Normalising Strategies for Orthogonal Systems and Beyond

Vincent van Oostrom

Master Seminar 1, Wednesday October 18th, 2017

Problem

Setting

term rewrite systems representing arbitrary partial functions

- normal form represents result

Problem

Setting

term rewrite systems representing arbitrary partial functions

- normal form represents result
- at most one result

Problem

Setting

term rewrite systems representing arbitrary partial functions

- normal form represents result
- at most one result
- termination not decidable

Problem

Setting

term rewrite systems representing arbitrary partial functions

- normal form represents result
- at most one result
- termination not decidable

Example

- combinatory logic (Schönfinkel, Curry), λ-calculus (Church)
- PCF (Scott, Plotkin, Milner)
- Haskell

Problem

Setting

term rewrite systems representing arbitrary partial functions

- normal form represents result
- at most one result
- termination not decidable

Example

- combinatory logic (Schönfinkel, Curry), λ-calculus (Church)
- PCF (Scott, Plotkin, Milner)
- Haskell

Question

strategy that always computes result, if that exists?

Strategy example

term:

$$f(1)$$

Strategy example

term:

$$f(1)$$

rewrite rules:

$$\begin{aligned} f(x) &\rightarrow \text{if } x = 0 \text{ then } 1 \text{ else } 2 \cdot f(x-1) \\ \text{if false then } x \text{ else } y &\rightarrow y \\ \text{if true then } x \text{ else } y &\rightarrow x \\ &\vdots \end{aligned}$$

Strategy example

$$f(1)$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot f(1-1)$$
$$\downarrow$$
$$\text{if false then } 1 \text{ else } 2 \cdot f(1-1)$$
$$\downarrow$$
$$2 \cdot f(1-1)$$
$$\downarrow$$
$$2 \cdot f(0)$$
$$\downarrow$$
$$2 \cdot (\text{if } 0 = 0 \text{ then } 1 \text{ else } 2 \cdot f(0-1))$$
$$\downarrow$$
$$2 \cdot (\text{if true then } 1 \text{ else } 2 \cdot f(0-1))$$
$$\downarrow$$
$$2 \cdot 1$$
$$\downarrow$$
$$2$$

Strategy example

$$f(1)$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot f(1-1)$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot f(0)$$
$$\downarrow$$
$$\text{if false then } 1 \text{ else } 2 \cdot f(0)$$
$$\downarrow$$
$$2 \cdot f(0)$$
$$\downarrow$$
$$2 \cdot (\text{if } 0 = 0 \text{ then } 1 \text{ else } 2 \cdot f(0-1))$$
$$\downarrow$$
$$2 \cdot (\text{if true then } 1 \text{ else } 2 \cdot f(0-1))$$
$$\downarrow$$
$$2 \cdot 1$$
$$\downarrow$$
$$2$$

And Yet Another

$$f(1)$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot f(1-1)$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot (\text{if } (1-1) = 0 \text{ then } 1 \text{ else } 2 \cdot f((1-1)-1))$$
$$\downarrow$$
$$\text{if } 1 = 0 \text{ then } 1 \text{ else } 2 \cdot (\text{if } (1-1) = 0 \text{ then } 1 \text{ else } 2 \cdot (\text{if } ((1-1)-1) = 0 \text{ then } 1 \text{ else } 2 \cdot f(((1-1)-1)-1)))$$
$$\downarrow$$
$$\vdots$$

Strategy

Definition

strategy is sub-system having same set of normal forms

Strategy

Definition

strategy is sub-system having same set of normal forms

Example

innermost reduction is a strategy
non-innermost reduction is not a strategy; lone redex is stuck
call-by-value is not a strategy; $(\lambda x.x)(z\lambda x.x)$ is stuck

Strategy

Definition

strategy is sub-system having same set of normal forms

Example

innermost reduction is a strategy
non-innermost reduction is not a strategy; lone redex is stuck
call-by-value is not a strategy; $(\lambda x.x)(z\lambda x.x)$ is stuck

Definition

strategy is normalising if terminating on terms having normal form

Strategy

Definition

strategy is sub-system having same set of normal forms

Example

innermost reduction is a strategy
non-innermost reduction is not a strategy; lone redex is stuck
call-by-value is not a strategy; $(\lambda x.x)(z\lambda x.x)$ is stuck

