Termination Methods

Higher-Order Rewriting

Simple Types

- Base types B (e.g. Bool, Nat)
- Arrow types [e.g. Nat $\to$ (Nat $\to$ Bool)]
- each constant/variable has a type
- Type($\lambda x\colon \sigma.s\colon \tau$) $= \sigma \to \tau$
- Type($s\colon \sigma \to \tau\, t\colon \sigma$) $= \tau$

Typing Rules

$$\frac{}{x : A \vdash x : A}\,(Id)$$

$$\frac{\Gamma, x : A \vdash u : B}{\Gamma \vdash \lambda x.u : A \to B}\,(\to I)$$

$$\frac{\Gamma \vdash s : A \to B \quad \Delta \vdash t : A}{\Gamma, \Delta \vdash st : B}\,(\to E)$$

Untyped Beta Rule does not Terminate

Untyped β -rule

$$(\lambda x.s)\, t \rightsquigarrow s\,[x \mapsto t]$$

- Recall: Untyped β-rule does not terminate.

$$\begin{aligned}(\lambda x.xx)\,(\lambda x.xx) &\rightsquigarrow (xx)\,[x \mapsto \lambda x.xx] = \\ &= (x\,[x \mapsto \lambda x.xx])\,(x\,[x \mapsto \lambda x.xx]) \\ &= (\lambda x.xx)\,(\lambda x.xx)\end{aligned}$$

Untyped Beta Rule does not Terminate

Untyped β -rule

$$(\lambda x.s)\, t \rightsquigarrow s\,[x \mapsto t]$$

- Recall: Untyped β-rule does not terminate.

$$\begin{aligned}(\lambda x.xx)\,(\lambda x.xx) &\rightsquigarrow (xx)\,[x \mapsto \lambda x.xx] = \\ &= (x\,[x \mapsto \lambda x.xx])\,(x\,[x \mapsto \lambda x.xx]) \\ &= (\lambda x.xx)\,(\lambda x.xx)\end{aligned}$$

Typed Beta Mortality

β -rule

$$(\lambda x\colon \sigma.s\colon \tau)\, t\colon \sigma \rightsquigarrow s\,[x\colon \sigma \mapsto t\colon \sigma]\colon \tau$$

$$\lambda x\colon \sigma \to \tau.\,(x\colon \sigma \to \tau x\colon \sigma)\colon (\sigma \to \tau) \to \tau$$

Typed Beta Mortality

β -rule

$$(\lambda x\colon \sigma.s\colon \tau)\, t\colon \sigma \rightsquigarrow s\,[x\colon \sigma \mapsto t\colon \sigma]\colon \tau$$

$$\lambda x\colon \sigma \to \tau.\,(x\colon \sigma \to \tau x\colon \sigma)\colon (\sigma \to \tau) \to \tau$$

Termination

- Turing gave first proof
- Tait's proof [1967] :
  - Induction on term structure
  - Induction on type structure

Termination of β-reduction Alone?

- In the simply-typed λ-calculus $\rightsquigarrow_\beta$ can be proved terminating by a direct induction on the type of the substituted variable [Sanchis 1967, van Daalen 1980].
- This does not extend to rewriting where the type of substituted variables can increase
  - e.g. $f\,(cx) \to x$ where $x : A \Rightarrow B$.

Proving Termination

Computability has been introduced for proving termination of β-reduction in typed λ-calculi [Tait 1967, Girard 1970]

- every type T is mapped to a set $[[T]]$ of computable terms
- every term $t : T$ is proved to be computable, i.e., $t \in [[T]]$

Predicates

- $S[t]$: t is terminating
- $C[t]$: t is computable

Computability

Definition (Computability predicate)

- Basic t: $C[t]$ if $S[t]$
- Arrow t: $C[t]$ if $C[t(s)]$ for all computable s (of the right type)

Lemmas

0. Reducts of computable terms are computable
1. Computable terms are terminating
2. Applications are computable if all reducts are
3. Applications are computable if all the inputs are terminating

Main. Computable substitutions yield computable terms

Lemma 0

Lemma (0)

Reducts of computable terms are computable.

$$C[t] \wedge t \rightsquigarrow u \Rightarrow C[u]$$

Proof.

