A Complete Superposition Calculus for Higher-Order Logic

Ahmed Bhayat and Giles Reger

University of Manchester, Manchester, UK

Abstract. Bláh-de-bláhs R_∞ R_{ECA} $\mathcal{G}_\Sigma(ECA)$

1 Preliminaries

Boring, but important stuff. Please read carefully!

- We use $\mathcal{C}$ to stand for an arbitrary member of the set $\{\mathbf{B}, \mathbf{C}, \mathbf{S}\}$
- We use $\tilde{\mathcal{C}}$ to stand for an arbitrary member of the set $\{\mathbf{I}, \mathbf{K}, \mathbf{B}, \mathbf{C}, \mathbf{S}\}$
- We use $x, y, z \ldots$ for variables.
- We use $\mathsf{a}, \mathsf{b}, \mathsf{c} \ldots$ for function symbols of atomic sorts and $\mathsf{f}, \mathsf{g}, \mathsf{h} \ldots$ for function symbols of function sort.
- By *fully-applied combinator* we refer to a $\mathbf{B}$, $\mathbf{C}$ or $\mathbf{S}$ with three or more arguments, a $\mathbf{K}$ with two or more arguments or an $\mathbf{I}$ with one or more argument.
- The symbols ζ and ξ range over head symbols that are not fully-applied combinators
- The symbol δ is used to represent an arbitrary head
- If t is a term then $(t){\downarrow}^w$ is used to denote the term formed from t by weak-reducing its left-most weak-redex.
- We define *first-order* subterms inductively as follows. For any term t, t is a first-order subterm of itself. If $t = \zeta \, \overline{t_n}$, then the first-order subterms of each t_i are also first-order subterms of t. the notation $s\langle u \rangle$ is to be read as u is a first-order subterm of s.
- We define *stable* positions as follows. Let $\mathsf{FNP}(t, p)$ be a partial function that takes a term t, a position p and returns the longest prefix p' of p such that $head(t|_{p'})$ is not a partially applied combinator if such a position exists. For a position $p \in pos(t)$, p is a stable position in t if $\mathsf{FNP}(t, p)$ is not defined or $head(t|_{\mathsf{FNP}(t,p)})$ is first-order. A *stable subterm* is a subterm occurring at a stable position. For example, the subterm a is not stable in $\mathsf{f}\,(\mathbf{S}\,\mathsf{a}\,\mathsf{b}\,\mathsf{c})$, $\mathbf{S}\,(\mathbf{S}\,\mathsf{a})\,\mathsf{b}\,\mathsf{c}$ and $\mathsf{a}\,\mathsf{c}$, but is in $\mathsf{g}\,\mathsf{a}\,\mathsf{b}$ and $\mathsf{f}\,(\mathbf{S}\,\mathsf{a})\,\mathsf{b}$. A subterm that is not stable is known as an *unstable* subterm. **Note that for ground terms, every first-order subterm is also a stable subterm, but not vice versa.**

This definition can be sharpened

- $(\!|\ \ |\!)$ is the following function from λ-terms to combinatory terms:

$$
\begin{aligned}
(\!|\lambda x.x|\!) &= \mathbf{I} \\
(\!|\lambda x.t|\!) &= \mathbf{K}(\!|t|\!), \quad x \text{ doesn't occur in } t \\
(\!|\lambda x.tt'|\!) &= \mathbf{B}(\!|t|\!)(\!|\lambda x.t'|\!), \quad x \text{ only occurs free in } t' \\
(\!|\lambda x.tt'|\!) &= \mathbf{C}(\!|\lambda x.t|\!)(\!|t'|\!), \quad x \text{ only occurs free in } t \\
(\!|\lambda x.tt'|\!) &= \mathbf{S}(\!|\lambda x.t|\!)(\!|\lambda x.t'|\!), \quad x \text{ occurs free in } t \text{ and } t'
\end{aligned}
$$

- Let $t \sim_R t'$ stand for $R \models \lfloor t \rfloor \approx \lfloor t' \rfloor$ for some interpretation R. Where the interpretation is obvious, the subscript is omitted.

2 The Logic

The logic we use is polymorphic applicative first-order logic otherwise known as λ-free higher-order logic.

Syntax Let $\mathcal{V}_{ty}$ be a set of type variables and Σ_{ty} be a set of type constructors with fixed arities. It is assumed that a binary type constructor $\rightarrow$ is present in Σ_{ty} which is written infix. The set of types is defined:

Polymorphic Types $\quad \tau ::= \kappa(\overline{\tau_n}) \mid \alpha \mid \tau \rightarrow \tau \qquad$ where $\alpha \in \mathcal{V}_{ty}$ and $\kappa \in \Sigma_{ty}$

The notation $\overline{t_n}$ is used to denote a tuple or list of types or terms depending on the context. A type declaration is of the form $\Pi \overline{\alpha_n} . \sigma$ where σ is a type and all type variables in σ appear in $\overline{\alpha}$. Let Σ be a set of typed function symbols and $\mathcal{V}$ a set of variables with associated types. In what follows, $\mathsf{a}, \mathsf{b}, \mathsf{c}, \mathsf{f}, \mathsf{g}, \mathsf{h}$ represent function symbols and x, y, z variables. It is assumed that Σ contains the following function symbols, known as *basic combinators*:

> How to introduce combinators nicely?

$$\mathsf{S} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \tau \rightarrow \gamma) \rightarrow (\alpha \rightarrow \tau) \rightarrow \alpha \rightarrow \gamma$$
$$\mathsf{C} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \tau \rightarrow \gamma) \rightarrow \tau \rightarrow \alpha \rightarrow \gamma$$
$$\mathsf{B} : \Pi\alpha\tau\gamma.(\alpha \rightarrow \gamma) \rightarrow (\tau \rightarrow \alpha) \rightarrow \tau \rightarrow \gamma$$
$$\mathsf{K} : \Pi\alpha\gamma.\alpha \rightarrow \gamma \rightarrow \alpha$$
$$\mathsf{I} : \Pi\alpha.\alpha \rightarrow \alpha$$

The set of terms over Σ and $\mathcal{V}$ is defined below. In what follows, type subscripts are generally omitted.

Terms $\quad \mathcal{T} ::= x \mid \mathsf{f}\langle\overline{\tau_n}\rangle \mid t_{\tau' \rightarrow \tau} t'_{\tau'} \qquad$ where $\mathsf{f} : \Pi \overline{\alpha_n} . \sigma \in \Sigma, x \in \mathcal{V}$ and $t, t' \in \mathcal{T}$

An equation $s \approx t$ is an unordered pair of terms and a literal is an equation or the negation of an equation represented $s \not\approx t$. A clause is a multiset of literals represented $L_1 \vee \cdots \vee L_n$ where each L_i is a literal.

Semantics We follow Bentkamp et. al. [2] closely in specifying the semantics. An interpretation is a triple $(\mathcal{U}, \mathcal{E}, \mathcal{J})$ where $\mathcal{U}$ is a ground-type indexed family of non-empty sets called *universes* and $\mathcal{E}$ is a family of functions $\mathcal{E}_{\tau,v} : \mathcal{U}_{\tau \rightarrow v} \rightarrow (\mathcal{U}_\tau \rightarrow \mathcal{U}_v)$. A *type valuation* ξ is a total function from type variables to ground types. A type valuation ξ is extended to a *valuation* by setting $\xi(x_\tau)$ to be a member of $\mathcal{U}_{(\tau\xi)}$. An interpretation function $\mathcal{J}$ maps a function symbol $\mathsf{f} : \Pi \overline{\alpha_n} . \sigma$ and a tuple of ground types $\overline{\tau_n}$ to a member of $\mathcal{U}_{(\sigma\{\alpha_i \rightarrow \tau_i\})}$. An interpretation is *extensional* if $\mathcal{E}_{\tau,v}$ is injective for all τ, v and is *standard* if $\mathcal{E}_{\tau,v}$ is bijective for all τ, v.

For an interpretation $\mathcal{I} = (\mathcal{U}, \mathcal{E}, \mathcal{J})$ and a valuation ξ, a term is denoted as follows: $[\![x]\!]_{\mathcal{I}}^{\xi} = \xi(x)$, $[\![f\langle\overline{\tau}\rangle]\!]_{\mathcal{I}}^{\xi} = \mathcal{J}(f, [\![\overline{\tau}]\!]^{\xi})$ and $[\![st]\!]_{\mathcal{I}}^{\xi} = \mathcal{E}([\![s]\!]_{\mathcal{I}}^{\xi})([\![t]\!]_{\mathcal{I}}^{\xi})$. An equation $s \approx t$ is true in an interpretation $\mathcal{I}$ with valuation function ξ if $[\![s]\!]_{\mathcal{I}}^{\xi}$ and $[\![t]\!]_{\mathcal{I}}^{\xi}$ are the same object and is false otherwise. A disequation $s \not\approx t$ is true $s \approx t$ is false. A clause is true if one of its literals is true and a clause set is true if every clause in the set is true. An interpretation $\mathcal{I}$ models a clause set N, written $\mathcal{I} \models N$, if N is true in $\mathcal{I}$ for all valuation functions ξ.

3   The Calculus

The calculus is closely based on Bentkamp et al.'s intensional non-purifying calculus [2]. The extensionality axioms can be added if extensionality is required. The axiom of choice is included as we are working with Henkin semantic with choice. The main difference between our calculus and that of [2] is that superposition inferences are not allowed beneath fully applied combinators and an extra inference rule is addded to deal with superposition beneath variables.

