Superposition for Full Higher-Order Logic

Alexander Bentkamp
Jasmin Blanchette
Sophie Tourret
Petar Vukmirović

Genealogy of the Calculus

Genealogy of the Calculus

Standard superposition

Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

+ applied variables

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

+ λ-expressions

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

+ applied variables

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

+ λ-expressions

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

+ applied variables

Standard superposition
Bachmair & Ganzinger (1994)

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
```

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
      = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer
```

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer
```

Translation to
first-order logic

First-Order ATPs
CVC4, E, Vampire, Z3

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer
```

Translation to first-order logic

First-Order ATPs
CVC4, E, Vampire, Z3

Proof reconstruction

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer
```

```
Sledgehammering...
"e": Timed out
"cvc4": Timed out
"z3": Timed out
"vampire": Timed out
```

Translation to first-order logic

First-Order ATPs
CVC4, E, Vampire, Z3

Proof reconstruction

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
```

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer [zipperposition]
```

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer [zipperposition]
```

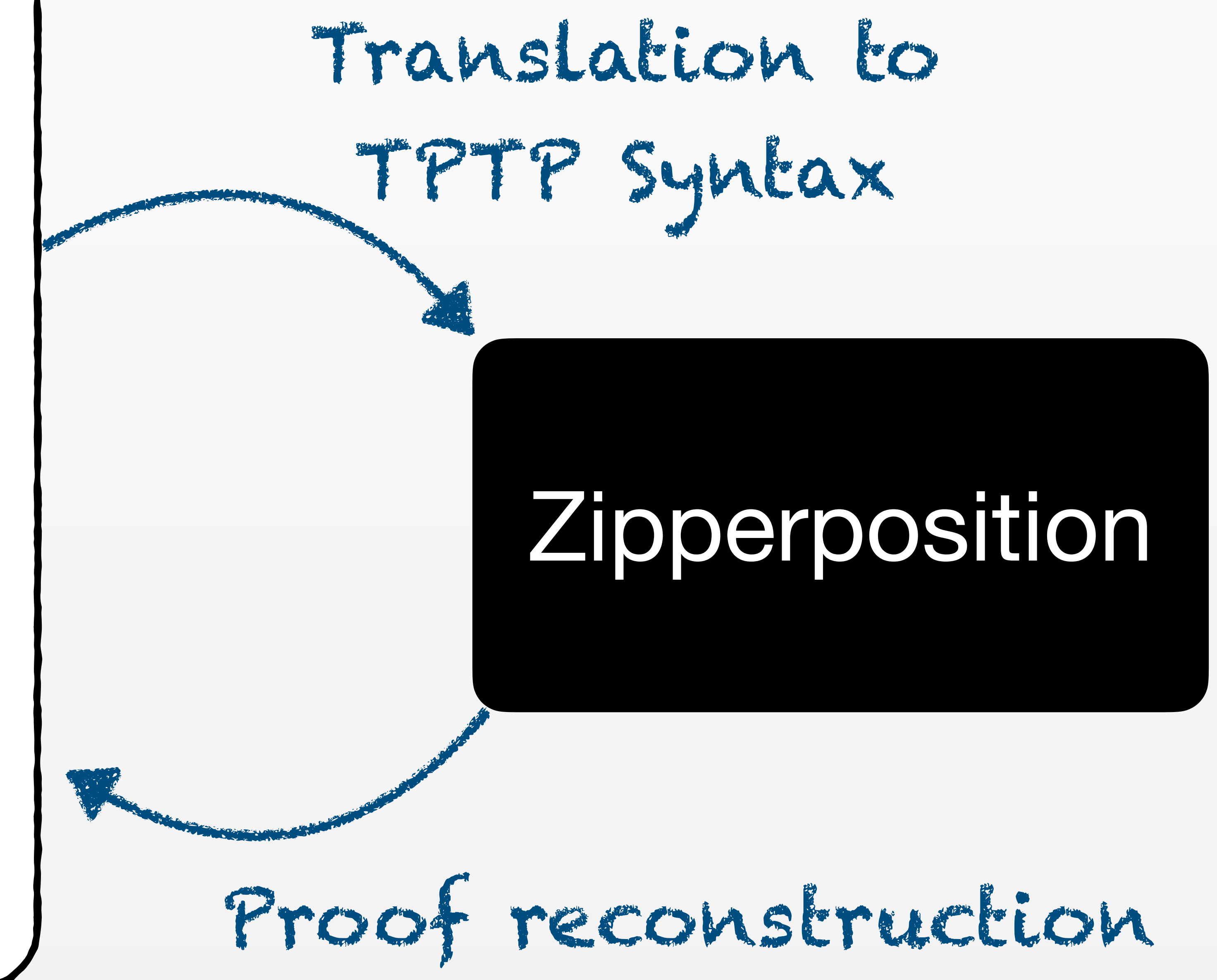
Translation to TPTP Syntax

Zipperposition

Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer [zipperposition]
```

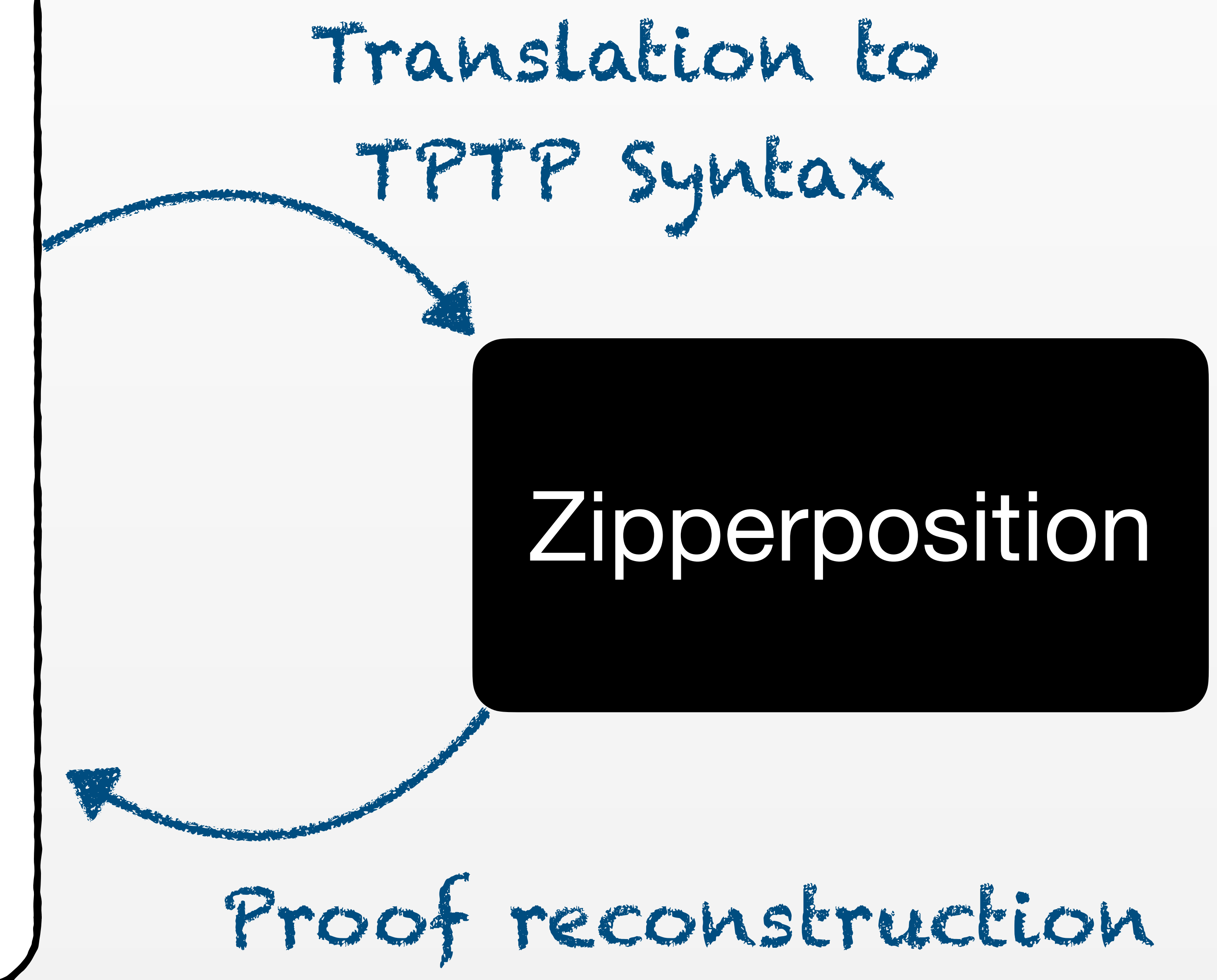


Motivation

Isabelle/HOL