- Induction on type
- Basic t: $C[u]$ if $S[u]$ if $S[t]$ if $C[t]$
- Arrow $t : \sigma \to \tau$:
  - By def, $C\,[t\,(s : \tau)]$ for all computable s.
  - By ind, $C\,[u\,(s : \tau)]$ for all s.
  - By def, $C[u]$.

Lemma 0

Lemma (0)

Reducts of computable terms are computable.

$$C[t] \wedge t \rightsquigarrow u \Rightarrow C[u]$$

Proof.

- Induction on type
- Basic t: $C[u]$ if $S[u]$ if $S[t]$ if $C[t]$
- Arrow $t : \sigma \to \tau$:
  - By def, $C\,[t\,(s : \tau)]$ for all computable s.
  - By ind, $C\,[u\,(s : \tau)]$ for all s.
  - By def, $C[u]$.

Lemma 1

Lemma (1)

Computable terms are terminating.

$$C[t] \Rightarrow S[t]$$

Proof.

- Induction on type
- Basic t: By definition.
- Arrow $t : \sigma \to \tau$:
  - By def, $C[t(s)]$ for all computable $s : \sigma$.
  - By ind, $S[t(s) : \tau]$.
  - It must be that $S[t]$, too.

□

Lemma 1

Lemma (1)

Computable terms are terminating.

$$C[t] \Rightarrow S[t]$$

Proof.

- Induction on type
- Basic t: By definition.
- Arrow $t : \sigma \to \tau$:
  - By def, $C\,[t\,(s)]$ for all computable $s : \sigma$.
  - By ind, $S\,[t\,(s) : \tau]$.
  - It must be that $S\,[t]$, too.

Neutrality

- applying creates no new redexes
- t neutral: redexes of $t(s)$ are in t or s
- computable if reducts are
- $C[t]$ if $C[r]$ $\text{for all } r \text{ s.t.} t \rightsquigarrow r$

Lemma 2

Lemma (2)

Applications are neutral.

$$C[s(t)] \text{ if } \forall r\,(s(t) \rightsquigarrow r \Rightarrow C[r])$$

Proof.

- Induction on type of $s(t)$
- Basic: $S[s(t)]$ iff $\forall r.S[r]$
- Arrow:
  - Show $C[s(t)(u)]$ for each computable u.
  - By ind, $\forall r.C[r(u)]$ suffices, which is just $C[r]$.

Lemma 2

Lemma (2)

Applications are neutral.

$$C\left[s\left(t\right)\right] \textit{ if } \forall r\left(s\left(t\right) \rightsquigarrow r \Rightarrow C\left[r\right]\right)$$

Proof.

- Induction on type of $s\left(t\right)$
- Basic: $S\left[s\left(t\right)\right]$ iff $\forall r.S\left[r\right]$
- Arrow:
  - Show $C\left[s\left(t\right)\left(u\right)\right]$ for each computable u.
  - By ind, $\forall r.C\left[r\left(u\right)\right]$ suffices, which is just $C\left[r\right]$.

Corollary

Corollary

$C[(\lambda x.s)(t)]$ *if* $C[s\{x \mapsto t\}] \wedge C[t]$

Proof.

- Well-founded induction on s.t
- By L0, $S[s] \wedge S[t]$
- Let $s \rightsquigarrow s'$ and $t \rightsquigarrow t'$
- $C[s'\{x \mapsto t\}] \wedge C[t] \Rightarrow C[(\lambda x.s')(t)]$
- $C[s\{x \mapsto t\}] \wedge C[t'] \Rightarrow C[(\lambda x.s)(t')]$
- By L2, $C[(\lambda x.s)(t)]$ if $C[(\lambda x.s')(t)] \wedge C[(\lambda x.s)(t')] \wedge C[s\{x \mapsto t\}]$
- But $C[t] \Rightarrow C[t']$ and $C[s\{x \mapsto t\}] \Rightarrow C[s'\{x \mapsto t\}]$

Corollary

Corollary

$C[(\lambda x.s)(t)]$ *if* $C[s\{x \mapsto t\}] \wedge C[t]$

Proof.