Term Ordering We also demand the the calculus is parametrised by an ordering $\succ$ which orients all instances of combinator axioms left-to-right. It is an open problem whether a simplification ordering enjoying this property exists, but it appears unlikely. However, the $>_{\mathsf{ski}}$ ordering introduced in REF has most of the properties of a simplification ordering and orients all instances of combinator axioms left-to-right. It lacks two monotonicity properties. Firstly, similar to the ordering $>_{\mathsf{hz}}$ introduced in [1], it is not compatible with arguments. For terms t_1 and t_2 such that $t_1 >_{\mathsf{ski}} t_2$, it is not necessarily the case that $t_1 u >_{\mathsf{ski}} t_2 u$. The second property it lacks is compatibility with unstable contexts. For terms t_1 and t_2 such that $t_1 >_{\mathsf{ski}} t_2$, it is not necessarily the case that $\mathsf{S}\, t_1\, a\, b >_{\mathsf{ski}} \mathsf{S}\, t_2\, a\, b$. Bentkamp et al. show how to circumvent the first issue by tuning the side conditions of the superposition calculus. We show that by not superposing underneath fully applied combinators and carrying out some restricted superposition beneath variables, the second problem can be circumvented. In a number of places in the completeness proof we assume the following conditions on the ordering (satisfied by the $>_{\mathsf{ski}}$ ordering). It may be possible to relax the conditions at the expense of an increased number of inferences.

P1 For terms t, t' such that $t \longrightarrow_w t'$, then $t \succ t'$

P2 For terms t, t' such that $t \succ t'$ and $head(t')$ is first-order, $u[t] \succ u[t']$

> prove this in ordering paper?

The order $>_{\mathsf{ski}}$ is defined for ground terms. The definition is semantically lifted to non-ground terms as follows. For non-ground terms t and t', $t >_{\mathsf{ski}} t'$ iff $t\theta >_{\mathsf{ski}} t'\theta$ for all ground substitutions θ. Following Bentkamp et al. again, we define the quasi order $\succsim$ as follows [3]. For terms t and t', $t \succsim t'$ iff $t\theta \succeq t'\theta$ for all grounding substitutions θ.

Inference Rules The calculus is further parametrised by a selection function that maps clauses to a subset of its negative literals. Due to the requirements of the completeness proof, if a term $t = x\, \overline{s_{n>0}}$ is the maximal term in a clause C, then a literal containing x as a first-order subterm may not be selected. A literal L is called *eligible* if it is selected

or there are no selected literals in the clause and it is maximal. In the latter case, it is *strictly eligible* if it is strictly maximal. A variable x has a *bad* occurrence in a clause C if it occurs in C at an unstable position. Occurrences of x in C at stable positions are *good*. A clause C is an *extended combinator axiom* if it is either a combinator axiom or the result of an ARGCONG inference on a combinator axiom.

Conventions: Often a clause is written with a single distinguished literal such as $C' \vee t \approx t'$. In this case:

1. the distinguished literal is always eligible.
2. the name of the clause is assumed to be the name of the remainder without the dash.
3. if the clause is involved in an inference, the distinguished literal is the literal that takes part.

Positive and negative superposition:

$$\frac{D' \vee t \approx t' \qquad C' \vee [\neg]s\langle u\rangle \approx s'}{(C' \vee D' \vee [\neg]s\langle t'\rangle \approx s')\sigma} \text{ SUP}$$

with the following side conditions:

1. The variable condition holds
2. C is not an extended combinator axiom;
3. $\sigma = mgu(t, u)$;
4. $t\sigma \not\prec t'\sigma$;
5. $s\langle u\rangle\sigma \not\prec s'\sigma$;

6. $C\sigma \not\prec D\sigma$ or D is an extended combinator axiom;
7. $(t \approx t')\sigma$ is strictly eligible in $D\sigma$;
8. $([\neg]\,s\langle u\rangle \approx s')\sigma$ is eligible in $C\sigma$, and strictly eligible if it is positive.

Definition 1. *Let* $l = \tilde{\mathcal{C}}\,\overline{x_n} \approx r$ *be an extended combinator axiom. A term* $v\,\overline{u_m}$ *is compatible with* $l \approx r$ *if* $\tilde{\mathcal{C}} = \mathsf{I}$ *and* $m = n$ *or if* $\tilde{\mathcal{C}} = \mathsf{K}$ *and* $m \geq n - 1$ *or if* $\tilde{\mathcal{C}} \in \{\mathsf{B}, \mathsf{C}, \mathsf{S}\}$ *and* $m \geq n - 2$.

Variable condition: $u \notin \mathcal{V}$. If $u = x\,\overline{s_n}$ and D is a combinator axiom, then D and u must be compatible.

Because the term ordering $\succ$ is not compatible with unstable contexts, there are instances when superposition beneath variables must be carried out. The SUBVARSUP rule deals with this.

$$\frac{D' \vee t \approx t' \qquad C' \vee [\neg]s\langle y\,\overline{u_n}\rangle \approx s'}{(C' \vee D' \vee [\neg]s\langle zt'\,\overline{u_n}\rangle \approx s')\sigma} \text{ SUBVARSUP}$$

with the following side conditions in addition to conditions 3 - 8 of SUP:

condition 6 can be tightened

1. y has another occurrence bad in C;
2. z is a fresh variable;
3. $\sigma = \{y \to z\,t\}$;
4. t' has a variable or combinator head;
5. $n \leq 1$;
6. D is not an extended combinator axiom.

The EQRES and EQFACT inferences:

$$\frac{C' \vee s \not\approx s'}{C'\sigma} \ \text{EQRES} \qquad\qquad \frac{C' \vee u' \approx v' \vee u \approx v}{(C' \vee v \not\approx v' \vee u \approx v')\sigma} \ \text{EQFACT}$$

For EQRES: $\sigma = mgu(s, s')$ and $(s \not\approx s')\sigma$ is eligible in the premise. For EQFACT: $\sigma = mgu(u, u')$, $u'\sigma \not\precsim v'\sigma$, $u\sigma \not\precsim v\sigma$, and $(u \approx v)\sigma$ is eligible in the premise.

$$\frac{C' \vee s \approx s'}{C'\sigma \vee (s\sigma)\,\overline{x} \approx (s'\sigma)\,\overline{x}} \ \text{ARGCONG}$$

The literal $s\sigma \approx s'\sigma$ must be eligible in $C\sigma$ and $\overline{x}$ is a non-empty tuple of variables that are fresh for C. The unifier σ is the most general type unifier that ensures the well-typedness of the conclusion.

Along with the above inferences, we assume the presence of the (Skolemised) axiom of choice:

$$\neg p\,(\mathsf{sk}\langle\alpha\rangle\, p) \vee p\,(\varepsilon\langle\alpha\rangle\, p)$$

Where the function symbols sk and ε are of type $\Pi\alpha.(\alpha \to o) \to \alpha$.

4 Examples

We provide some examples of how the calculus works. Some of the examples utilised come from Bentkamp et. al.'s paper [3] in order to allow a comparison of the two methods. In all examples, it is assumed that the clause set has been enriched with the combinator axioms.

Example 1. Consider the unsatisfiable clause:

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

Superposing onto the left-hand side with the extended K axiom $\mathsf{K}\, x_1\, x_2\, x_3 \approx x_1\, x_3$ results in the clause $x_1\,\mathsf{b} \not\approx x_1\,\mathsf{a}$. Superposing onto the left-hand side of this clause, this time with the standard K axiom adds the clause $x \not\approx x$ from which $\bot$ is derived by an EQRES inference.

Example 2. Consider unsatisfiable the clause set where $\mathsf{f}\,\mathsf{a} \succ \mathsf{c}$:

$$\mathsf{f}\,\mathsf{a} \approx \mathsf{c} \qquad \mathsf{h}\,(y\,\mathsf{b})(y\,\mathsf{a}) \not\approx \mathsf{h}\,(\mathsf{g}(\mathsf{f}\,\mathsf{b}))(\mathsf{g}\,\mathsf{c})$$

A SUP inference between the B axiom the subterm $y\,\mathsf{b}$ of the second clause adds the clause $\mathsf{h}(x_1(x_2\,\mathsf{b}))(x_1(x_2\,\mathsf{a})) \not\approx \mathsf{h}(\mathsf{g}(\mathsf{f}\,\mathsf{b}))(\mathsf{g}\,\mathsf{c})$ to the set. By superposing onto the subterm $x_2\,\mathsf{a}$ of this clause with the equation $\mathsf{f}\,\mathsf{a} \approx \mathsf{c}$, we derive the clause $\mathsf{h}(x_1(\mathsf{f}\,\mathsf{b}))(x_1\,\mathsf{c}) \not\approx \mathsf{h}(\mathsf{g}(\mathsf{f}\,\mathsf{b}))(\mathsf{g}\,\mathsf{c})$ from which $\bot$ can be derived by an EQRES inference.

Example 3. Consider unsatisfiable the clause set where $\mathsf{f}\,\mathsf{a} \succ \mathsf{c}$. This example is the combinatory equivalent of Bentkamp et. al.'s Example 6.

$$\mathsf{f}\,\mathsf{a} \approx \mathsf{c} \qquad \mathsf{h}\,(y\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f})\,\mathsf{a})\,y \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathsf{I}$$

A Sup inference between the extended $\mathbf{I}$ axiom $\mathbf{I}\,x_1\,x_2 \approx x_1\,x_2$ and the subterm $y\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f})\,\mathsf{a}$ of the second clause adds the clause $\mathsf{h}\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f}\,\mathsf{a})\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$ to the set. Superposing onto the subterm $\mathbf{B}\,\mathsf{g}\,\mathsf{f}\,\mathsf{a}$ of this clause with the $\mathbf{B}$ axiom results in the clause $\mathsf{h}\,(\mathsf{g}\,(\mathsf{f}\,\mathsf{a}))\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$. Superposition onto the subterm $\mathsf{f}\,\mathsf{a}$ with the first clause of the original set gives $\mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$ from which $\bot$ can be derived via EQRES.