```
lemma "(∑ i ∈ A. i ^ 2 + 2 * i + 1)
        = (∑ i ∈ A. i ^ 2) + (∑ i ∈ A. 2 * i) + (∑ i ∈ A. 1)"
  by sledgehammer [zipperposition]


Sledgehammering...
Proof found...
"zipperposition": Try this: by (simp add: sum.distrib) (31 ms)
```



Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

+ λ-expressions

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

+ applied variables

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans
+ delayed clausification

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

+ λ-expressions

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

+ applied variables

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans
+ delayed clausification

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

+ λ-expressions

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

Superposition with Equivalences and Delayed Clausification
Ganzinger & Stuber (2005)

+ applied variables

+ delayed clausification

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans
+ delayed clausification

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

Superposition with Booleans
Nummelin, Bentkamp, Tourret, Vukmirović (2021)

+ λ-expressions

+ Booleans

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

Superposition with Equivalences and Delayed Clausification
Ganzinger & Stuber (2005)

+ applied variables

+ delayed clausification

Standard superposition
Bachmair & Ganzinger (1994)

Genealogy of the Calculus

Superposition for Full Higher-Order Logic
Bentkamp, Blanchette, Tourret, Vukmirović (2021)

+ Booleans
+ delayed clausification

+ applied variables
+ lambda expressions

Boolean-free λ-superposition
Bentkamp, Blanchette, Tourret, Vukmirović, Waldmann (2019)

Superposition with Booleans
Nummelin, Bentkamp, Tourret, Vukmirović (2021)

+ λ-expressions

+ Booleans

Boolean-free λ-free superposition
Bentkamp, Blanchette, Cruanes, Waldmann (2018)

Superposition with Equivalences and Delayed Clausification
Ganzinger & Stuber (2005)

+ applied variables

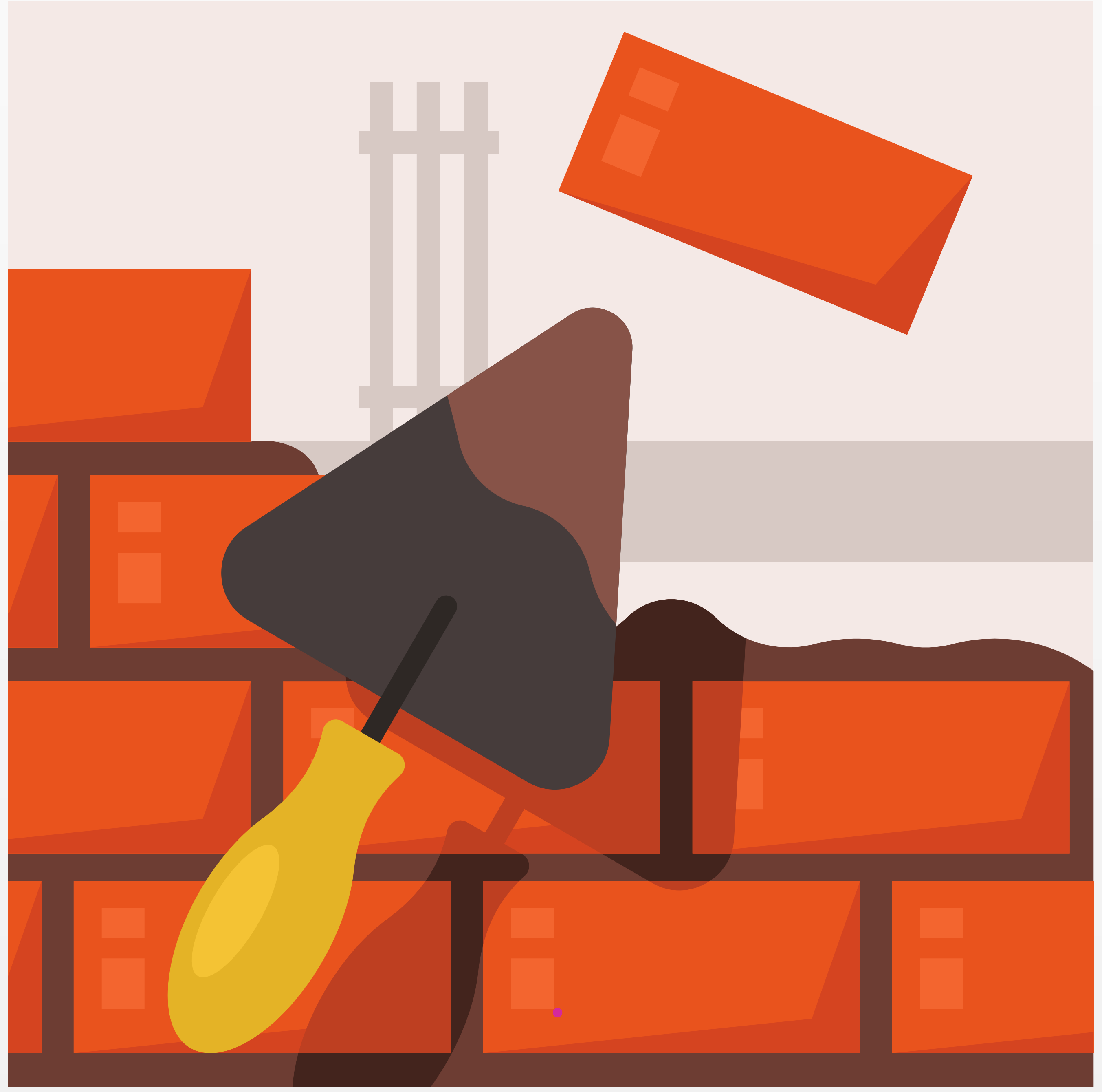
+ delayed clausification

Standard superposition
Bachmair & Ganzinger (1994)

Guiding Principles

Guiding Principles

Build on what works for first-order logic.



The superposition calculus is extremely successful, especially for Sledgehammer.

Guiding Principles

Be graceful.

The calculus should gracefully generalize first-order superposition.