- Well-founded induction on s.t
- By L0, $S[s] \wedge S[t]$
- Let $s \rightsquigarrow s'$ and $t \rightsquigarrow t'$
- $C[s'\{x \mapsto t\}] \wedge C[t] \Rightarrow C[(\lambda x.s')(t)]$
- $C[s\{x \mapsto t\}] \wedge C[t'] \Rightarrow C[(\lambda x.s)(t')]$
- By L2, $C[(\lambda x.s)(t)]$ if
  $C[(\lambda x.s')(t)] \wedge C[(\lambda x.s)(t')] \wedge C[s\{x \mapsto t\}]$
- But $C[t] \Rightarrow C[t']$ and $C[s\{x \mapsto t\}] \Rightarrow C[s'\{x \mapsto t\}]$

Lemma 3

Lemma (3)

Applications are computable if all the inputs are terminating.

$$S[t_1] \wedge ... \wedge S[t_n] \Rightarrow C[x(t_1)(t_2)...(t_n)]$$

Proof.

- Induction on type of $t = x(t_1)(t_2)...(t_n)$
- Basic t: Since only reducible inside terminating t_i, $S[t]$. By def, $C[t]$.
- Arrow $t : \sigma \to \tau$:
  - By L1, for any computable, $S[s]$.
  - By ind, $C[t(s) : \tau]$.
  - By def, $C[t]$.

Lemma 3

Lemma (3)

Applications are computable if all the inputs are terminating.

$$S[t_1] \wedge \ldots \wedge S[t_n] \Rightarrow C[x(t_1)(t_2)\ldots(t_n)]$$

Proof.

- Induction on type of $t = x(t_1)(t_2)\ldots(t_n)$
- Basic t: Since only reducible inside terminating t_i, $S[t]$. By def, $C[t]$.
- Arrow $t : \sigma \to \tau$:
  - By L1, for any computable, $S[s]$.
  - By ind, $C[t(s) : \tau]$.
  - By def, $C[t]$.

Main Lemma

Lemma (Main Lemma)

Computable substitutions yield computable terms.
$C[u\sigma]$ for all u and computable σ, where $C[\sigma]$ if $C[t]$ $\forall$ $x \rightsquigarrow t$ in σ

Proof.

- Structural induction on u
- u constant: $u = u\sigma$ is basic and terminating; so $C[u]$ by def.
- u variable x: If $x\sigma = x$, L3 applies; otherwise $C[x\sigma]$.
- $u = t(s)$: $u\sigma = t\sigma(s\sigma)$. By ind, $C[t\sigma]$; by def, $C[t\sigma(s\sigma)]$, since $C[s\sigma]$ by ind.
- $u = \lambda x.s$: Let $\sigma' = \sigma - \{x \mapsto x\sigma\} \cup \{x \mapsto t\}$ for computable t. By ind, $C[s\sigma']$. By L2C, $C[((\lambda x.s)\,\sigma)\,(t)]$, as $(\lambda x.s)\,\sigma = \lambda x.s\,(\sigma - \{x \mapsto x\sigma\})$ and $s\,(\sigma - \{x \mapsto x\sigma\})\,\{x \mapsto t\} = s\sigma'$. By def, $C[(\lambda x.s)\,\sigma]$.

Main Lemma

Lemma (Main Lemma)

Computable substitutions yield computable terms.
$C[u\sigma]$ for all u and computable σ, where $C[\sigma]$ if $C[t]\ \forall\ x \rightsquigarrow t$ in σ

Proof.

- Structural induction on u
- u constant: $u = u\sigma$ is basic and terminating; so $C[u]$ by def.
- u variable x: If $x\sigma = x$, L3 applies; otherwise $C[x\sigma]$.
- $u = t(s)$: $u\sigma = t\sigma(s\sigma)$. By ind, $C[t\sigma]$; by def, $C[t\sigma(s\sigma)]$, since $C[s\sigma]$ by ind.
- $u = \lambda x.s$: Let $\sigma' = \sigma - \{x \mapsto x\sigma\} \cup \{x \mapsto t\}$ for computable t. By ind, $C[s\sigma']$. By L2C, $C[((\lambda x.s)\,\sigma)\,(t)]$, as $(\lambda x.s)\,\sigma = \lambda x.s\,(\sigma - \{x \mapsto x\sigma\})$ and $s\,(\sigma - \{x \mapsto x\sigma\})\,\{x \mapsto t\} = s\sigma'$. By def, $C[(\lambda x.s)\,\sigma]$.