Example 4. Consider unsatisfiable the clause set where $\mathsf{f}\,\mathsf{a} \succ \mathsf{c}$. This example is the combinatory equivalent of Bentkamp et. al.'s Example 6.

$$\mathsf{f}\,\mathsf{a} \approx \mathsf{c} \qquad \mathsf{h}\,(y\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f})\,\mathsf{a})\,y \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$$

A Sup inference between the extended $\mathbf{I}$ axiom $\mathbf{I}\,x_1\,x_2 \approx x_1\,x_2$ and the subterm $y\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f})\,\mathsf{a}$ of the second clause adds the clause $\mathsf{h}\,(\mathbf{B}\,\mathsf{g}\,\mathsf{f}\,\mathsf{a})\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$ to the set. Superposing onto the subterm $\mathbf{B}\,\mathsf{g}\,\mathsf{f}\,\mathsf{a}$ of this clause with the $\mathbf{B}$ axiom results in the clause $\mathsf{h}\,(\mathsf{g}\,(\mathsf{f}\,\mathsf{a}))\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$. Superposition onto the subterm $\mathsf{f}\,\mathsf{a}$ with the first clause of the original set gives $\mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I} \not\approx \mathsf{h}\,(\mathsf{g}\,\mathsf{c})\,\mathbf{I}$ from which $\bot$ can be derived via EQRES.

Note that in Examples 2 and 3, no use is made of SUBVARSUP even though the analogous FLUIDSUP rule in required in Bentkamp et. al.'s calculus. We have been unable to develop an example that requires the SUBVARSUP rule.

5 Redundancy Criteria

We define a weaker logic by an encoding $\lfloor\,\rfloor$ of ground terms into first-order terms with $\lceil\,\rceil$ as its inverse. The encoding works by indexing each symbol with its type arguments and argument number. For example, $\lfloor f \rfloor = f_0$, $\lfloor f\langle\overline{\tau}\rangle a \rfloor = f_1^{\overline{\tau}}(a_0)$. Fully applied combinators are translated to constants such that syntactically identical terms are translated to the same constant. For example, $\lfloor \mathbf{S}\,t_1\,t_2\,t_3 \rfloor = s_0$. The weaker logic is known as the *floor logic* whilst the original logic is called the *ceiling logic*.

An inference is the ground instance of an inference I if it is equivalent to I after the application of some grounding substitution θ to the premise(s) and conclusion of I and the result is still an inference.

A ground ceiling clause C is redundant with respect to a set of ground ceiling clauses N if $\lfloor C \rfloor$ is entailed by clauses in $\lfloor N \rfloor$ smaller than itself. An arbitrary ceiling clause C is redundant to a set of ceiling clauses N if all its ground instances are redundant with respect to $\mathcal{G}_\Sigma(N)$, the set of all ground instances of clauses in N. $Red(N)$ is the set of all clauses redundant with respect to N.

For ground inferences other than ARGCONG, an inference with right premise C and conclusion E is redundant with respect to a set of clauses N if $\lfloor E \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C \rfloor$. A non-ground inference is redundant if all its ground instances are redundant.

An ARGCONG inference from a combinator axiom is redundant with respect to a set of clauses N if its conclusion is in N. For any other ARGCONG inference, it is redundant with respect N if its premise is redundant with respect to N, or its conclusion is in N or redundant with respect to N. A set N is saturated upto redundancy if every inference with premises in N is redundant with respect to N.

6 Refutational Completeness

The completeness proof is presented in the style of Bentkamp et al. [2] who in turn follow the style of Waldmann [4]. Let N be a set of clauses saturated upto redundancy by the above set of inferences. Let $\mathcal{G}_\Sigma(N)$ be the set of all ground instances of N. By the fact that N is saturated upto redundancy and by the definition of inference redundancy for ARGCONG inferences on combinator axioms, it follows that the following extended combinator axioms are in N for all $n \in \mathbb{N}$.

$$
\begin{aligned}
\mathsf{I}\, x\, \overline{x'_n} &= x\, \overline{x'_n} \\
\mathsf{K}\, x\, y\, \overline{x'_n} &= x\, \overline{x'_n} \\
\mathsf{B}\, x\, y\, z\, \overline{x'_n} &= x\, (y\, z)\, \overline{x'_n} \\
\mathsf{C}\, x\, y\, z\, \overline{x'_n} &= x\, z\, y\, \overline{x'_n} \\
\mathsf{S}\, x\, y\, z\, \overline{x'_n} &= x\, z\, (y\, z)\, \overline{x'_n}
\end{aligned}
$$

Thus, there exists an infinite set $ECA \subseteq N$ consisting of all extended combinator axioms. The set of all ground instances of clauses in ECA is denoted by $\mathcal{G}_\Sigma(ECA)$ and $\mathcal{G}_\Sigma(ECA) \subseteq \mathcal{G}_\Sigma(N)$. As per [2] we build a model for $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ which can then be lifted to a model of $\mathcal{G}_\Sigma(N)$ and N. We define a term-rewriting system R_∞ and use it to construct an interpretation of $\lfloor \mathcal{G}_\Sigma(N) \rfloor$. We then use induction to prove that this interpretation is a model of $\lfloor \mathcal{G}_\Sigma(N) \rfloor$.

Candidate Interpretation We define the set of rewrite rules R_{ECA} as $\{l \to r \mid l \approx r \in \lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ and $l \succ r\}$. By the condition that the term order orient all instances of combinator axioms left-to-right, we have that R_{ECA} is the set of all combinator axioms turned into left-to-right rewrite rules.

Lemma 1. *Let $R_{ECA} = R'_{ECA} \cup \{l \to r\}$. Then, l is not reducible by any rule in R'_{ECA}.*

Proof. Let $l = \lfloor \tilde{\mathcal{C}}\, \overline{t_n} \rfloor$ and $l' = \lfloor \tilde{\mathcal{C}}\, \overline{t'_n} \rfloor$ be the left hand side of two difference members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$. By the definition of the floor translation $\lfloor \rfloor$ we have that $\lfloor \tilde{\mathcal{C}}\, \overline{t_n} \rfloor$ and $\lfloor \tilde{\mathcal{C}}\, \overline{t'_n} \rfloor$ are both constants and $\lfloor \tilde{\mathcal{C}}\, \overline{t_n} \rfloor \neq \lfloor \tilde{\mathcal{C}}\, \overline{t'_n} \rfloor$ proving the lemma.

For every clause $C \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$, induction on $\succ$ is used to defined sets of rewrite rules E_C and R_C. Assume that E_D has been defined for all clauses $D \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$ such that $D \prec C$. Then R_C is defined as $R_{ECA} \cup (\bigcup_{D \prec C} E_D)$. The set E_C contains the rewrite rule $s \to t$ if the following conditions are met. Otherwise $E_C = \emptyset$.

(a) $C = C' \vee s \approx t$
(b) $s \approx t$ is strictly maximal in C
(c) $s \succ t$
(d) C is false in R_C
(e) C' is false in $R_C \cup \{s \to t\}$

(f) s is not reducible by any rule in R_C

In this case C is called *productive*. Finally, R_∞ is defined as $R_{ECA} \cup (\bigcup_{C \in \lfloor \mathcal{G}_\Sigma(N) \rfloor} E_C)$. Note that due to the definition of R_C and condition (d) an extended combinator axiom is never productive.

Lemma 2. *The rewrite systems R_C and R_∞ are confluent and terminating.*

Proof. Condition (c) ensures that for every rule $s \to t$ in R_C or R_∞ we have $s \succ t$. By the well-foundedness of $\succ$ we have that R_C and R_∞ must be terminating.

By Lemma 1 there are no critical pairs between rules in R_{ECA}. Using this fact and condition (f), we have that there are no critical pairs between rules in R_C and R_∞. Absence of critical pairs implies local confluence. Local confluence and termination implies confluence.

Lemma 3. *If a clause D is true in R_D then it is true in R_C for all $C \succ D$ and in R_∞.*

Proof. As per Waldmann's proof [4]

Lemma 4. *If a clause $D = D' \lor s \approx t$ is productive then D' is false and $s \approx t$ is true in R_C and R_∞ for all $C \succ D$.*

Proof. As per Waldmann's proof [4]

Lemma 5. *Every member of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ is true in R_C for all $C \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$.*

Proof. By construction of R_{ECA}, all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in R_{ECA}. Thus, by definition of R_C all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in R_C.

Lemma 6 (Lifting non-SUP inferences). *Let $C\theta \in \mathcal{G}_\Sigma(N)$, such that the selected literals in $C\theta$ match those in C. Then, every EQRES, EQFACT and ground instance of an ARGCONG inference from $C\theta$ is the ground instance of a corresponding inference from C.*

Proof. The proof is identical to that in [2] ignoring purification.

Definition 2. *A ground inference between clauses $D\theta = D'\theta \lor t\theta \approx t'\theta$ and $C\theta$ is liftable in two circumstances:*

1. *The superposed subterm in $C\theta$ is not at or below a variable in C and the variable conditions holds between C and D*
2. *The superposed subterm in $C\theta$ is at a variable y in C which has another occurrence bad in C, $head(t'\theta)$ is a combinator, and D is not a combinator axiom.*
3. *The superposed subterm in $C\theta$ is below a variable y in C which has another occurrence bad in C, $head(y\theta)$ and $head(t'\theta)$ are combinators and D is not a combinator axiom.*

Lemma 7 (Lifting SUP inferences). *Let $D\theta$ and $C\theta$ be members of $\mathcal{G}_\Sigma(N)$ where θ is a substitution and the selected literals of $D, C \in N$ correspond to those of $D\theta$ and $C\theta$. Then for every liftable SUP inference between $C\theta$ and $D\theta$, there exists a ground substitution θ' such that $C\theta = C\theta'$, $D\theta = D\theta'$ and the conclusion of the inference between $C\theta$ and $D\theta$ is combinatory congruent to the conclusion of the θ' instance of a SUP or SUBVARSUP inference between C and D.*

Prove that $C\theta' = C\theta$.

Proof. We assume that $D = D' \vee t \approx t'$ and that $C = C' \vee [\neg]s \approx s'$ and that the inference has the form:

$$\frac{D'\theta \vee t\theta \approx t'\theta \qquad C'\theta \vee [\neg]s\theta\langle t\theta\rangle|_p \approx s'\theta}{C'\theta \vee D'\theta \vee [\neg]s\theta\langle t'\theta\rangle|_p \approx s'\theta} \;\text{S{\scriptsize UP}}$$

where $t\theta \approx t'\theta$ is strictly eligible, $[\neg]s\theta \approx s'\theta$ is strictly eligible if positive and eligible if negative, $C\theta \not\precsim D\theta$, $t\theta \not\precsim t'\theta$ and $s\theta \not\precsim s'\theta$. The proof is broken into two cases.

