$



produces

$$\{f(b) \approx b,\ f(f(a)) \approx f(a),\ f(a) \not\approx a,\ f(f(b)) \not\approx b,\ \ldots\}$$

A "syntactic" notion!
Not an E-Interpretation

Productivity is a central concept in the combination of ME and superposition via *constrained clauses*

ME+Sup - Constrained Clauses

Constraint clause $C \cdot \Gamma$

- C is an ordinary clause
- Constraint Γ is a multiset of literals

$$x \leq y \vee y \leq x \;\cdot\; \emptyset$$

$$y \leq x \;\cdot\; \neg(x \leq y)$$

Semantics

- Given context Λ and E-interpretation I
- $\Lambda, I \models C \cdot \Gamma$ iff (if Λ produces Γ then $I \models C$)

$$\begin{array}{c} A \\ | \\ \neg B \end{array}, \{C\} \models C \vee D \cdot A, \neg B$$

C is evaluated "semantically", Γ is evaluated "syntactically"

ME+Sup - Constrained Clauses

Constraint clause $C \cdot \Gamma$

$$x \le y \vee y \le x \;\cdot\; \emptyset$$
$$y \le x \;\cdot\; \neg(x \le y)$$

- C is an ordinary clause
- Constraint Γ is a multiset of literals

Semantics

- Given context Λ and E-interpretation I
- $\Lambda, I \models C \cdot \Gamma$ iff (if Λ produces Γ then $I \models C$)

$$\begin{array}{c} A \\ | \\ \neg B \end{array}, \{\} \not\models C \vee D \cdot A, \neg B$$

C is evaluated "semantically", Γ is evaluated "syntactically"

ME+Sup - Constrained Clauses

$$x \le y \vee y \le x \;\cdot\; \emptyset$$

$$y \le x \;\cdot\; \neg(x \le y)$$

Constraint clause $C \cdot \Gamma$

- C is an ordinary clause
- Constraint Γ is a multiset of literals

Semantics

- Given context Λ and E-interpretation I
- $\Lambda, I \models C \cdot \Gamma$ iff (if Λ produces Γ then $I \models C$)

$$\begin{array}{c} A \\ | \\ \neg B \end{array} , \mathbf{?} \models C \vee D \cdot A, B$$

***C* is evaluated "semantically", *Γ* is evaluated "syntactically"**

ME+Sup - Constrained Clauses

Constraint clause $C \cdot \Gamma$

$$x \leq y \vee y \leq x \;\cdot\; \emptyset$$

$$y \leq x \;\cdot\; \neg(x \leq y)$$

- C is an ordinary clause
- Constraint Γ is a multiset of literals

Semantics

- Given context Λ and E-interpretation I
- $\Lambda, I \models C \cdot \Gamma$ iff (if Λ produces Γ then $I \models C$)

$$\begin{array}{c} f(x) \approx x \\ | \\ f(a) \not\approx a \end{array}, \mathbf{?} \models f(b) \not\approx b \cdot f(a) \approx a$$

C is evaluated "semantically", Γ is evaluated "syntactically"

ME+Sup Calculus - Initialisation

- **Given**: clause set M
- **Initialisation**
  - Context $\neg x$
  - Constrained clause set $\Phi = \{ \ C \cdot \emptyset \mid C \in M \ \}$

    It holds $\Lambda, I \models \Phi$ iff $I \models M$
- User-supplied **control parameters**
  - Term ordering, as usual
  - Labelling on ground atoms:

    split atoms $\cup$ **superposition atoms** $=$ Herbrand base
  - Can also configure pure ME or pure Superposition calculus:

    superposition atoms $= \emptyset$ or split atoms $= \emptyset$

Labelling Example

(1) $x \leq z \vee \neg(x \leq y) \vee \neg(y \leq z)$

(2) $x \leq y \vee y \leq x$

(3) $x \approx y \vee \neg(x \leq y) \vee \neg(y \leq x)$

(4) $\mathrm{select}(\mathrm{store}(a, i, e), i) \approx e$

(5) $\mathrm{select}(\mathrm{store}(a, i, e), j) \approx \mathrm{select}(a, j) \vee i \approx j$