Termination Proof

Theorem (Termination)

All typed terms are terminating.

Proof of the Theorem.

- $C[t]$ for all t
  - Main lemma (empty substitution)
- $S[t]$ for all t
  - By Lemma 1

□

Termination Proof

Theorem (Termination)

All typed terms are terminating.

Proof of the Theorem.

- $C[t]$ for all t
  - Main lemma (empty substitution)
- $S[t]$ for all t
  - By Lemma 1

□

Higher-Order Rewriting - Examples

Functional

$$D\left(\lambda x.y\right) \rightsquigarrow \lambda x.0$$
$$D\left(\lambda x.x\right) \rightsquigarrow \lambda x.1$$
$$D\left(\lambda x.sin\left(F\left(x\right)\right)\right) \rightsquigarrow \lambda x.D\left(F\left(x\right)\right) \cdot cos\left(F\left(x\right)\right)$$

Higher-Order Rewriting - Examples

Gödel's System T

$$\begin{aligned}
\mathbb{N}, \alpha &: * \\
0, x &: \mathbb{N} \\
s &: \mathbb{N} \Rightarrow \mathbb{N} \\
\mathit{rec} &: \mathbb{N} \times \alpha \times (\mathbb{N} \to \alpha \to \alpha) \Rightarrow \alpha \\
u &: \alpha \\
F &: \mathbb{N} \to \alpha \to \alpha
\end{aligned}$$

$$\begin{aligned}
\mathit{rec}\,(0, u, F) &\rightsquigarrow u \\
\mathit{rec}\,(s\,(x)\,, u, F) &\rightsquigarrow F\,(x, \mathit{rec}\,(x, u, F))
\end{aligned}$$

HORPO: Higher-Order Recursive Path Ordering

- Jouannaud and Rubio generalization of RPO to the higher-order case.
- HORPO can be used for proving termination of higher-order term rewriting systems by proving well-foundedness of the union of HORPO and β-reduction of simply typed lambda calculus.
- multisets are used for comparison of arguments in the HORPO ordering.

Higher-Order Recursive Path Ordering – Ingredients

- precedence $\geq_F$
  - @ is minimal
  - assume total (for simplicity)
- type quasi-ordering $\geq_T$
  - well-founded
  - Arrow preservation:
    $\tau \to \sigma = \alpha$ iff $\alpha = \tau' \to \sigma'$, $\tau = \tau'$ and $\sigma = \sigma'$.
  - Arrow decreasingness:
    $\tau \to \sigma >_T \alpha$ implies $\sigma \geq_T \alpha$ or $\alpha = \tau' \to \sigma'$, $\tau = \tau'$ and $\sigma >_T \sigma'$.
  - Arrow monotonicity:
    $\tau \geq_T \sigma$ implies $\alpha \to \tau \geq_T \alpha \to \sigma$ and $\tau \to \alpha \geq_T \sigma \to \alpha$.
- status $stat_f \in \{Mul, Lex\}$ for every symbol f in the signature.

HOPRO – Definition

Definition (HORPO)

$s : \sigma \succ t : \tau$ iff $\sigma \geq_{\mathcal{T}} \tau$ and either of

1. $s = f(\bar{s})$ and either of
   1. $s_i \succeq t$ for some $s_i \in \bar{s}$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \succ_{stat_f} \bar{t}$ and $s \succ \bar{t}$
2. $s = @(\lambda x.u, w)$ and $u\{x \mapsto w\} \succ t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $u\{x \mapsto y\} \succ t$, for some fresh $y : \alpha$
   2. $t = \lambda y : \alpha.v$, $y \notin Var(v)$ and $u \succ v$
   3. $u = @(v, x)$, $x \notin Var(v)$and $v \succ t$

HOPRO – Definition

Definition (HORPO)

$s : \sigma \succ t : \tau$ iff $\sigma \geq_{\mathcal{T}} \tau$ and either of

1. $s = f(\bar{s})$ and either of
   1. $s_i \succeq t$ for some $s_i \in \bar{s}$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \succ_{stat_f} \bar{t}$ and $s \succ \bar{t}$
2. $s = @(\lambda x.u, w)$ and $u\{x \mapsto w\} \succ t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $u\{x \mapsto y\} \succ t$, for some fresh $y : \alpha$
   2. $t = \lambda y : \alpha.v$, $y \notin Var(v)$ and $u \succ v$
   3. $u = @(v, x)$, $x \notin Var(v)$and $v \succ t$