Guiding Principles

Be complete.

Completeness proof and implementation give insight to each other.

The Calculus Rules

The Calculus Rules

Boolean-free λ-superposition

SUP

ERES

EFACT

FLUIDSUP

ARGCONG

EXT

The Calculus Rules

Boolean-free λ-superposition

SUP

ERES

EFACT

FLUIDSUP

ARGCONG

EXT

Superposition with Booleans

BOOLHOIST

EQHOIST

NEQHOIST

FORALLHOIST

EXISTSHOIST

FALSEELIM

BOOLRW

FORALLRW

EXISTSRW

The Calculus Rules

Boolean-free λ-superposition

SUP

ERES

EFACT

FLUIDSUP

ARGCONG

EXT

Superposition with Booleans

BOOLHOIST

EQHOIST

NEQHOIST

FORALLHOIST

EXISTSHOIST

FALSEELIM

BOOLRW

FORALLRW

EXISTSRW

New

CHOICE

FLUIDBOOLHOIST

FLUIDLOOBHOIST

Three Challenges

Primitive substitution

Primitive substitution

$a \not\approx b$

$\neg z\, a \lor z\, b$

Primitive substitution

$\boxed{\mathsf{a} \not\approx \mathsf{b}}$

$\neg z\ \mathsf{a} \lor z\ \mathsf{b}$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$

$\boxed{\neg z\ \mathsf{a}} \lor z\ \mathsf{b}$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$

$\neg z\ \mathsf{a} \lor \boxed{z\ \mathsf{b}}$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$

$\neg z\ \mathsf{a} \lor z\ \mathsf{b}$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$

$\neg z\ \mathsf{a} \lor z\ \mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\, v \approx \mathsf{a}$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$ $\neg z\,\mathsf{a} \lor z\,\mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v.\, v \approx \mathsf{a}$ ➠ $\neg(\mathsf{a} \approx \mathsf{a}) \lor (\mathsf{b} \approx \mathsf{a})$

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\neg z\,\mathsf{a} \lor z\,\mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\,v \approx \mathsf{a}$ $\quad\Rightarrow\quad$ $\neg(\mathsf{a} \approx \mathsf{a}) \lor (\mathsf{b} \approx \mathsf{a})$

Cannot be found through unification

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$

$\neg z\ \mathsf{a} \lor z\ \mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\,v \approx \mathsf{a}$ ➠ $\neg(\mathsf{a} \approx \mathsf{a}) \lor (\mathsf{b} \approx \mathsf{a})$

Cannot be found through unification

Primitive substitution: $z \mapsto \lambda v\,.\,y\ v \approx y'\ v$

Blindly enumerate logical symbols

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\neg z\,\mathsf{a} \vee z\,\mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\, v \approx \mathsf{a}$ ➠ $\neg(\mathsf{a} \approx \mathsf{a}) \vee (\mathsf{b} \approx \mathsf{a})$