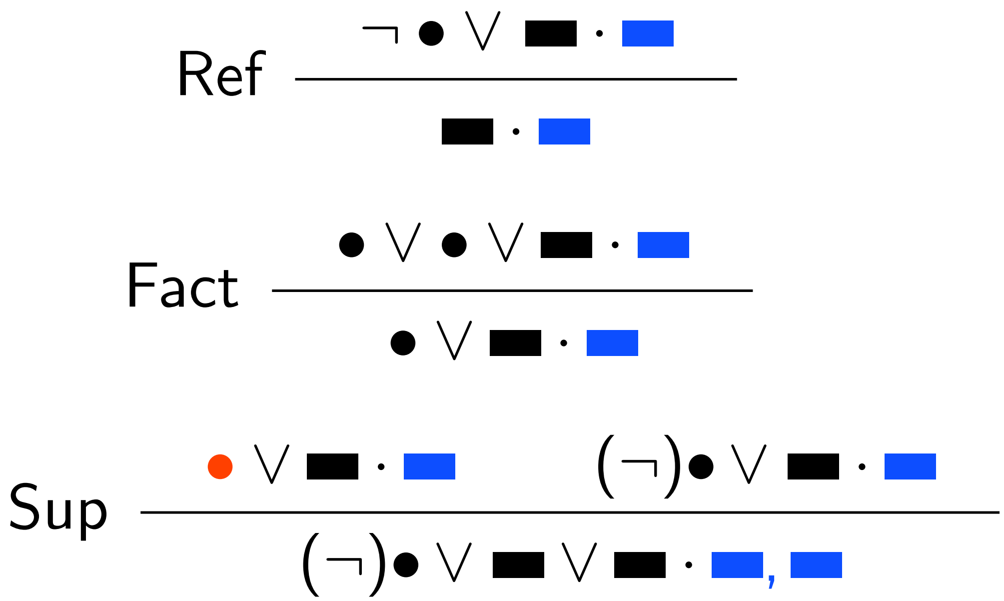
(6) $i \leq j \vee \neg(\mathrm{select}(\mathrm{a0}, i) \leq \mathrm{select}(\mathrm{a0}, j))$

split / **superposition** atom

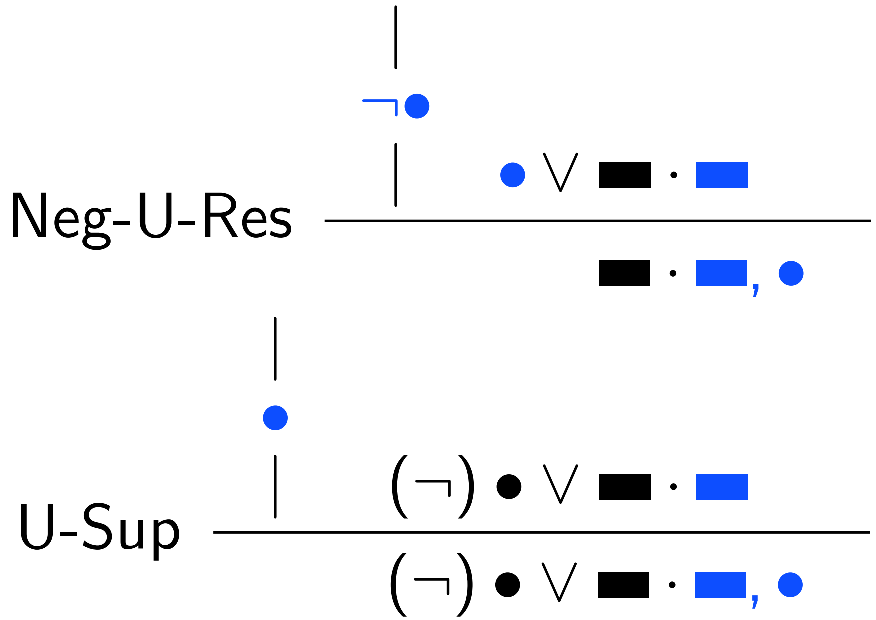
Labelling is used to control inference rule applications