HOPRO – Definition

Definition (HORPO)

$s : \sigma \succ t : \tau$ iff $\sigma \geq_{\mathcal{T}} \tau$ and either of

1. $s = f(\bar{s})$ and either of
   1. $s_i \succeq t$ for some $s_i \in \bar{s}$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \succ_{stat_f} \bar{t}$ and $s \succ \bar{t}$
2. $s = @(\lambda x.u, w)$ and $u\{x \mapsto w\} \succ t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $u\{x \mapsto y\} \succ t$, for some fresh $y : \alpha$
   2. $t = \lambda y : \alpha.v$, $y \notin Var(v)$ and $u \succ v$
   3. $u = @(v, x)$, $x \notin Var(v)$and $v \succ t$

HOPRO – Definition

Definition (HORPO)

$s : \sigma \succ t : \tau$ iff $\sigma \geq_{\mathcal{T}} \tau$ and either of

1. $s = f(\bar{s})$ and either of
   1. $s_i \succeq t$ for some $s_i \in \bar{s}$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \succ_{stat_f} \bar{t}$ and $s \succ \bar{t}$
2. $s = @(\lambda x.u, w)$ and $u\{x \mapsto w\} \succ t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $u\{x \mapsto y\} \succ t$, for some fresh $y : \alpha$
   2. $t = \lambda y : \alpha.v$, $y \notin Var(v)$ and $u \succ v$
   3. $u = @(v, x)$, $x \notin Var(v)$and $v \succ t$

Example: Gödel's System T

Gödel's system T

$$rec\,(0, u, F) \rightsquigarrow u$$
$$rec\,(s\,(x)\,, u, F) \rightsquigarrow F\,(x, rec\,(x, u, F))$$

Assume $rec \in Mul$.

- The first rule: immediately by case 1.1.
- The second rule: apply case 1.2, and it remains to show:
  - $F \succeq F$ – trivial
  - $s\,(x) \succeq x$ – by case 1.1
  - $rec\,(s\,(x)\,, u, F) \succ rec\,(x, u, F)$ – by case 1.3, recursively calling $s\,(x) \succ x$

This proves Gödel's *polymorphic* recursor, i.e. the output type of *rec* is any given type.

HORPO – Not Strong Enough

Extended System T

$$x + 0 \rightsquigarrow 0$$
$$x + s(y) \rightsquigarrow s(x + y)$$
$$x * y \rightsquigarrow \mathit{rec}(y, 0, \lambda z_1 z_2 . x + z_2)$$

- The first two rules: immediately
- The third rule: use the precedence $* >_F \mathit{rec}$ to eliminate the *rec* operator.
  - *The computation fails!* there is no higher-order subterm of $x * y$ to handle $\lambda z . x + z$.
    - Requires a more subtle ordering.

Computable Closure

Definition

The computability closure $\mathcal{CC}(t = f(\bar{t}))$, is the set $\mathcal{CC}(t, \emptyset)$, s.t. $\mathcal{CC}(t, \mathcal{V})$, with $\mathcal{V} \cap Var(t) = \emptyset$, is the *smallest set* of typable terms containing all variables in $\mathcal{V}$ and terms in $\bar{t}$, *closed under*:

1. *abstraction*: $\lambda x.s \in \mathcal{CC}(t, \mathcal{V})$ if $x \notin Var(t) \cup \mathcal{V}$ and $s \in \mathcal{CC}(t, \mathcal{V} \cup \{x\})$
2. *application*: $@(s, u_1, ..., u_n) \in \mathcal{CC}(t, \mathcal{V})$ *if* $s : \sigma_1 \to ... \to \sigma_n \to \sigma \in \mathcal{CC}(t, \mathcal{V})$ *and* $u_i : \sigma_i \in \mathcal{CC}(t, \mathcal{V})$
3. *precedence*: $g(\bar{s}) \in \mathcal{CC}(t, \mathcal{V})$ if $f >_F g$, and $\bar{s} \in \mathcal{CC}(t, \mathcal{V})$
4. *recursive call*: $f(\bar{s}) \in \mathcal{CC}(t, \mathcal{V})$ if $f(\bar{s})$ is a well typed term s.t. terms in $\bar{s}$ belong to $\mathcal{CC}(t, \mathcal{V})$ and $\bar{t} \left(\rightsquigarrow_\beta \cup \rhd\right)_{stat_f} \bar{s}$
5. *reduction*: $v \in \mathcal{CC}(t, \mathcal{V})$ if $u \in \mathcal{CC}(t, \mathcal{V})$ and $u \rightsquigarrow_\beta v$