Case 1: $t\theta$ is not at or beneath a variable in C. In this case, p must be a position of s. Let $u = s|_p$. We have that θ is a unifier of u and t and therefore, there must exist an idempotent mgu σ of t and u. The inference conditions can be lifted. $C\theta \not\precsim D\theta$ implies $C \not\precsim D$, $t\theta \not\precsim t'\theta$ implies $t \not\precsim t'$ and $s\theta \not\precsim s'\theta$ implies $s \not\precsim s'$. Moreover $t\theta \approx t'\theta$ being strictly eligible in $D\theta$ implies that $t \approx t'$ is strictly eligible in D and $[\neg]s\theta \approx s'\theta$ being (strictly) eligible in $C\theta$ implies that $[\neg]s \approx s'$ is (strictly) eligible in C. By liftability, we have that the variable condition holds between D and C. Therefore there is the following S{\scriptsize UP} inference between D and C:

$$\frac{D' \vee t \approx t' \qquad C' \vee [\neg]s\theta\langle t\rangle|_p \approx s'}{(C' \vee D' \vee [\neg]s\langle t'\rangle|_p \approx s')\sigma} \;\text{S{\scriptsize UP}}$$

We have $(C' \vee D' \vee [\neg]s\langle t'\rangle|_p \approx s')\sigma\theta = (C' \vee D' \vee [\neg]s\langle t'\rangle|_p \approx s')\theta = C'\theta \vee D'\theta \vee [\neg]s\theta\langle t'\theta\rangle|_p \approx s'\theta$ by the idempotency of σ proving that the conclusion of the ground inference is the θ-ground instance of the conclusion of the non-ground inference and we can take $\theta' = \theta$.

Case 2: $t\theta$ is at or beneath a variable in C which has another instance bad in C. In this case, there must be exist p', p'' such that $p = p'.p''$ and $s|_{p'} = y\,\overline{u_n}$. Let $u = s|_{p'}$. As per case 1, (strict) eligibility of the ground literals implies (strict) eligibility of the non-ground literals. Further $t'\theta$ having a combinator head implies that t' has a variable or combinator head.

If the inference is beneath a variable in C, then by liftability, $y\theta = \tilde{\mathcal{C}}\,\overline{v_{m>0}}$. If $u = y\,\overline{u_{n>1}}$, then $u\theta = \tilde{\mathcal{C}}\,\overline{v_m}\,(\overline{u_n})\theta$. Thus, $t\theta$ which is a proper subterm of $u\theta$ would be beneath a fully applied combinator which is impossible. Therefore, we have that $n \leq 1$. If the inference is at a variable in C, then we must have $n = 0$ otherwise $y\theta = t\theta$ would not be a first-order subterm.

Thus in both cases $n \leq 1$ and there is a S{\scriptsize UB}V{\scriptsize AR}S{\scriptsize UP} inference between C and D:

$$\frac{D' \vee t \approx t' \qquad C' \vee [\neg]s\theta\langle y\,\overline{u_n}\rangle|_{p'} \approx s'}{(C' \vee D' \vee [\neg]s\langle zt'\,\overline{u_n}\rangle|_{p'} \approx s')\{y \to zt\}} \;\text{S{\scriptsize UB}V{\scriptsize AR}S{\scriptsize UP}}$$

where z is a fresh variable. Define the substitution θ' that maps z to $(\!(\lambda x.(y\theta)\langle x\rangle|_{p''})\!)$ and all other variables w to $w\theta$. Since z is fresh, $C\theta = C\theta'$ and $D\theta = D\theta'$. Further, we have that $(zt')\theta' = (\!(\lambda x.(y\theta)\langle x\rangle|_{p''})\!)t'\theta \to_w^* y\theta\langle t'\theta\rangle|_{p''}$ and $(zt)\theta' \to_w^* y\theta\langle t\theta\rangle|_{p''} = y\theta$. Therefore, $(C'\vee D'\vee[\neg]s\langle zt'\,\overline{u_n}\rangle|_{p'} \approx s')\{y \to zt\}\theta' = (C'\vee D'\vee[\neg]s\langle(\!(\lambda x.(y\theta)\langle x\rangle|_{p''})\!)t'\,\overline{u_n}\rangle\theta[y \to z\theta't\theta] \to_w^* C'\theta\vee D'\theta\vee[\neg]s\theta\langle t'\theta\rangle|_p \approx s'\theta$ proving that the conclusion of the ground inference is combinatory congruent to the θ'-instance of the conclusion of the non-ground inference.

Model Construction

In this section the candidate interpretation R_∞ developed in the previous section is shown to be a model of $\lfloor \mathcal{G}_\Sigma(N) \rfloor$. As per the standard proof, this is done by induction on the clause order $\succ$. For some fixed clause $\lfloor C\theta \rfloor \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$, by Lemma 5, we have that all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in $R_{\lfloor C\theta \rfloor}$. By the induction hypothesis, we have that for all clauses $\lfloor D\theta \rfloor \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $C\theta$ are true in $R_{\lfloor C\theta \rfloor}$. In the induction step, it is shown that this implies that $\lfloor C\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$.

Lemma 8. *Let $C\theta, D\theta \in \mathcal{G}_\Sigma(N)$ where the selected literals of $C, D \in N$ correspond with those of $C\theta$ and $D\theta$. Let E be the conclusion of a liftable* SUP *inference from $C\theta$ and $D\theta$ or an* EQRES *or* EQFACT *inference from $C\theta$. Assume that $C\theta$ and $D\theta$ are not redundant with respect to $\mathcal{G}_\Sigma(N)$. Then, $\lfloor E \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ along with clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$.*

Proof. By Lemmas 6 and 7, we have that the conclusion of a SUP inference form $C\theta$ and $D\theta$ or an EQRES or EQFACT inference from $C\theta$ is combinatory congruent to the conclusion of the θ'-ground instance of an inference from C and D (or just C as the case may be) where θ' is a ground substitution such that $D\theta' = D\theta$ and $C\theta' = C\theta$. Let E' be the conclusion of the inference between C and D, we have that $E'\theta' \rightarrow^*_w E$. Because N is saturated upto redundancy, the inference is redundant with respect to N. Thus, the θ' ground instance of the inference is redundant with respect to $\mathcal{G}_\Sigma(N)$. By the definition of inference redundancy, this means that $\lfloor E'\theta' \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta' \rfloor = \lfloor C\theta \rfloor$. Using this and the fact that $E'\theta' \rightarrow^*_w E$, we have that $\lfloor E \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ along with clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$.

Lemma 9. *Let R be an interpretation such that every member of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ holds in R. Let u and u' be ground terms such that $u\,\overline{s} \sim_R u'\,\overline{s}$ for every type correct tuple of ground terms $\overline{s}$. Then for ground terms s and s' such that s and s' only differ at positions where s contains u and s' contains u', we have $s \sim_R s'$.*

Proof. The proof proceeds by showing how equations that are true in R can be used to rewrite s into s'. Since R is a model of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ all ground instances of combinator and extended combinator axioms are true in R.

Let $s_0 = \lfloor s \rfloor$ and $\tilde{s}_0 = \lfloor s' \rfloor$. Terms $s_1, s_2, s_3 \ldots$ are defined inductively as follows: s_{i+1} is formed from s_i by rewriting all subterms of the form $\lfloor u\,\overline{v} \rfloor$ in s_i to $\lfloor u'\,\overline{v} \rfloor$ and then reducing the outermost leftmost weak-redex in the resulting term. Terms $\tilde{s}_1, \tilde{s}_2, \tilde{s}_3 \ldots$ are also defined inductively: $\tilde{s}_{i+1}$ is formed from $\tilde{s}_i$ by reducing the left-most outermost weak-redex in $\tilde{s}_i$.

The algorithm maintains the invariant that for all i, $\lceil s_i \rceil$ and $\lceil \tilde{s}_i \rceil$ are identical other than at positions where $\lceil s_i \rceil$ contains a u and $\lceil \tilde{s}_i \rceil$ contains a u'. Let $\tilde{s}_*$ be the final term in the $\tilde{s}_n$ series. Such a term must exist as weak reduction is terminating. Let s_* be the equivalent term in s_n series. We show that $s_* = \tilde{s}_*$.

Assume that $s_* \neq \tilde{s}_*$. Consider the outermost position at which $\lceil s_* \rceil$ and $\lceil \tilde{s}_* \rceil$ differ. If this position is not beneath a fully applied combinator in $\lceil s_* \rceil$, then s_* and $\tilde{s}_*$ must contain $\lfloor u\,\overline{v} \rfloor$ and $\lfloor u'\,\overline{v'} \rfloor$ at the corresponding positions. But this is impossible

since $\lfloor u\,\overline{v}\,\rfloor$ would have been rewritten to $\lfloor u'\,\overline{v'}\,\rfloor$ at the previous step of the algorithm. Therefore assume that the outermost position at which $\lceil s_*\rceil$ and $\lceil \tilde{s}_*\rceil$ differ is beneath a fully applied combinator in $\lceil s_*\rceil$. Since we are considering the outermost position, this combinator cannot be a part of u. Thus we must have that the same combinator occurs fully applied in $\lceil \tilde{s}_*\rceil$. This contradicts the fact that $\tilde{s}_*$ is in weak normal form. Therefore the original assumption is false and $s_* = \tilde{s}_*$.

Lemma 10. *Let R be an interpretation such that every member of $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$ holds in R. Let u and u' be ground terms such that $u \sim_R u'$. Let $s[]_n$ be a ground context with n holes at stable positions. Then $s[\,\overline{u}\,]_n \sim_R s[\,\overline{u'}\,]_n$.*

definie multi-hole contexts

Proof. The proof proceeds by induction on $\|s[]_n\|$. In the base case $\|s[]_n\| = 0$ and no instance of u occurs beneath a fully applied combinator. By the definition of stable positions, no instance of u occurs with arguments. Thus, $\lfloor s[\,\overline{u}\,]_n\rfloor$ can be rewritten to $\lfloor s[\,\overline{u'}\,]_n\rfloor$ directly using the equation $\lfloor u \approx u'\rfloor$.