ME+Sup Calculus - Inference Rules

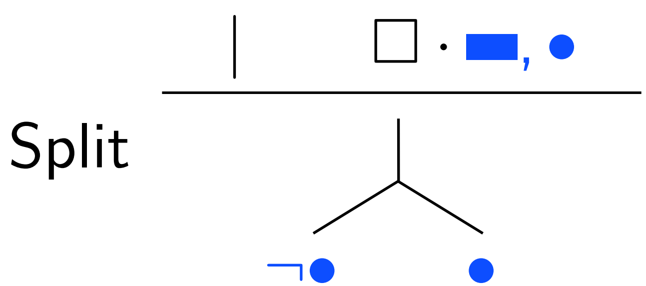
Clause (x Clause) $\mapsto$ Clause

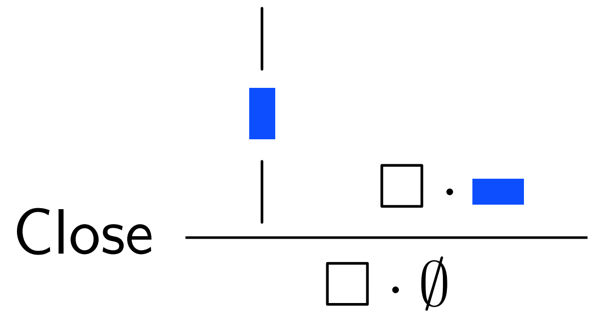


Context x Clause $\mapsto$ Clause



Context x Clause $\mapsto$ Context x Context





split / **superposition** atom

U-Sup: Context x Clause $\mapsto$ Clause

$$\begin{array}{c} \neg x \\ | \\ f(x) \approx x \\ | \\ f(a) \not\approx a \end{array} \qquad f(b) \not\approx c \cdot \emptyset$$

$$\overline{\qquad\qquad b \not\approx c \cdot f(b) \approx b \qquad\qquad}$$

Must add $f(b) \approx b$ to the constraint because $f(b) \approx b$ could be false in the induced interpretation

U-Sup: Context x Clause $\mapsto$ Clause

$$\text{U-Sup}\ \frac{\Lambda, l \approx r \qquad s[u]_p \not\approx t \vee C \cdot \Gamma}{(s[r]_p \not\approx t \vee C \cdot \Gamma, l \approx r)\sigma}$$

where

(i) σ is a mgu of l and u,

(ii) u is not a variable,

(iii) $(l \approx r)\sigma$ is a **split** atom,

At least one ground instance is a split atom

(iv) $r\sigma \not\succeq l\sigma$,

(v) $t\sigma \not\succeq s\sigma$,

(vi) $(s \not\approx t)\sigma$ is maximal in $(s \not\approx t \vee C)\sigma$, and

(vii) $\Lambda \cup \{l \approx r\}$ **produces** $(l \approx r)\sigma$

Sup: Clause x Clause $\mapsto$ Clause

$$\text{Sup}\ \frac{l \approx r \vee C' \cdot \Gamma' \qquad s[u]_p \not\approx t \vee C \cdot \Gamma}{(s[r]_p \not\approx t \vee C \vee C' \cdot \Gamma, \Gamma')\sigma\pi}$$

Standard Superposition is a special case

(i) σ is a mgu of l and u,

(ii) u is not a variable,

(iii) π merges $x_1 \approx t_1 \vee \cdots \vee x_n \approx t_n \subseteq C'\sigma$ with $(l \approx r)\sigma$,

(iv) $\{x_1, \ldots, x_n\} \subseteq \mathcal{V}ar(\Gamma'\sigma)$,

(v) $(l \approx r)\sigma$ is a **superposition** atom,

(vi) $r\sigma\pi \not\succeq l\sigma\pi$,

(vii) $(l \approx r)\sigma\pi$ is strictly maximal in $(l \approx r \vee C')\sigma\pi$,

(viii) $t\sigma \not\succeq s\sigma$, and

(ix) $(s \not\approx t)\sigma$ is maximal in $(s \not\approx t \vee C)\sigma$.

Split: Context x Clause $\mapsto$ Context x Context

$$\frac{f(x) \approx x \qquad \square \cdot f(a) \approx a}{f(x) \approx x}$$

$$f(a) \not\approx a \qquad\qquad f(a) \approx a$$

$$\text{Split} \quad \frac{\Lambda \qquad \square \cdot \Gamma,\ s \approx t}{\Lambda,\ s \not\approx t \qquad \Lambda,\ s \approx t}$$

(Similarly for $\square \cdot \Gamma, s \not\approx t$)

(i) Λ produces every literal in $\Gamma \cup \{s \approx t\}$

(ii) neither $s \approx t \in \Lambda$ nor $s \not\approx t \in \Lambda$

(iii) $s \approx t$ is a **split** atom

Derivation Example

(No equality in this example, empty constraints not shown)