General Schema

Definition

a rewrite system R satisfies the *general schema* if

$$R = \left\{ f\left(\bar{l}\right) \rightsquigarrow r \mid r \in \mathcal{CC}(f(\bar{l})) \right\}$$

- computability with respect to the rewrite relation $\rightsquigarrow_R \cup \rightsquigarrow_\beta$
- a new case in Tait's Main Lemma, for terms headed by an algebraic function symbol.

Theorem (Blanqui, Jouannaud and Okada, 2001)

$\rightsquigarrow_R \cup \rightsquigarrow_\beta$ *is terminating.*

HORPO + Computational Closure – Ingredients

- A set of strictly positive inductive types inducing an accessibility relationship $\bar{s} \trianglerighteq_{acc} v$ s.t. $v \in \bar{u}$ or v is accessible from $u \in \bar{s}$.
- a precedence on function symbols
- a congruence on types
- $s \succ^{X} t$ for the main ordering, with X a finite set of variables (omitted when empty)
- $s : \alpha \succ^{X}_{T} t : \tau$ for $s \succ^{X} t$ and $\sigma =_{T} \tau$
- $l \succ^{\emptyset}_{T} r$ as initial call for each $l \rightsquigarrow r \in R$

HORPO + Computational Closure

Definition

$s \succ^X t$ iff $t \in X$ or

1. $s = f(\bar{s})$ and either of
   1. $u \succeq_T^X t$ for some u s.t. $\bar{s} \trianglerighteq_{acc} u$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ^X \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \left(\succ_T^X\right)_{stat_f} \bar{t}$ and $s \succ^X \bar{t}$
   4. $t = \lambda x.u$ with $x \notin X$ and $f(\bar{s}) \succ^{X \cup \{x\}} u$
2. $s = @(u, v)$ and either of
   1. $t = @(u', v')$ and $\{u, v\} (\succ_T)_{mul} \{u', v'\}$
   2. $u = \lambda x : \alpha.w$ and $w\{x \mapsto v\} \succ^X t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $t = \lambda y : \alpha.w$, $x \notin X$ and $u \succ^{X \cup \{x\}} w$
   2. $u = @(v, x)$, $x \notin Var(v)$ and $u \succ^X t$

HORPO + Computational Closure

Definition

$s \succ^X t$ iff $t \in X$ or

1. $s = f(\bar{s})$ and either of
   1. $u \succeq_T^X t$ for some u s.t. $\bar{s} \trianglerighteq_{acc} u$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ^X \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \left(\succ_T^X\right)_{stat_f} \bar{t}$ and $s \succ^X \bar{t}$
   4. $t = \lambda x.u$ with $x \notin X$ and $f(\bar{s}) \succ^{X \cup \{x\}} u$
2. $s = @(u, v)$ and either of
   1. $t = @(u', v')$ and $\{u, v\} (\succ_T)_{mul} \{u', v'\}$
   2. $u = \lambda x : \alpha.w$ and $w\{x \mapsto v\} \succ^X t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $t = \lambda y : \alpha.w$, $x \notin X$ and $u \succ^{X \cup \{x\}} w$
   2. $u = @(v, x)$, $x \notin Var(v)$ and $u \succ^X t$