Cannot be found through unification

Primitive substitution: $z \mapsto \lambda v\,.\, y\,v \approx y'\,v$ ➠ $\neg(x\,\mathsf{a} \approx y\,\mathsf{a}) \vee (x\,\mathsf{b} \approx y\,\mathsf{b})$

Blindly enumerate logical symbols

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\neg z\,\mathsf{a} \vee z\,\mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\,v \approx \mathsf{a}$ $\quad \Rightarrow \quad$ $\neg(\mathsf{a} \approx \mathsf{a}) \vee (\mathsf{b} \approx \mathsf{a})$

Cannot be found through unification

Primitive substitution: $z \mapsto \lambda v\,.\,y\,v \approx y'\,v$ $\quad \Rightarrow \quad$ $\neg(x\,\mathsf{a} \approx y\,\mathsf{a}) \vee (x\,\mathsf{b} \approx y\,\mathsf{b})$

Blindly enumerate logical symbols

Problem: Primitive substitutions are redundant.

Primitive substitution

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\neg z\, \mathsf{a} \lor z\, \mathsf{b}$

Unsatisfiable because: $z \mapsto \lambda v\,.\, v \approx \mathsf{a}$ $\quad\Rightarrow\quad$ $\neg(\mathsf{a} \approx \mathsf{a}) \lor (\mathsf{b} \approx \mathsf{a})$

Cannot be found through unification

Primitive substitution: $z \mapsto \lambda v\,.\, y\, v \approx y'\, v$ $\quad\Rightarrow\quad$ $\neg(x\, \mathsf{a} \approx y\, \mathsf{a}) \lor (x\, \mathsf{b} \approx y\, \mathsf{b})$

Blindly enumerate logical symbols

Problem: Primitive substitutions are redundant.

Solution: Immediately clausify.

FluidBoolHoist

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\mathsf{h}\ (y\ \mathsf{b}) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (y\ \mathsf{a}) \not\approx \mathsf{h}\ (\mathsf{g}\ \top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\,(x \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$

$\mathsf{h}\,(y\,\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \lor \mathsf{h}\,(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\,\top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\ (x \approx \mathsf{a})$

$\mathsf{g}\ (\mathsf{b} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$

$\mathsf{h}\ (y\ \mathsf{b}) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (y\ \mathsf{a}) \not\approx \mathsf{h}\ (\mathsf{g}\ \top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\,(x \approx \mathsf{a})$