For the inductive case, we have that $\|s[]_n\| > 0$. This splits into two cases depending on whether any of the holes are underneath the leftmost fully applied combinator or not.

Case 1: There are no holes underneath the leftmost fully applied combinator. Consider the context $s'[]_n$ formed by reducing the leftmost fully applied combinator in s. Because $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$ holds in R, we have $s[\,\overline{u}\,]_n \sim_R s'[\,\overline{u}\,]_n$. By the induction hypothesis we have $s'[\,\overline{u}\,]_n \sim_R s'[\,\overline{u'}\,]_n$ and then using the relevant member of $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$, we have $s'[\,\overline{u'}\,]_n \sim_R s[\,\overline{u'}\,]_n$. By the transitivity of $\sim_R$, we have $s[\,\overline{u}\,]_n \sim_R s[\,\overline{u'}\,]_n$.

Case 2: The leftmost fully applied combinator is of the form $\tilde{\mathcal{C}}\ \overline{t_k}$ where at least one t_i has a hole as a proper subterm. Note that by the definition of stable subterms, none of the t_is can be a hole. Consider the context $s'[]_m$ formed by replacing $\tilde{\mathcal{C}}\ \overline{t_k}$ with $(\tilde{\mathcal{C}}\ \overline{t_k})\downarrow^w$ in s. Because any holes occurring as subterms of $\tilde{\mathcal{C}}\ \overline{t_k}$ occur at stable positions, any holes occurring in $(\tilde{\mathcal{C}}\ \overline{t_k})\downarrow^w$ also occur at stable positions. Thus all holes in $s'[]_m$ occur at stable positions. Because $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$ holds in R, we have $s[\,\overline{u}\,]_n \sim_R s'[\,\overline{u}\,]_m$. By the induction hypothesis we have $s'[\,\overline{u}\,]_m \sim_R s'[\,\overline{u'}\,]_m$ and then using the relevant member of $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$, we have $s'[\,\overline{u'}\,]_m \sim_R s[\,\overline{u'}\,]_n$. By the transitivity of $\sim_R$, we have $s[\,\overline{u}\,]_n \sim_R s[\,\overline{u'}\,]_n$.

Lemma 11. *Let R be an interpretation such that every member of $\lfloor \mathcal{G}_\Sigma(ECA)\rfloor$ holds in R. Let u and u' be ground terms such that $head(u')$ is first-order and $u \succ u'$. Let $s[]_n$ be a ground context with n holes. Furthermore, assume that for all type correct tuple of ground terms $\overline{t}$, $R \models \lfloor u\,\overline{t} \approx u'\,\overline{t}\,\rfloor$ whenever $s[u]_n \succeq u\,\overline{t}$ or $s[u]_n \succ u\,\overline{t}$ if $head(s[u]_n) \not\approx u$. Then $s[\,\overline{u}\,]_n \sim_R s[\,\overline{u'}\,]_n$.*

Proof. Proof is by induction on $\|s[]_n\|$. If $\|s[]_n\| = 0$ then $\lfloor s[\,\overline{u}\,]_n\rfloor$ can be rewritten to $\lfloor s[\,\overline{u'}\,]_n\rfloor$ by rewriting all occurrences of $\lfloor u\,\overline{t}\,\rfloor$ to $\lfloor u'\,\overline{t}\,\rfloor$ starting with the outermost occurrence. By property P2 of the ordering, after each rewrite the resulting term is smaller than the original term.

The inductive case again splits into two depending on whether the leftmost fully applied combinator in $s[]_n$ has any holes beneath it or not. If it does not, the proof proceeds identically to that of 10.

Otherwise, The leftmost fully applied combinator is of the form $\tilde{\mathcal{C}}\ \overline{t_k}$ where at least one t_i has a hole as a subterm. Consider the context $s'[\,]_m$ formed from $s[\,]_n$ by rewriting $\tilde{\mathcal{C}}\ \overline{t_k}$ with the equation:

$$\tilde{\mathcal{C}}\ \overline{t_k}\ =\ (\tilde{\mathcal{C}}\ \overline{t_k})\!\downarrow^w \tag{*}$$

Since $s[\,\overline{u}\,]_n \to_w s'[\,\overline{u}\,]_m$, we have $s[\,\overline{u}\,]_n \succ s'[\,\overline{u}\,]_m$. Via the induction hypothesis $s'[\,\overline{u}\,]_m \sim_R s'[\,\overline{u'}\,]_m$. Using (*) right-to-left, we have $s'[\,\overline{u'}\,]_m \sim_R s[\,\overline{u'}\,]_n$ and the theorem follows by the tansitivity of $\sim_R$.

Lemma 12. *Let $C\theta, D\theta \in \mathcal{G}_\Sigma(N)$, where the selected literals of $C\theta = C'\theta \vee [\neg]s\theta \approx s'\theta$ and $D\theta = D'\theta \vee t\theta \approx t'\theta$ match those of $C, D \in N$. Consider a non-liftable* SUP *inference between $C\theta$ and $D\theta$. Assume that $C\theta$ and $D\theta$ are not redundant with respect to $\mathcal{G}_\Sigma(N)$. Then $\lfloor C\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than it, clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ and $\neg\lfloor D'\theta \rfloor$.*

Proof. Assume that the $s\theta \succ s'\theta$ and $t\theta \succ t'\theta$. Let R be an interpretation such that clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ and all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in R whilst $\lfloor D'\theta \rfloor$ is false. By the SUP conditions, we have that either $C\theta \succ D\theta$ or $D\theta \in \mathcal{G}_\Sigma(ECA)$ and in both cases $R \models \lfloor t\theta \approx t'\theta \rfloor$. We need to show that this implies $R \models \lfloor C\theta \rfloor$.

The inference can be non-liftable for one of the following reasons:

1. The variable condition not holding between C and D;
2. It is at or below a variable in C all of whose other occurrences are good in C;
3. It is below a variable x in C and $x\theta$ has a first-order head;
4. It is at or below a variable in C and $t'\theta$ has a first-order head;
5. It is at or below a variable in C and D is an extended combinator axiom;

We fix some terminology common to cases 2-5. Let x be the variable at or beneath which the inference takes place. Then $t\theta$ is a first-order subterm of $x\theta$. Let v be the result of replacing $t\theta$ by $t'\theta$ in $x\theta$ at the relevant position. If the inference is beneath x, by congruence and the fact that $R \models t\theta \approx t'\theta$ we have $R \models x\theta\,\overline{w} \approx v\,\overline{w}$ for any tuple of terms $\overline{w}$. Let $C' = C\theta[x \to v]$.

Case 1: The variable condition fails to hold because D is an extended combinator axiom not compatible with u By the definition of compatibility, $u = x\,\overline{s_m}$ and $t = \tilde{\mathcal{C}}\ \overline{x_n} \approx t'$ and $\tilde{\mathcal{C}}\ \overline{x_{n-m}}$ is a fully applied combinator. Thus, $x\theta = (\tilde{\mathcal{C}}\ \overline{x_{n-m}})\theta$ is also a fully applied combinator. Let $C'' = C\{x \mapsto ((\tilde{\mathcal{C}}\ \overline{x_{n-m}})\theta)\!\downarrow^w\}$. Because the maximal weak reduction from the largest term in $C\theta$ is greater than the maximal weak reduction from the largest term in $C''\theta$, we have $C\theta \succ C''\theta$ and thus $R \models \lfloor C''\theta \rfloor$. But then, as $C\theta \longrightarrow_w C''\theta$ and all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in R, we have $R \models \lfloor C\theta \rfloor$ by congruence.

Case 2: The superposed subterm $t\theta$ is at or beneath a variable x in C all of whose other occurences are good in C. From the existence of the inference between $C\theta$ and $D\theta$, it follows that $t\theta$ must occur at a first-order and thus stable position in $x\theta$. Likewise, the existence of the inference implies that the $x\theta$ involved in the inference occurs at a stable position within $s\theta$. For all other instances of $x\theta$, the fact that x is good in C implies that $x\theta$ is stable in $C\theta$. Thus, by compatibility with stable contexts and the fact that

$t\theta \succ t'\theta$, we have that $C\theta \succ C'$ and thus by assumption on R, $R \models \lfloor C' \rfloor$. Since $C\theta$ and C' only differ at stable positions where one contains $x\theta$ and the other v, Lemma 10 can be used to show that every literal of $C\theta$ is equivalent in R to the corresponding literal of C'. This implies $R \models \lfloor C\theta \rfloor$ by congruence.

Case 3: The superposed subterm $t\theta$ is beneath a variable x in C and $x\theta$ has a first order head. We have that $x\theta = (\mathsf{f}\ \overline{s_n})\langle t\theta \rangle$ and $v = (\mathsf{f}\ \overline{s_n})\langle t'\theta \rangle$. Thus, $x\theta \succ v$ follows from the ordering's compatibility with stable context. As v has a first order head and C' is formed from $C\theta$ by replacing $x\theta$ with v, $C\theta \succ C'$ follows from property P2 of the ordering. Hence, $R \models \lfloor C' \rfloor$ and $R \models \lfloor C\theta \rfloor$ via Lemma 9 and congruence.

Case 4: The superposed subterm $t\theta$ is at or beneath a variable x in C and $t'\theta$ has a first order head. We have $t\theta \succ t'\theta$ and C' can be formed from $C\theta$ by replacing all occurrences of t with t'. Thus, $C\theta \succ C'$ follows from property P2 of the ordering. Hence, $R \models \lfloor C' \rfloor$. If the inference is beneath a variable in C, then $R \models \lfloor C\theta \rfloor$ via Lemma 9 and congruence.