HORPO + Computational Closure

Definition

$s \succ^X t$ iff $t \in X$ or

1. $s = f(\bar{s})$ and either of
   1. $u \succeq_T^X t$ for some u s.t. $\bar{s} \trianglerighteq_{acc} u$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ^X \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \left(\succ_T^X\right)_{stat_f} \bar{t}$ and $s \succ^X \bar{t}$
   4. $t = \lambda x.u$ with $x \notin X$ and $f(\bar{s}) \succ^{X \cup \{x\}} u$
2. $s = @(u, v)$ and either of
   1. $t = @(u', v')$ and $\{u, v\} (\succ_T)_{mul} \{u', v'\}$
   2. $u = \lambda x : \alpha.w$ and $w\{x \mapsto v\} \succ^X t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $t = \lambda y : \alpha.w$, $x \notin X$ and $u \succ^{X \cup \{x\}} w$
   2. $u = @(v, x)$, $x \notin Var(v)$ and $u \succ^X t$

HORPO + Computational Closure

Definition

$s \succ^X t$ iff $t \in X$ or

1. $s = f(\bar{s})$ and either of
   1. $u \succeq_T^X t$ for some u s.t. $\bar{s} \trianglerighteq_{acc} u$
   2. $t = g(\bar{t})$ with $f >_F g$ and $s \succ^X \bar{t}$
   3. $t = f(\bar{t})$ with $\bar{s} \left(\succ_T^X\right)_{stat_f} \bar{t}$ and $s \succ^X \bar{t}$
   4. $t = \lambda x.u$ with $x \notin X$ and $f(\bar{s}) \succ^{X \cup \{x\}} u$
2. $s = @(u, v)$ and either of
   1. $t = @(u', v')$ and $\{u, v\} (\succ_T)_{mul} \{u', v'\}$
   2. $u = \lambda x : \alpha.w$ and $w\{x \mapsto v\} \succ^X t$
3. $s = \lambda x : \alpha.u$ and either of
   1. $t = \lambda y : \alpha.w$, $x \notin X$ and $u \succ^{X \cup \{x\}} w$
   2. $u = @(v, x)$, $x \notin Var(v)$ and $u \succ^X t$

Example: Extended System T

Extended System T

$$x + 0 \rightsquigarrow 0$$
$$x + s(y) \rightsquigarrow s(x + y)$$
$$x * y \rightsquigarrow rec(y, 0, \lambda z_1 z_2 . x + z_2)$$

- The third rule can now be proved terminating.
  - by case 1.2
    - use the precedence $* >_F \{rec, +, 0\}$
    - by base case $x, z_2 \in \mathcal{CC}(x * y, \{z_1, z_2\})$
    - thus, $x + z_2 \in \mathcal{CC}(x * y, \{z_1, z_2\})$
    - by applying abstraction $\lambda z_1 z_2 . x + z_2 \in \mathcal{CC}(x * y)$

Example: Brouwer's Ordinals

Brouwer's Ordinals

$$
\begin{aligned}
0 &: \mathit{Ord} \\
S &: \mathit{Ord} \Rightarrow \mathit{Ord} \\
\mathit{lim} &: (\mathbb{N} \to \mathit{Ord}) \Rightarrow \mathit{Ord} \\
n &: \mathbb{N} \\
F &: \mathbb{N} \to \mathit{Ord} \\
\mathit{rec} &: \mathit{Ord} \times \alpha \times (\mathit{Ord} \to \alpha \to \alpha) \times ((\mathbb{N} \to \mathit{Ord}) \to (\mathbb{N} \to \alpha) \to \alpha) \\
&\quad \Rightarrow \alpha
\end{aligned}
$$

$$
\begin{aligned}
\mathit{rec}\,(0, u, x, w) &\rightsquigarrow u \\
\mathit{rec}\,(s\,(z), u, x, w) &\rightsquigarrow @\,(x, z, \mathit{rec}\,(z, u, x, w)) \\
\mathit{rec}\,(\mathit{lim}\,(F), u, x, w) &\rightsquigarrow @\,(w, F, \lambda n.\mathit{rec}\,(@\,(F, n), u, x, w))
\end{aligned}
$$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_{T}^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\{lim\left(F\right), u, x, w\} \left(\succ_{T}^{\{n\}}\right)_{mul} \{@\left(F, n\right), u, x, w\}$ (1.3) if
    - $lim\left(F\right) \succ^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} F$ if $lim\left(F\right) \succ^{\{n\}} F\left(n\right)$ and $lim\left(F\right) \succ^{\{n\}} n$ or $n \succ^{\{n\}} n$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{@\left(F, n\right), u, x, w\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{F, n\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F\left(n\right)$ and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ or $n \succ^{\{n\}} n$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right),u,x,w\right) \succ_T^{\emptyset} @\left(w,F,\lambda n.rec\left(@\left(F,n\right),u,x,w\right)\right)$ if
2. $rec\left(lim\left(F\right),u,x,w\right) \succ \left\{w,F,\lambda n.rec\left(@\left(F,n\right),u,x,w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right),u,x,w\right) \succ w,F$ (1.1) and