$\mathsf{g}\,(\mathsf{b} \approx \mathsf{a})$ $\qquad$ $\mathsf{g}\,(\mathsf{a} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad$ $\mathsf{h}\,(y\,\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \vee \mathsf{h}\,(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\,\top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\ (x \approx \mathsf{a})$

$\mathsf{g}\ (\mathsf{b} \approx \mathsf{a})$ $\qquad\qquad$ $\mathsf{g}\ (\mathsf{a} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad\qquad$ $\mathsf{h}\ (y\ \mathsf{b}) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (y\ \mathsf{a}) \not\approx \mathsf{h}\ (\mathsf{g}\ \top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\ (x \approx \mathsf{a})$

$\mathsf{g}\ (\mathsf{b} \approx \mathsf{a})$ $\qquad$ $\mathsf{g}\ (\mathsf{a} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$ $\qquad$ $\mathsf{h}\ (y\ \mathsf{b}) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (y\ \mathsf{a}) \not\approx \mathsf{h}\ (\mathsf{g}\ \top)$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\ (x \approx \mathsf{a})$

$\mathsf{g}\ (\mathsf{b} \approx \mathsf{a})$ $\qquad\qquad$ $\mathsf{g}\ (\mathsf{a} \approx \mathsf{a})$

$$\mathsf{a} \not\approx \mathsf{b} \qquad \frac{\mathsf{h}\ (y\ \mathsf{b}) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (y\ \mathsf{a}) \not\approx \mathsf{h}\ (\mathsf{g}\ \top)}{\mathsf{h}\ (z\ \mathsf{b}\ \bot) \not\approx \mathsf{h}\ (\mathsf{g}\ \bot) \lor \mathsf{h}\ (z\ \mathsf{a}\ (x\ \mathsf{a})) \not\approx \mathsf{h}\ (\mathsf{g}\ \top) \lor x\ \mathsf{b} \approx \top}\ \textsc{FluidBoolHoist}$$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\,(x \approx \mathsf{a})$

$\mathsf{g}\,(\mathsf{b} \approx \mathsf{a})$ $\qquad$ $\mathsf{g}\,(\mathsf{a} \approx \mathsf{a})$

$$\mathsf{a} \not\approx \mathsf{b} \qquad \frac{\mathsf{h}\,(y\,\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \vee \mathsf{h}\,(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\,\top)}{\mathsf{h}\,(z\,\mathsf{b}\,\bot) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \vee \mathsf{h}\,(z\,\mathsf{a}\,(x\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \vee x\,\mathsf{b} \approx \top}\,\textsc{FluidBoolHoist}$$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\,(x \approx \mathsf{a})$

$\mathsf{g}\,(\mathsf{b} \approx \mathsf{a})$ $\mathsf{g}\,(\mathsf{a} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$