Otherwise, if the inference is at a variable in C, we perform a case analysis depending on whether $[\neg]s\theta \approx s'\theta$ is a maximal or selected literal in $C\theta$. First though, note that for all type correct tuple of ground terms $\overline{u}$, if $t\theta\,\overline{u}$ and $t'\theta\,\overline{u}$ are smaller than the maximal term of $C\theta$ then:

$$R \models \lfloor t\theta\,\overline{u} \approx t'\theta\,\overline{u} \rfloor$$

This can be shown exactly as Bentkamp et. al. do so [2]. If $[\neg]s\theta \approx s'\theta$ is the maximal literal in $C\theta$, consider the clause C'' formed by rewriting $\lfloor t\theta \rfloor$ to $\lfloor t'\theta \rfloor$ wherever possible in $C\theta$. As $t\theta$ is a first-order subterm of the maximal term of $C\theta$, $s\theta$, we have that $C\theta \succ C''$. Further, we have that C'' and C' differ only at positions where C'' contains a t and C contains a t'. We can thus use Lemma 11 and congruence to show that C'' is true in R and hence $R \models C\theta$.

If $[\neg]s\theta \approx s'\theta$ is a selected literal, then by the selection criteria x cannot be the head of the maximal literal of $C\theta$. Hence Lemma 11 and congruence can be used to prove that $R \models C\theta$.

Case 5: The superposed subterm $t\theta$ is at or beneath a variable x in C and $t\theta \approx t'\theta$ is a member of $\mathcal{G}_\Sigma(ECA)$. In this case $C\theta \longrightarrow_w C'$ and thus $C\theta \succ C'$ by property P1 of the ordering. Hence, $R \models \lfloor C' \rfloor$. Since all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in R, every side of a literal in $C\theta$ is equal in R to the equivalent term in C' and $R \models C\theta$ follows by congruence.

Lemma 13 (R_∞ **is a model**)**.** *Let* $\lfloor C\theta \rfloor \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$, *then*

(i) $E_{\lfloor C\theta \rfloor} = \emptyset$ *if and only if* $R_{\lfloor C\theta \rfloor} \models C\theta$;
(ii) *if* $C\theta$ *is redundant with respect to* $\mathcal{G}_\Sigma(N)$ *then* $\lfloor C\theta \rfloor$ *is true in* $R_{\lfloor C\theta \rfloor}$;
(iii) $\lfloor C\theta \rfloor$ *holds in* R_∞ *and* R_D *for all* $D \in \lfloor \mathcal{G}_\Sigma(N) \rfloor, D \succ C\theta$;
(iv) *if* $C\theta$ *has selected literals, then* $R_{\lfloor C\theta \rfloor} \models \lfloor C\theta \rfloor$;
(v) *if* $C\theta$ *is a member of* $\mathcal{G}_\Sigma(ECA)$, *then* $R_{\lfloor C\theta \rfloor} \models \lfloor C\theta \rfloor$.

Proof. The proof proceeds by induction on ground clauses of the floor logic. We assume that (i) to (v) are satisfied by all clauses $D \in \lfloor \mathcal{G}_\Sigma(N) \rfloor$ such that $\lfloor C\theta \rfloor \succ D$. We prove that (i) to (v) hold for $C\theta$. The $\Rightarrow$ direction of (i) follows from the construction. Part

(iii) follows from (i) by Lemmas 3 and 4. Part (v) is a straightforward extension of 5. Therefore it remains to prove the $\Leftarrow$ direction of (i) and (ii) and (iv) for the case where $C\theta \notin \mathcal{G}_\Sigma(ECA)$. We assume that the selected literals of $C\theta$ match those of $C \in N$.

Case 1: $C\theta$ is redundant with respect to $\lfloor \mathcal{G}_\Sigma(N) \rfloor$. Then $\lfloor C\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ that are smaller than it. By part (iii) of the induction hypothesis, these clauses are true $R_{\lfloor C\theta \rfloor}$ and therefore $\lfloor C\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$.

Case 2: $C\theta = C'\theta \vee s\theta \not\approx s'\theta$ and $C\theta$ is not redundant with respect to $\mathcal{G}_\Sigma(N)$.

Case 2.1: $s\theta = s'\theta$. In this case, there is an EqRes inference from $C\theta$:

$$\frac{C'\theta \vee s\theta \not\approx s'\theta}{C'\theta} \text{ EqRes}$$

by Lemma 8, we have that $\lfloor C'\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$. By part (iii) of the induction hypothesis, these clauses are true in $R_{\lfloor C\theta \rfloor}$. Therefore, $\lfloor C'\theta \rfloor$ and thus $\lfloor C\theta \rfloor$ are true in $R_{\lfloor C\theta \rfloor}$.

Case 2.2: $s\theta \succ s'\theta$. If $R_{\lfloor C\theta \rfloor} \models s\theta \not\approx s'\theta$ then the lemma follows. Therefore, assume that it doesn't hold and $s\theta \downarrow_{R_{\lfloor C\theta \rfloor}} s'\theta$. There must exist some rule in $R_{\lfloor C\theta \rfloor}$ which reduces $s\theta$. Such a rule must have been produced by some clause $\lfloor D\theta \rfloor = \lfloor D'\theta \vee t\theta \approx t'\theta \rfloor$. Without loss of generality, it is assumed that the C and D are variable disjoint, so that the same substitution θ can be used. It is also assumed that the selected literals of $D\theta$ match those of D. Then there exists the following Sup inference:

$$\frac{D'\theta \vee t\theta \approx t'\theta \qquad C'\theta \vee s\theta\langle t\theta\rangle \not\approx s'\theta}{C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \not\approx s'\theta} \text{ Sup}$$

by Lemma 12, if the inference is non-liftable then $\lfloor C\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ that are smaller than it, by $\neg\lfloor D'\theta \rfloor$ and by clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$. By part (iii) of the induction hypothesis, clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $C\theta$ are true in $R_{\lfloor C\theta \rfloor}$. By Lemma 5 all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in $R_{\lfloor C\theta \rfloor}$. By Lemma 4 $\neg\lfloor D'\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$. Therefore $C\theta$ is true in $R_{\lfloor C\theta \rfloor}$.

If the inference is liftable then by Lemma 8, we have that $\lfloor C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \not\approx s'\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ along with clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$. By part (iii) of the induction hypothesis and Lemma 5, all these clauses are true in $R_{\lfloor C\theta \rfloor}$. Therefore, $\lfloor C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \not\approx s'\theta \rfloor$ holds in $R_{\lfloor C\theta \rfloor}$. Since $\lfloor D'\theta \rfloor$ is false in $R_{\lfloor C\theta \rfloor}$, either $R_{\lfloor C\theta \rfloor} \models \lfloor C'\theta \rfloor$ or $R_{\lfloor C\theta \rfloor} \models \lfloor s\theta\langle t'\theta\rangle \not\approx s'\theta \rfloor$. In the latter case, $R_{\lfloor C\theta \rfloor} \models \lfloor s\theta\langle t\theta\rangle \not\approx s'\theta \rfloor$ because $\lfloor t\theta \rfloor \to \lfloor t'\theta \rfloor \in R_{\lfloor C\theta \rfloor}$. Thus, in both cases $R_{\lfloor C\theta \rfloor} \models \lfloor C\theta \rfloor$.

Case 3: $C\theta$ is not redundant and contains no negative literals. In this case $C\theta = C'\theta \vee s\theta \approx s'\theta$. If $E_{\lfloor C\theta \rfloor} = \{\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor\}$ or $R_{\lfloor C\theta \rfloor} \models C'\theta$ or $s\theta = s'\theta$ then $R_{\lfloor C\theta \rfloor} \models C\theta$. Therefore, assume $E_{\lfloor C\theta \rfloor} = \emptyset$, $s\theta \neq s'\theta$ and $C'\theta$ is false in $R_{\lfloor C\theta \rfloor}$.

Case 3.1: $\lfloor s\theta \approx s'\theta \rfloor$ is not strictly maximal in $C\theta$. In this case, $C\theta$ can be written as $C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta$ where $t\theta = s\theta$ and $t'\theta = s'\theta$. Then there is an EqFact inference from $C\theta$:

$$\frac{C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta}{C''\theta \vee t'\theta \not\approx s'\theta \vee t\theta \approx t'\theta} \text{ EqFact}$$

by Lemma 8 the conclusion of the inference is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ and therefore by part (iii) of the induction hypothesis true in $R_{\lfloor C\theta \rfloor}$. Since $t'\theta = s'\theta$, $\lfloor t'\theta \not\approx s'\theta \rfloor$ is false in $R_{\lfloor C\theta \rfloor}$. Therefore $\lfloor t\theta \approx t'\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$ and hence $\lfloor C\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$.

Case 3.2: The literal $s\theta \approx s'\theta$ is strictly maximal and $\lfloor s\theta \rfloor$ is reducible by some rule in $R_{\lfloor C\theta \rfloor}$. This rule must be produced by a clause $\lfloor D\theta \rfloor = \lfloor D'\theta \vee t\theta \approx t'\theta \rfloor$. Without loss of generality, we assume that D and C are variable disjoint so that the same substitution θ can be used. We also assume that the selected literals of $D\theta$ match those of $\lfloor D \rfloor$. then there is an SUP inference:

$$\frac{D'\theta \vee t\theta \approx t'\theta \qquad C'\theta \vee s\theta\langle t\theta\rangle \approx s'\theta}{C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \approx s'\theta} \text{ SUP}$$

by Lemma 12, if the inference is non-liftable then $\lfloor C\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ that are smaller than it, by $\neg\lfloor D'\theta \rfloor$ and by clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$. By part (iii) of the induction hypothesis, clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $C\theta$ are true in $R_{\lfloor C\theta \rfloor}$. By Lemma 5 all members of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ are true in $R_{\lfloor C\theta \rfloor}$. By Lemma 4 $\neg\lfloor D'\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$. Therefore $C\theta$ is true in $R_{\lfloor C\theta \rfloor}$.