- $rec\left(lim\left(F\right),u,x,w\right) \succ \lambda n.rec\left(@\left(F,n\right),u,x,w\right)$ if
- $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} rec\left(@\left(F,n\right),u,x,w\right)$ (1.4) if
  - $\{lim\left(F\right),u,x,w\} \left(\succ_T^{\{n\}}\right)_{mul} \{@\left(F,n\right),u,x,w\}$ (1.3) if
    - $lim\left(F\right) \succ^{\{n\}} @\left(F,n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} F$ if $lim\left(F\right) \succ^{\{n\}} F\left(n\right)$ and $lim\left(F\right) \succ^{\{n\}} n$ or $n \succ^{\{n\}} n$
  - $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} \{@\left(F,n\right),u,x,w\}$ if
    - $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} u,x,w$ (1.2)
    - $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} @\left(F,n\right)$ if
    - $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} F\left(n\right)$ and
    - $rec\left(lim\left(F\right),u,x,w\right) \succ^{\{n\}} n$ or $n \succ^{\{n\}} n$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\{lim\left(F\right), u, x, w\} \left(\succ_T^{\{n\}}\right)_{mul} \{@\left(F, n\right), u, x, w\}$ (1.3) if
    - $lim\left(F\right) \succ^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ \{F, n\}$ if $lim\left(F\right) \succ^{\{n\}} F$ (1.2) and $lim\left(F\right) \succ^{\{n\}} n$ (1.5) $n \succ^{\{n\}} n$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{@\left(F, n\right), u, x, w\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{F, n\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.2) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ (1.5)

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_{\mathcal{T}}^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
- $\{lim\left(F\right), u, x, w\} \left(\succ_{\mathcal{T}}^{\{n\}}\right)_{mul} \{@\left(F, n\right), u, x, w\}$ (1.3) if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{@\left(F, n\right), u, x, w\}$ if

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$ if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\{lim\left(F\right), u, x, w\} \left(\succ_T^{\{n\}}\right)_{mul} \{@\left(F, n\right), u, x, w\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \{F, n\}$
      if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{@\left(F, n\right), u, x, w\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \{F, n\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$ if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$
      if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$
      if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$
      if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$

Termination for Brouwer's Ordinals

1. $rec\left(lim\left(F\right), u, x, w\right) \succ_T^{\emptyset} @\left(w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right)$ if
2. $rec\left(lim\left(F\right), u, x, w\right) \succ \left\{w, F, \lambda n.rec\left(@\left(F, n\right), u, x, w\right)\right\}$ (1.2) if
3. $rec\left(lim\left(F\right), u, x, w\right) \succ w, F$ (1.1) and

- $rec\left(lim\left(F\right), u, x, w\right) \succ \lambda n.rec\left(@\left(F, n\right), u, x, w\right)$ if
- $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} rec\left(@\left(F, n\right), u, x, w\right)$ (1.4) if
  - $\left\{lim\left(F\right), u, x, w\right\} \left(\succ_T^{\{n\}}\right)_{mul} \left\{@\left(F, n\right), u, x, w\right\}$ (1.3) if
    - $lim\left(F\right) \succ_T^{\{n\}} @\left(F, n\right)$ if
    - $lim\left(F\right) \succ^{\{n\}} \left\{F, n\right\}$
      if $lim\left(F\right) \succ^{\{n\}} F$ (1.1) and $lim\left(F\right) \succ^{\{n\}} n$ as $n \in \{n\}$
  - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{@\left(F, n\right), u, x, w\right\}$ if
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} u, x, w$ (1.1)
    - $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} @\left(F, n\right)$ if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} \left\{F, n\right\}$ (1.2) if
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} F$ (1.1) and
      $rec\left(lim\left(F\right), u, x, w\right) \succ^{\{n\}} n$ as $n \in \{n\}$