$$\frac{\mathsf{h}\,(y\;\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\;\bot) \lor \mathsf{h}\,(y\;\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\;\top)}{\mathsf{h}\,(z\;\mathsf{b}\;\bot) \not\approx \mathsf{h}\,(\mathsf{g}\;\bot) \lor \mathsf{h}\,(z\;\mathsf{a}\,(x\;\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\;\top) \lor x\;\mathsf{b} \approx \top}\;\textsc{FluidBoolHoist}$$

FluidBoolHoist

Unsatisfiable because: $y \mapsto \lambda x\,.\,\mathsf{g}\,(x \approx \mathsf{a})$

$\mathsf{g}\,(\mathsf{b} \approx \mathsf{a})$ $\qquad$ $\mathsf{g}\,(\mathsf{a} \approx \mathsf{a})$

$\mathsf{a} \not\approx \mathsf{b}$

$$\frac{\mathsf{h}\,(y\,\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \lor \mathsf{h}\,(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\,\top)}{\mathsf{h}\,(z\,\mathsf{b}\,\bot) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \lor \mathsf{h}\,(z\,\mathsf{a}\,(x\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor x\,\mathsf{b} \approx \top}\ \textsc{FluidBoolHoist}$$

FluidBoolHoist

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathsf{a} \not\approx \mathsf{b} \qquad \dfrac{\dfrac{\dfrac{\mathsf{h}\,(y\,\mathsf{b}) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \lor \mathsf{h}\,(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}\,\top)}{\mathsf{h}\,(z\,\mathsf{b}\,\bot) \not\approx \mathsf{h}\,(\mathsf{g}\,\bot) \lor \mathsf{h}\,(z\,\mathsf{a}\,(x\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor x\,\mathsf{b} \approx \top}\,\textsc{FluidBoolHoist}}{\mathsf{h}\,(\mathsf{g}\,(x'\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor x'\,\mathsf{b} \approx \top}\,\textsc{ERes}}{\mathsf{h}\,(\mathsf{g}\,(x''\,\mathsf{a} \approx x'''\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor \bot \approx \top \lor x''\,\mathsf{b} \approx x'''\,\mathsf{b}}\,\textsc{EqHoist}}{\mathsf{h}\,(\mathsf{g}\,(\mathsf{a} \approx x'''\,\mathsf{a})) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor \bot \approx \top \lor \mathsf{a} \not\approx x'''\,\mathsf{b}}\,\textsc{Sup}}{\mathsf{h}\,(\mathsf{g}\,\top) \not\approx \mathsf{h}\,(\mathsf{g}\,\top) \lor \bot \approx \top \lor \mathsf{a} \not\approx \mathsf{a}}\,\textsc{BoolRw}}{\bot \approx \top \lor \mathsf{a} \not\approx \mathsf{a}}\,\textsc{ERes}}{\bot \approx \top}\,\textsc{ERes}}{\bot}\,\textsc{FalseElim}$$

Delayed Clausification of Quantifiers