If the inference is liftable then by Lemma 8, we have that $\lfloor C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \approx s'\theta \rfloor$ is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ along with clauses in $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$. By part (iii) of the induction hypothesis and Lemma 5, all these clauses are true in $R_{\lfloor C\theta \rfloor}$. Therefore, $\lfloor C'\theta \vee D'\theta \vee s\theta\langle t'\theta\rangle \approx s'\theta \rfloor$ holds in $R_{\lfloor C\theta \rfloor}$. Since $\lfloor D'\theta \rfloor$ is false in $R_{\lfloor C\theta \rfloor}$, either $R_{\lfloor C\theta \rfloor} \models \lfloor C'\theta \rfloor$ or $R_{\lfloor C\theta \rfloor} \models \lfloor s\theta\langle t'\theta\rangle \approx s'\theta \rfloor$. In the latter case, $R_{\lfloor C\theta \rfloor} \models \lfloor s\theta\langle t\theta\rangle \approx s'\theta \rfloor$ because $\lfloor t\theta \rfloor \to \lfloor t'\theta \rfloor \in R_{\lfloor C\theta \rfloor}$. Thus, in both cases $R_{\lfloor C\theta \rfloor} \models \lfloor C\theta \rfloor$.

Case 3.3: $s\theta \approx s'\theta$ is strictly maximal and $\lfloor s\theta \rfloor$ is not reducible by any rule in $R_{\lfloor C\theta \rfloor}$. By assumption we have that $E_{\lfloor C\theta \rfloor} = \emptyset$. Assume that $\lfloor C\theta \rfloor$ is false in $R_{\lfloor C\theta \rfloor}$. By the construction of $E_{\lfloor C\theta \rfloor}$, it must be the case that $C'\theta$ is true in $R_{\lfloor C\theta \rfloor} \cup \{\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor\}$. Thus $C'\theta$ must have the form $C''\theta \vee t\theta \approx t'\theta$ where the literal $\lfloor t\theta \approx t'\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor} \cup \{\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor\}$, but not in $R_{\lfloor C\theta \rfloor}$. By the confluence of $R_{\lfloor C\theta \rfloor}$, this is equivalent to saying $t\theta \downarrow_{R_{\lfloor C\theta \rfloor} \cup \{\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor\}} t'\theta$, but not $t\theta \downarrow_{R_{\lfloor C\theta \rfloor}} t'\theta$. Therefore, the rule $\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor$ must be used at least once in rewriting either $t\theta$ or $t'\theta$ to a normal form. As $s\theta \succ s'\theta$, $t\theta \succ t'\theta$ and $s\theta \approx s'\theta \succ t\theta \approx t'\theta$, we have that $s\theta \succ t'\theta$ and $s\theta \succeq t\theta$. But the fact that $\lfloor s\theta \rfloor \to \lfloor s'\theta \rfloor$ is used in the rewrite proof implies that $s\theta = t\theta$ and that the rewrite proof looks like this: $\lfloor t\theta \rfloor \to \lfloor s'\theta \rfloor \to^* u \;^*\!\leftarrow \lfloor t'\theta \rfloor$. Hence, we have that $R_{\lfloor C\theta \rfloor} \models \lfloor s'\theta \approx t'\theta \rfloor$. Now consider the follwoing EQFACT inference from $C\theta$:

$$\frac{C''\theta \vee t\theta \approx t'\theta \vee s\theta \approx s'\theta}{C''\theta \vee t'\theta \not\approx s'\theta \vee t\theta \approx t'\theta} \text{ EQFACT}$$

by Lemma 8 the conclusion of the inference is entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ smaller than $\lfloor C\theta \rfloor$ and therefore by part (iii) of the induction hypothesis true in $R_{\lfloor C\theta \rfloor}$. Since $\lfloor t'\theta \not\approx s'\theta \rfloor$ is false in $R_{\lfloor C\theta \rfloor}$, $\lfloor t\theta \approx t'\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$ and hence $\lfloor C\theta \rfloor$ is true in $R_{\lfloor C\theta \rfloor}$ contradicting our assumption.

7 Construction of Higher-order Model

Lemma 14. *For terms, t, t' and u, if $t \sim t'$ then $tu \sim t'u$*

Proof. The proof proceeds by induction on the number of rewrite steps between $\lfloor t \rfloor$ and $\lfloor t' \rfloor$. If $t = t'$ then the Lemma follows trivially. Let the number of rewrite steps between $\lfloor t \rfloor$ and $\lfloor t' \rfloor$ be n. Let t'' be the term such that $\lfloor t' \rfloor$ rewrites to $\lfloor t'' \rfloor$ in $n - 1$ steps and $\lfloor t'' \rfloor$ rewrites to $\lfloor t \rfloor$ in a single step. By the induction hypothesis we have that $t'u \sim t''u$. Thus, if it can be shown that $t''u \sim tu$ the Lemma follows by the transitivity of $\sim$.

Since $\lfloor t'' \rfloor$ rewrites to $\lfloor t \rfloor$ in a single step, it must be the case that either $\lfloor t'' \rfloor \to \lfloor t \rfloor$ or $\lfloor t \rfloor \to \lfloor t'' \rfloor$ is a rule in R_∞. Without loss of generality, assume that it is the first. This rule must stem from a productive clause of the form $\lfloor C\theta \rfloor \approx \lfloor C'\theta \vee t_1\theta \approx t_2\theta \rfloor$ where $t_1\theta \approx t''$ and $t_2\theta \approx t$. By the definition of a productive clause $t'' \approx t$ is strictly eligible in $C\theta$ and likewise $t_1 \approx t_2$ is strictly eligble in C. Further, t'' and t are of functional type, so t_1 and t_2 must be of functional or polymorphic type. Thus, there is an ARGCONG inference from C with conclusion $(C' \vee t_1x \approx t_2x)\sigma$. Call this conclusion E.

By part (ii) of Lemma 13, $C\theta$ is not redundant and therefore C is not redundant. Thus E is either in N or $Red(N)$. The ground instance of $\lfloor E \rfloor$, $\lfloor C\theta \vee t''u \approx tu \rfloor$ is thus either in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$ or entailed by clauses in $\lfloor \mathcal{G}_\Sigma(N) \rfloor$. Therefore it is true in R_∞. By Lemma 4, we have that $\lfloor C'\theta \rfloor$ is false in R_∞ which implies that $\lfloor t''u \approx tu \rfloor$ must be true.

Lemma 15. *For all terms t, t', u, u', if $t \sim t'$ and $u \sim u'$, then $tu \sim t'u'$ if neither t nor t' is of the form $\mathcal{C}\ \overline{s_{n>1}}$, $\mathsf{K}\ \overline{t_{n>0}}$ or $\mathsf{I}\ \overline{t_n}$.*

Proof. By Lemma 14, we have that $t'u' \sim tu'$, so if it can be shown that $tu' \sim tu$ the Lemma follows immediately by the transitivity of $\sim$. By the condition on the form of t and t', we have that $t = \mathsf{f}\ \overline{s_n}$. Therefore $\lfloor t \rfloor = \mathsf{f_n}(\overline{s_n})$ and $\lfloor tu' \rfloor = \mathsf{f_{n+1}}(\overline{s_n}, u')$. It is obvious that any rewrite steps from $\lfloor u' \rfloor$ can be carried out from $\lfloor tu' \rfloor$ and therefore $tu' \sim tu$.

Lemma 16. *For all terms u and u' such that $u \sim u'$, $\mathcal{C}\ t_1\ t_2\ u \sim \mathcal{C}\ t_1\ t_2\ u'$ and $\mathsf{K}\ t\ u \sim \mathsf{K}\ t\ u'$ and $\mathsf{I}\ u \sim \mathsf{I}\ u'$.*

Proof. By multiple applications of Lemma 14, we have that $\lfloor u\ \overline{t} \rfloor \approx \lfloor u'\ \overline{t} \rfloor$ holds in R_∞ for all type correct tuple of terms $\overline{t}$. We also have that every member of $\lfloor \mathcal{G}_\Sigma(ECA) \rfloor$ holds in R_∞. Thus, the lemma follows by an appeal to Lemma 9 with $R = R_\infty$.

Theorem 1. *For all terms t, t', u and u', if $t \sim t'$ and $u \sim u'$, then $tu \sim t'u'$.*

Proof. Proof is by induction on $\|t\| + \|u\| + \|t'\| + \|u'\|$. the base case splits into two cases.

Case 1: Neither t nor t' is of the form $\mathcal{C}\ t_1\ t_2$, $\mathsf{K}\ t$ or I. Then the proof follows by an application of Lemma 14.

> and parts (i) and (iv)?

Case 2: One or both of t and t' are of the form $\mathcal{C}\, t_1\, t_2$, $\mathsf{K}\, t$ or I. Without loss of generality, assume that t is of the form $\mathcal{C}\, t_1\, t_2$. By Lemma 15, $t'u' \sim tu'$. Thus, if it can be proven that $tu' = \mathcal{C}\, t_1\, t_2\, u' \sim \mathcal{C}\, t_1\, t_2\, u = tu$, the theorem follows by the transitivity of $\sim$. By Lemma 16, $\mathcal{C}\, t_1\, t_2\, u' \sim \mathcal{C}\, t_1\, t_2\, u$ completing the base case.

For the inductive case, one or more of $\|t\|, \|u\|, \|t'\|$ or $\|u'\|$ is greater than 0. We show that the theorem holds for the first two cases. The latter two can be proved in a like manner.

Case 1: $\|t\| > 0$. Let $t'' = (t)\downarrow^w$. Since R_∞ is a model of $\lfloor \mathcal{G}_\Sigma(N) \rfloor$, the floors of the ground instances of all combinator and extended combinator axioms must be true in R_∞. Thus, $t \sim t''$ and by Lemma 14, $tu \sim t''u$. Since $\|t''\| < \|t\|$, the induction hypothesis can be used to conclude that $t''u \sim t'u'$. By the transitivity of $\sim$, $tu \sim t'u'$ follows.

Case 2: $\|u\| > 0$ Since R_∞ models all combinator and extended combinator axioms, we have $u \sim u_2$ where $u_2 = (u)\downarrow^w$. By the induction hypothesis, we have $tu_2 \sim t'u'$. Either Lemma 15 or Lemma 16 is applicable to prove $tu \sim tu_2$. The theorem follows by the transitivity of $\sim$.