(1a) $x \leq z \vee \neg(x \leq y) \vee \neg \mathrm{leq}(y, z)$

(1b) $\mathrm{leq}(y, z) \vee \neg(y \leq z)$

(2) $x \leq y \vee y \leq x$

leq-atoms: split

$\leq$-atoms: superposition

leq-atoms $>$ $\leq$-atoms

Initial context is $\neg \mathrm{x}$

Resolution on (1a) and (1b) is blocked

(3) $x \leq x$ (Factoring (2))

(4) $\neg(y \leq z) \cdot \neg \mathrm{leq}(y, z)$ (Neg-U-Res of (1b))

(5) $z \leq y \cdot \neg \mathrm{leq}(y, z)$ (Resolution (2)+(4))

(6) $\square \cdot \neg \mathrm{leq}(x, x)$ (Resolution (3)+(4))

Derivation Example

(1a) $x \leq z \lor \neg(x \leq y) \lor \neg \mathrm{leq}(y, z)$

$\vdots$

(6) $\square \cdot \neg \mathrm{leq}(x, x)$

(ctxt-1) $\mathrm{leq}(x, x)$ $\neg \mathrm{leq}(x, x)$ (Split)

$\star$

(7) $x \leq y \lor \neg(x \leq y) \cdot \mathrm{leq}(y, y)$ ((ctxt-1)+(1a))

(Derivation continues, but will terminate eventually)

Inference rule applications controlled by
labelling, orderings, productivity, *redundancy/simplification*

Redundancy and Simplification

- A ground clause $C \cdot \Gamma$ is redundant if it follows from smaller ground clauses and "certain additional conditions" are satisfied
- DPLL-style simplification rules by elements from current context Λ

$$\cancel{f(x) \approx x \cdot g(a) \approx a}$$
if $g(a) \not\approx a \in \Lambda$

$$f(x) \approx x \cdot \cancel{g(a) \approx a}$$
if $g(a) \approx a \in \Lambda$

- Simplification by clauses from current clause set Φ

$$f(x) \approx h(x) \cdot g(x) \approx x$$
$$\Big\downarrow \; f(x) \approx x \,.\, g(x) \approx x \;\in \Phi$$
$$x \approx h(x) \cdot g(x) \approx x,\; g(x) \approx x$$

Generalizes redundancy/simplification of ME and Superposition

Soundness and Completeness

- ME+Sup is sound
- ME+Sup (with simplification) is refutationally complete
  - Every fair derivation from an unsatisfiable clause set ends in a derivation tree where every leaf is closed
  - Fairness: every inference from persistent non-redundant premises becomes redundant eventually
  - But input clauses must not contain constraints
    $P(x) \cdot \emptyset$ is OK $\qquad$ $\square \cdot \neg P(x)$ is not OK
  - Proof by adaptation of Bachmair/Ganzinger model construction technique

Model Construction

Given: Λ, Φ is saturated and $\Box \cdot \emptyset \notin \Phi$. Construct rewrite system R

Need a total ordering:

- $C_1 \cdot \Gamma_1$ and $C_2 \cdot \Gamma_2$ compared lexicographically
- Context equation $s \approx t$ is taken as $s \approx t \cdot \bot$ where $\bot \prec \emptyset$

Inspect $\Pi_\Lambda \cup \text{ground}(\Phi)$ in increasing order

(i) If $s \approx t \in \Pi_\Lambda$, $s \succ t$, $s \approx t$ and s and t are irreducible wrt. $R_{s \to t}$ then add $s \to t$ to R

$\{P(x), \neg P(b), a \approx b\}$ assigns false to $P(a)$

(ii) If $s \approx t \vee C \cdot \Gamma$ in ground(Φ), Λ produces Γ and "other conditions" apply then add $s \to t$ to R

Can show that inference rules reduce smallest relevant counterexample

Conclusions

- ME+Sup
  - Properly generalizes Superposition with redundancy criteria
  - Generalize essentials of Model Evolution with Equality (universal variables and some optional inference rules missing)
  - Symmetric integration, configuration of mixed calculi
- Technical complications required some new concepts
- Future work
  - New decision procedures?
  - Generalization of full Model Evolution with Equality
  - "Basic" variants of inference rules