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x\,.\,t \succ t[x \mapsto u]$$

for all ground terms u

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x\,.\,t \succ t[x \mapsto u]$$

for all ground terms u

Problem: $\forall x\,.\,x\,\mathsf{a} \succ (x\,\mathsf{a})[x \mapsto \lambda z\,.\,\forall x\,.\,x\,z] = \forall x\,.\,x\,\mathsf{a}$

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x \,.\, t \succ t[x \mapsto u]$$

for all ground terms u

where the only Boolean subterms of u are $\top$ and $\bot$

Problem: $\forall x \,.\, x\,\mathsf{a} \succ (x\,\mathsf{a})[x \mapsto \lambda z \,.\, \forall x \,.\, x\, z] = \forall x \,.\, x\,\mathsf{a}$

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x\,.\,t \succ t[x \mapsto u]$$

for all ground terms u

where the only Boolean subterms of u are $\top$ and $\bot$

Problem: $\forall x\,.\,x\,\mathsf{a} \succ (x\,\mathsf{a})[x \mapsto \underbrace{\lambda z\,.\,\forall x\,.\,x\,z}_{\text{This is neither } \top \text{ nor } \bot}] = \forall x\,.\,x\,\mathsf{a}$

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x\,.\,t \succ t[x \mapsto u]$$

for all ground terms u

where the only Boolean subterms of u are $\top$ and $\bot$

Problem: $\forall x\,.\,x\,\mathsf{a} \succ (x\,\mathsf{a})[x \mapsto \lambda z\,.\,\underbrace{\forall x\,.\,x\,z}_{\text{This is neither } \top \text{ nor } \bot}] = \forall x\,.\,x\,\mathsf{a}$

Solution: Eliminate quantifiers binding variables that occur applied.

Delayed Clausification of Quantifiers

Desired property in the completeness proof:

$$\forall x\,.\,t \succ t[x \mapsto u]$$

for all ground terms u

where the only Boolean subterms of u are $\top$ and $\bot$

Problem: $\forall x\,.\,x\,\mathsf{a} \succ (x\,\mathsf{a})[x \mapsto \lambda z\,.\,\underbrace{\forall x\,.\,x\,z}_{\text{This is neither } \top \text{ nor } \bot}] = \forall x\,.\,x\,\mathsf{a}$

Solution: Eliminate quantifiers binding variables that occur applied.

$$\forall x\,.\,t \;\Rightarrow\; \lambda x\,.\,t \approx \lambda x\,.\,\top$$

Modular Completeness Proof

Higher-order logic

Ground higher-order logic

Ground first-order logic

Modular Completeness Proof

saturated set N,
$\bot \notin N$

Higher-order logic

Ground higher-order logic

Ground first-order logic

Modular Completeness Proof

saturated set N,
$\perp \notin$ N

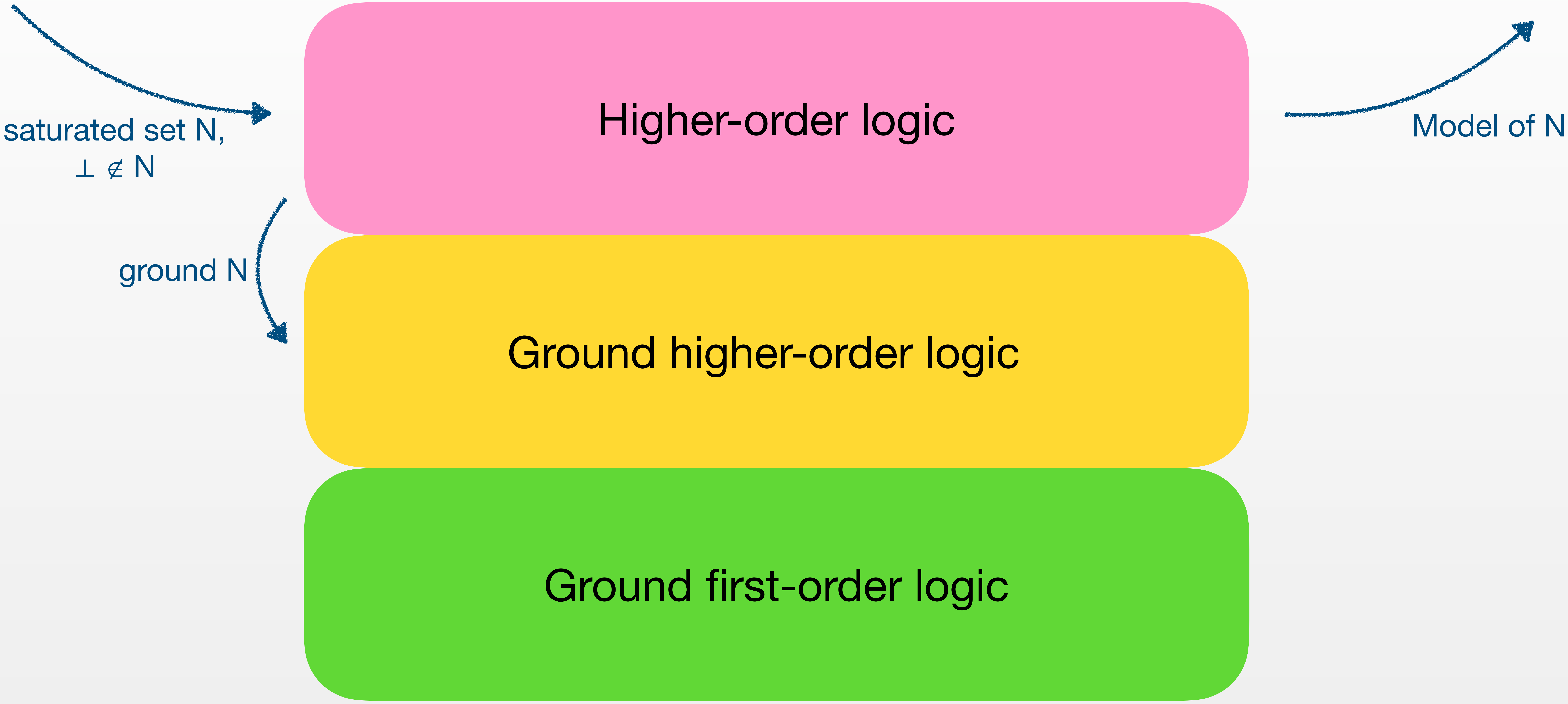
Higher-order logic

Ground higher-order logic