Definition 3. *Define an interpretation $R_\infty^\uparrow = (\mathcal{U}^\uparrow, \mathcal{E}^\uparrow, \mathcal{J}^\uparrow)$ in the ceiling logic as follows. Let $(\mathcal{U}, \mathcal{E}, \mathcal{J}) = R_\infty$. Let $\mathcal{U}_\tau^\uparrow = \mathcal{U}_{\lfloor \tau \rfloor}$ and $\mathcal{J}^\uparrow(\mathsf{f}, \overline{\tau}) = \mathcal{J}(\mathsf{f}_0^{\overline{\tau}}, \lfloor \overline{\tau} \rfloor)$. Since R_∞ is term-generated, for every $a \in \mathcal{U}_{\lfloor \tau \rightarrow \upsilon \rfloor}$ and $b \in \mathcal{U}_{\lfloor \tau \rfloor}$, there exists ground terms $s : \tau \rightarrow \upsilon$ and $t : \tau$ such that $[\![\lfloor s \rfloor]\!]_{R_\infty}^\xi = a$ and $[\![\lfloor t \rfloor]\!]_{R_\infty}^\xi = b$.*
Define $\mathcal{E}^\uparrow$ by

$$\mathcal{E}_{\tau,\upsilon}^\uparrow(a)(b) = [\![\lfloor st \rfloor]\!]_{R_\infty}^\xi$$

This interpretation is well defined if the definition of $\mathcal{E}^\uparrow$ does not depend on the choice of the ground terms s and t. To show this, we assume that there exists other ground terms s' and t' such that $[\![\lfloor s' \rfloor]\!]_{R_\infty}^\xi = a$ and $[\![\lfloor t' \rfloor]\!]_{R_\infty}^\xi = b$. By Lemma 1, it follows from $[\![\lfloor s \rfloor]\!]_{R_\infty}^\xi = [\![\lfloor s' \rfloor]\!]_{R_\infty}^\xi$ and $[\![\lfloor t \rfloor]\!]_{R_\infty}^\xi = [\![\lfloor t' \rfloor]\!]_{R_\infty}^\xi$ that

$$[\![\lfloor st \rfloor]\!]_{R_\infty}^\xi = [\![\lfloor s't' \rfloor]\!]_{R_\infty}^\xi$$

indicating that the definition of $\mathcal{E}^\uparrow$ is independent of the choice of s and t.

Since R_∞ is a term-generated model of $\lfloor \mathcal{G}_\Sigma(N) \rfloor$, we can show that $R_\infty^\uparrow$ is also term-generated.

Lemma 17 (Model transfer to ceiling logic). $R_\infty^\uparrow$ *is a term-generated model of* $\mathcal{G}_\Sigma(N)$.

Proof. By induction on ground terms t of the ceiling logic it is shown that $[\![t]\!]_{R_\infty^\uparrow}^\xi = [\![\lfloor t \rfloor]\!]_{R_\infty}^\xi$. Let t be a ground ceiling term. If t is of the form $\mathsf{f}\langle \overline{\tau} \rangle$, then $[\![t]\!]_{R_\infty^\uparrow}^\xi = \mathcal{J}^\uparrow(\mathsf{f}, [\![\overline{\tau}]\!]^\xi) = \mathcal{J}^\uparrow(\mathsf{f}, \overline{\tau}) = \mathcal{J}(\mathsf{f}_0, \lfloor \overline{\tau} \rfloor) = \mathcal{J}(\mathsf{f}_0, [\![\lfloor \overline{\tau} \rfloor]\!]^\xi) = [\![\lfloor t \rfloor]\!]_{R_\infty}^\xi$
If t is an application $t = t_1 t_2$, where t_1 is of type $\tau \rightarrow \upsilon$, then we have:

$$\begin{aligned}
[\![t_1\, t_2]\!]_{R_\infty^\uparrow}^\xi &= \mathcal{E}_{\tau,\upsilon}^\uparrow([\![t_1]\!]_{R_\infty^\uparrow}^\xi)([\![t_2]\!]_{R_\infty^\uparrow}^\xi) \\
&\overset{\text{IH}}{=} \mathcal{E}_{\tau,\upsilon}^\uparrow([\![\lfloor t_1 \rfloor]\!]_{R_\infty}^\xi)([\![\lfloor t_2 \rfloor]\!]_{R_\infty}^\xi) \\
&\overset{\text{Def}\,\mathcal{E}^\uparrow}{=} [\![\lfloor t_1\, t_2 \rfloor]\!]_{R_\infty}^\xi
\end{aligned}$$

We have shown that $[\![t]\!]^{\xi}_{R^{\uparrow}_{\infty}} = [\![\lfloor t \rfloor]\!]^{\xi}_{R_{\infty}}$ for all ground ceiling logic terms t. It follows that a ground equation $s \approx t$ or inequality $s \not\approx t$ is true in $R^{\uparrow}_{\infty}$ if and only if $\lfloor s \approx t \rfloor$ or $\lfloor s \not\approx t \rfloor$ is true in R_{∞}. Hence, a ground clause C is true in $R^{\uparrow}_{\infty}$ if and only if $\lfloor C \rfloor$ is true in R_{∞}.

By Lemma 13, R_{∞} is a model of $\lfloor \mathcal{G}_{\Sigma}(N) \rfloor$, i.e., all clauses $\lfloor C \rfloor \in \lfloor \mathcal{G}_{\Sigma}(N) \rfloor$ are true in R_{∞}. Hence, all clauses $C \in \mathcal{G}_{\Sigma}(N)$ are true in $R^{\uparrow}_{\infty}$ and therefore $R^{\uparrow}_{\infty}$ is a model of $\mathcal{G}_{\Sigma}(N)$.

It remains to show that $R^{\uparrow}_{\infty}$ is term-generated. Let $a \in \mathcal{U}^{\uparrow}_{\tau}$. Since $\mathcal{U}^{\uparrow}_{\tau} = \mathcal{U}_{\lfloor \tau \rfloor}$ and R_{∞} is term-generated, we have a ground term t of the floor logic with $[\![t]\!]^{\xi}_{R_{\infty}} = a$. Using what we showed above, we have $[\![\lfloor t \rfloor]\!]^{\xi}_{R^{\uparrow}_{\infty}} = [\![t]\!]^{\xi}_{R_{\infty}} = a$. Hence, $R^{\uparrow}_{\infty}$ is term-generated.

Lemma 18 (Model Transfer). $R^{\uparrow}_{\infty}$ *is a model of* N.

Proof. To be completed

Lemma 19 (Extensional Model). *If the extensionality axiom is present in* N*, then* $R^{\uparrow}_{\infty}$ *is an extensional model.*

Proof. To be completed

8 Removing Combinator Axioms

Next, we show that it is possible to replace the combinator axioms with a dedicated inference rule. We name the inference NARROW. Unlike the other infernce rules, it works at prefix positions. We define *nearly first-order* positions inductively. For any term t, either $t = \zeta \overline{t_n}$ or $t = \tilde{\mathcal{C}} \overline{t_n}$. In the first case, the nearly first-order subterms of t are $\zeta \overline{t_i}$ for $0 \leq i \leq n$ and all the nearly first-order subterms of the t_i. In the second case, the nearly first-order subterms are $\tilde{\mathcal{C}} \overline{t_i}$ for $0 \leq i \leq n$. The notation $s\langle\!\langle u \rangle\!\rangle$ is to be read as u is a nearly first-order subterm of s. The NARROW inference:

$$\frac{C' \vee [\neg] s\langle\!\langle u \rangle\!\rangle \approx s'}{(C' \vee [\neg] s\langle\!\langle r \rangle\!\rangle \approx s')\sigma} \text{ NARROW}$$

with the following side conditions:

1. $u \notin \mathcal{V}$
2. Let $l \approx r$ be a combinator axiom. $\sigma = mgu(l, u)$;
3. $s\langle\!\langle u \rangle\!\rangle \sigma \not\lesssim s'\sigma$;
4. $([\neg] s\langle\!\langle u \rangle\!\rangle \approx s')\sigma$ is eligible in $C\sigma$, and strictly eligible if it is positive.

We show that any inference that can be carried out using an extended combinator axiom can be simulated with NARROW proving completeness. It is obvious that an EQRES or EQFACT inference cannot have an extended combinator axiom as its premise. By the SUBVARSUP side conditions, an extended combinator axiom cannot be either of its premises. Thus we only need to show that SUP inferences with extended combinator axioms can be simulated. Note that an extended axiom can only be the left premise of a SUP inference. Consider the following inference:

$$\frac{l \approx r \qquad C' \vee [\neg]s\langle u\rangle \approx s'}{(C' \vee [\neg]s\langle r\rangle \approx s')\sigma} \text{ SUP}$$

Let $l = \mathsf{S}\,\overline{x_{n>3}}$ and $u = \delta\,\overline{t_m}$. By the variable condition, we must have that m
Then we have that $r\sigma =$

References

1. Heiko Becker, Jasmin Christian Blanchette, Uwe Waldmann, and Daniel Wand. A transfinite knuth–bendix order for lambda-free higher-order terms. In Leonardo de Moura, editor, *Automated Deduction – CADE 26*, pages 432–453, Cham, 2017. Springer International Publishing.
2. Alexander Bentkamp, Jasmin Christian Blanchette, Simon Cruanes, and Uwe Waldmann. Superposition for lambda-free higher-order logic. In Didier Galmiche, Stephan Schulz, and Roberto Sebastiani, editors, *Automated Reasoning*, pages 28–46, Cham, 2018. Springer International Publishing.
3. Alexander Bentkamp, Jasmin Christian Blanchette, Sophie Tourret, Petar Vukmirović, and Uwe Waldmann. Superposition with lambdas. *submitted for publication*, 2019.
4. Uwe Waldmann. Automated reasoning II. Lecture notes, Max-Planck-Institut für Informatik, 2016. `http://resources.mpi-inf.mpg.de/departments/rg1/teaching/autrea2-ss16/script-current.pdf`.