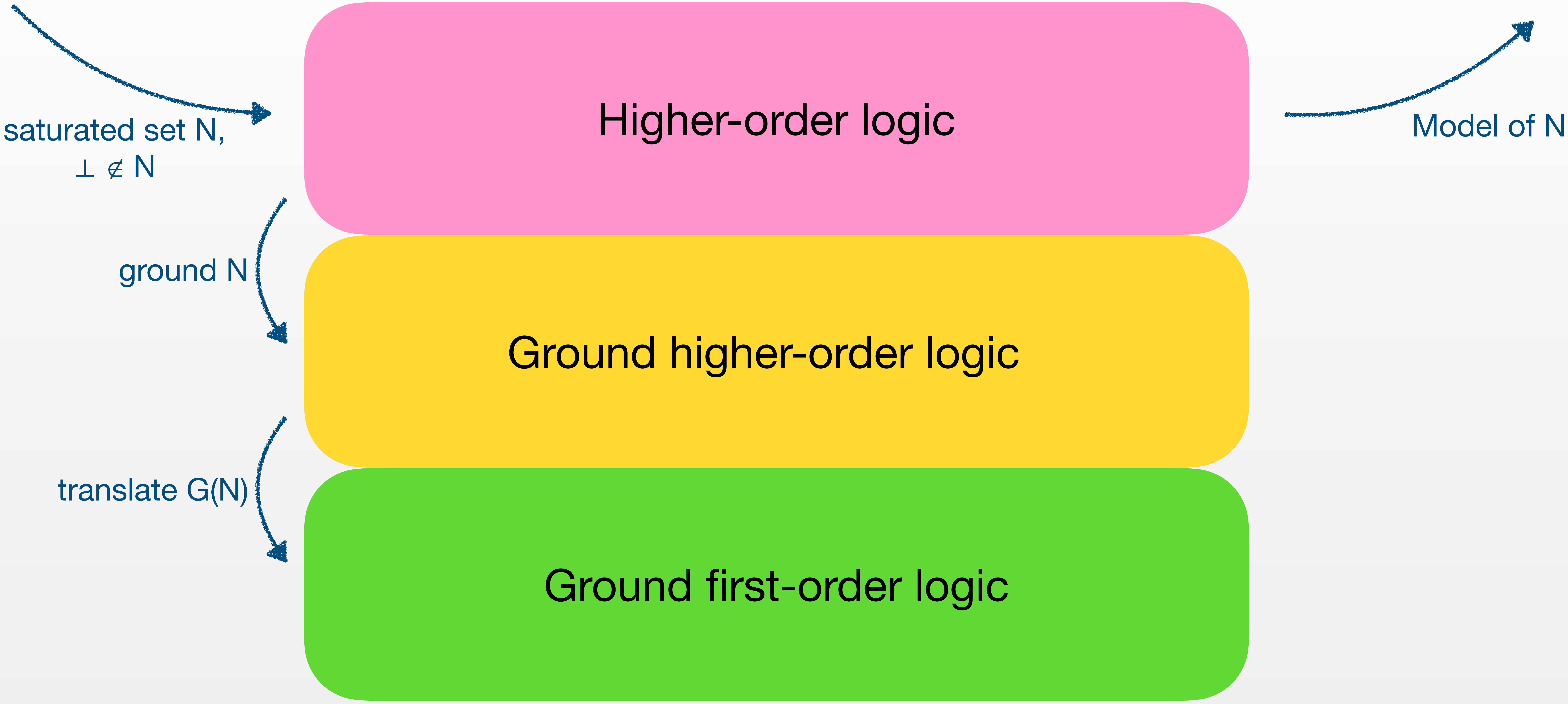
Ground first-order logic

Model of N

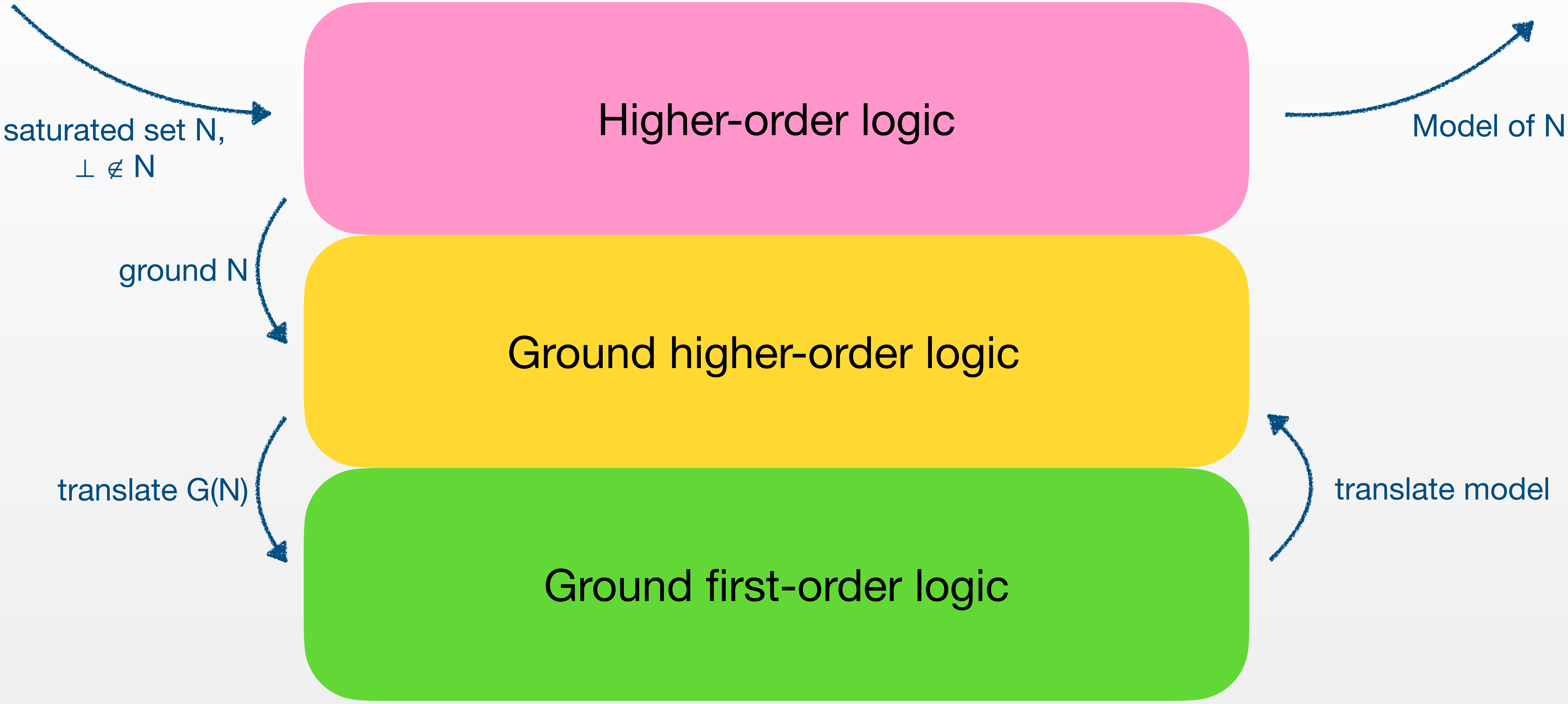
Modular Completeness Proof



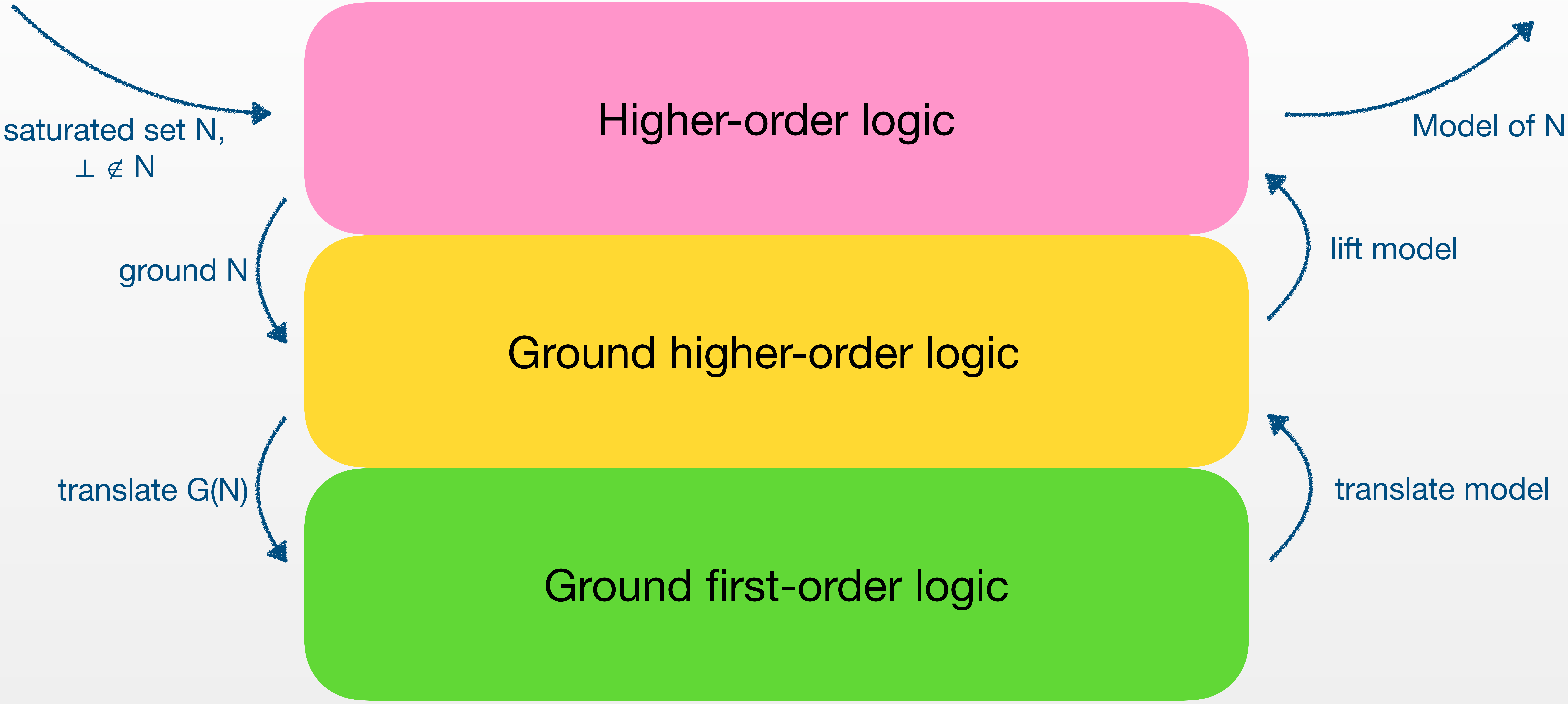
Modular Completeness Proof



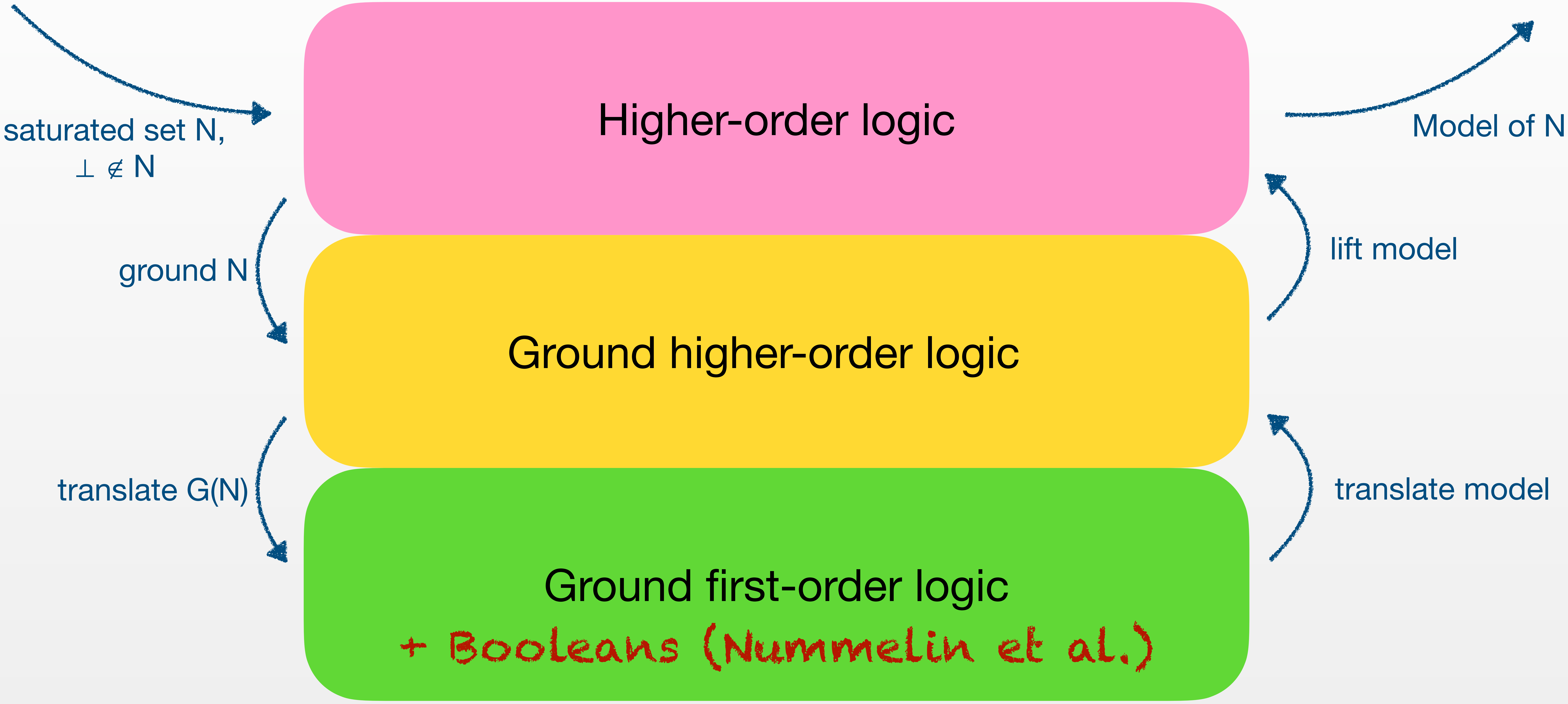
Modular Completeness Proof



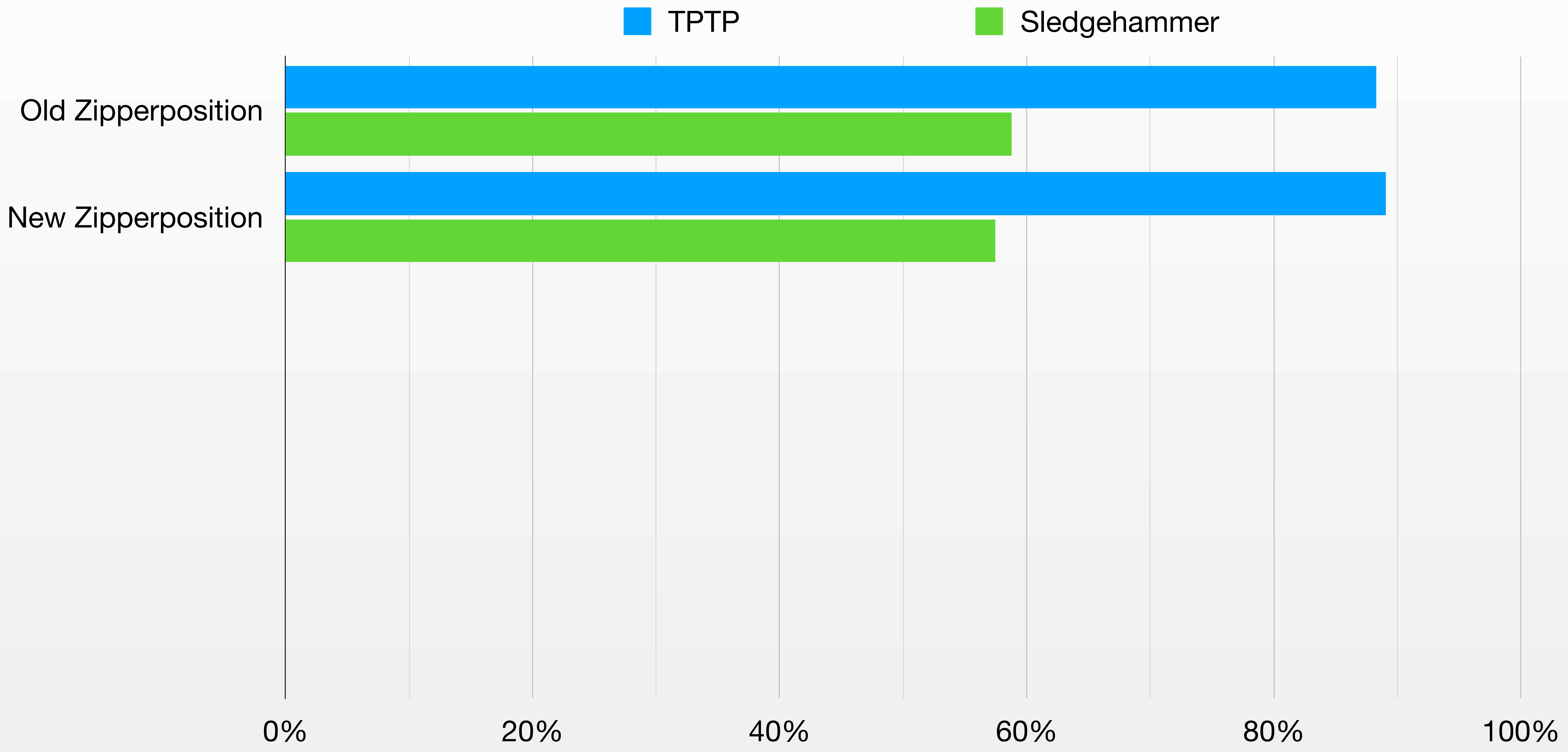
Modular Completeness Proof



Modular Completeness Proof

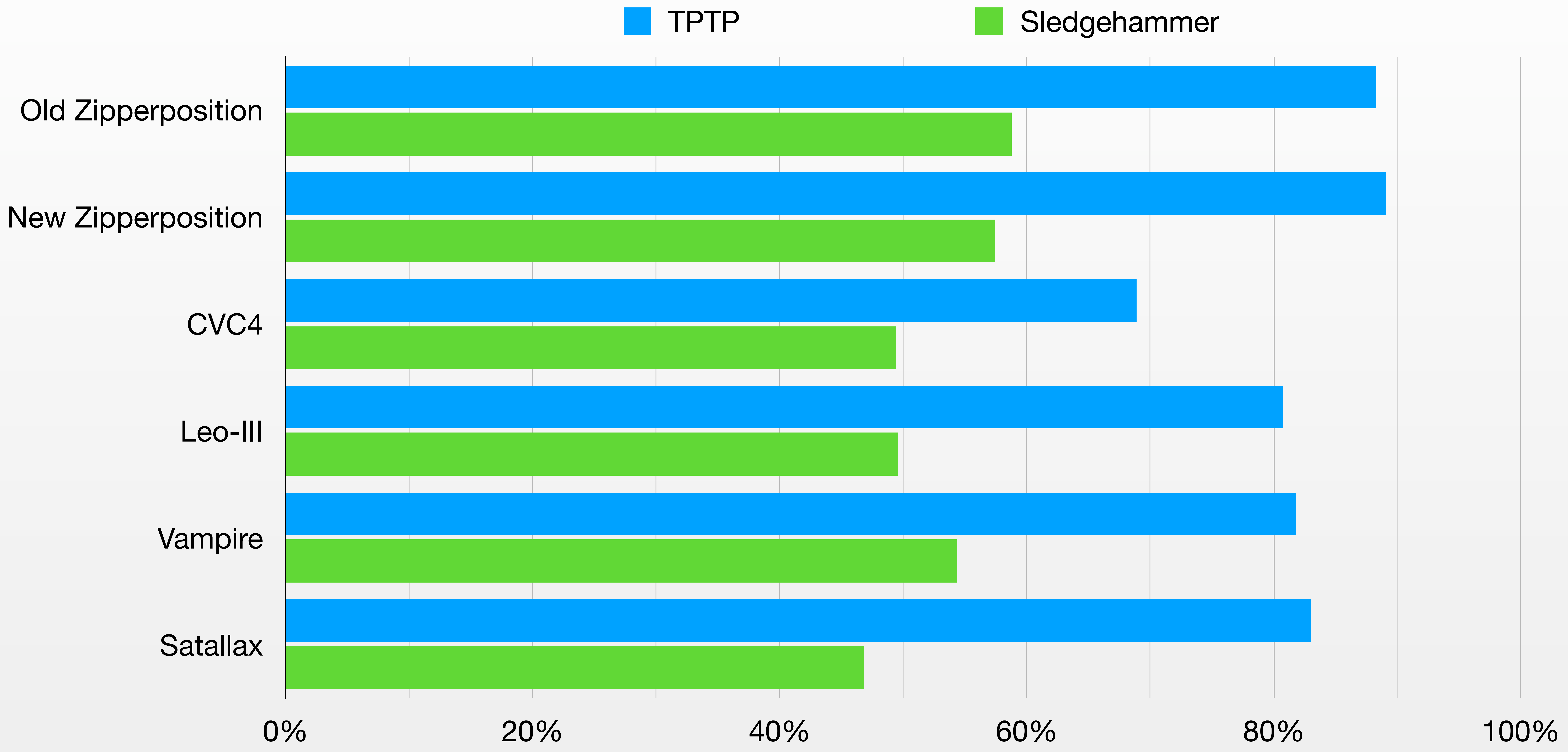


Evaluation



TPTP
Sledgehammer
Old Zipperposition
New Zipperposition
0%
20%
40%
60%
80%
100%

Evaluation



TPTP
Sledgehammer
Old Zipperposition
New Zipperposition
CVC4
Leo-III
Vampire
Satallax
0%
20%
40%
60%
80%
100%

Conclusion

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification

- **Competitive implementation in Zipperposition**

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification

- **Competitive implementation in Zipperposition**

Future work:

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification
- **Competitive implementation in Zipperposition**

Future work:

- **Extensionality**: Axiom is rather inefficient.

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification
- **Competitive implementation in Zipperposition**

Future work:

- **Extensionality**: Axiom is rather inefficient.
- **Preunification**: Currently we compute explicit substitutions, but we might want to store partial unification results in constraints.

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification

- **Competitive implementation in Zipperposition**

Future work:

- **Extensionality**: Axiom is rather inefficient.

- **Preunification**: Currently we compute explicit substitutions, but we might want to store partial unification results in constraints.

- **Implementation in E:** Implementation in a more efficient prover will eradicate the need for a backend.

Conclusion

- **Calculus for full higher-order logic**,
  including a Boolean type and delayed clausification
- **Competitive implementation in Zipperposition**

Future work:

- **Extensionality**: Axiom is rather inefficient.
- **Preunification**: Currently we compute explicit substitutions, but we might want to store partial unification results in constraints.
- **Implementation in E:** Implementation in a more efficient prover will eradicate the need for a backend.
- **Dependent types**: Superposition for dependent types could make hammers for dependently typed systems more efficient.