Definition

strategy is normalising if terminating on terms having normal form

Example

normal order is normalising for left-normal systems (λ-calculus/CL)
innermost reduction is not normalising; $KI\Omega$

Normalisation by Random Descent

VvO and Toyama, FSCD 2016
hyper-normalisation results for left-normal (CL, λ) systems:

Future work

- Retrofit known (hyper-)normalisation results in setting (may require slight generalisation of conversion monoid)
- Extend to other sets of normal forms (head, weak head)

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

Example

$$
\begin{aligned}
@(I, x) &\to x \\
@(@(K, x), y) &\to x \\
@(@(@(S, x), y), z) &\to @(@(x, z), @(y, z)) \\
f(x, a, b) &\to c \\
f(b, x, a) &\to c \\
f(a, b, x) &\to c
\end{aligned}
$$

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

Example

$$
\begin{aligned}
Ix &\to x \\
Kxy &\to x \\
Sxyz &\to xz(yz) \\
f(x,a,b) &\to c \\
f(b,x,a) &\to c \\
f(a,b,x) &\to c
\end{aligned}
$$

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

Example

$$
\begin{aligned}
Ix &\to x \\
Kxy &\to x \\
Sxyz &\to xz(yz) \\
f(x,a,b) &\to c \\
f(b,x,a) &\to c \\
f(a,b,x) &\to c
\end{aligned}
$$

$a \to b$, $a \to c$ is not non-overlapping

$e(x,x) \to \top$, $e(x,f(x)) \to \bot$, $a \to f(a)$ is not left-linear

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

Theorem (Rosen 73)

term in orthogonal rewrite system has at most one normal form

Proof.

by confluence/Church–Rosser property: any peak $s \leftarrow t \rightarrow u$ can be completed into valley $s \twoheadrightarrow r \twoheadleftarrow u$ by contracting residuals
$Sxyz \leftarrow Sxy(Iz) \rightarrow x(Iz)(y(Iz))$

This talk: retrofitting to orthogonal systems

Definition

term rewrite system is orthogonal if non-overlapping and left-linear

Theorem (Rosen 73)

term in orthogonal rewrite system has at most one normal form

Proof.

by confluence/Church–Rosser property: any peak $s \leftarrow t \rightarrow u$ can be completed into valley $s \twoheadrightarrow r \twoheadleftarrow u$ by contracting residuals

$Sxyz \leftarrow Sxy(Iz) \rightarrow x(Iz)(y(Iz))$

$Sxyz \rightarrow xz(yz) \leftarrow x(Iz)(yz) \leftarrow x(Iz)(y(Iz))$

colours indicate descendants; residual if redex-pattern descends □

Normalising strategies for orthogonal systems?

Non-trivial

normal order strategy not normalising for orthogonal systems

Normalising strategies for orthogonal systems?

Non-trivial

normal order strategy not normalising for orthogonal systems

Example

normal order reduces $f(\Omega, Ia, b)$ to itself in union of CL and

$$
\begin{array}{rcl}
Ix & \to & x \\
Kxy & \to & x \\
Sxyz & \to & xz(yz) \\
f(x,a,b) & \to & c \\
f(b,x,a) & \to & c \\
f(a,b,x) & \to & c
\end{array}
$$

$\Omega = SII(SII)$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Existence?

Ia is needed in $f(\Omega, Ia, b)$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Existence?

Ia is needed in $f(\Omega, Ia, b)$
needed redex in $f(t_1, t_2, t_3)$? (assuming t_i not in normal form)

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Existence?

Ia is needed in $f(\Omega, Ia, b)$
needed redex in $f(t_1, t_2, t_3)$? (assuming t_i not in normal form)
no needed redex in $g(Ia, Ia)$ for union of CL with

$$\begin{array}{rcl} g(a, x) & \to & a \\ g(x, a) & \to & a \end{array}$$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Existence?

Ia is needed in $f(\Omega, Ia, b)$
needed redex in $f(t_1, t_2, t_3)$? (assuming t_i not in normal form)
no needed redex in $g(Ia, Ia)$ for union of CL with

$$\begin{array}{rcl} g(a, x) & \rightarrow & a \\ g(x, a) & \rightarrow & a \end{array}$$

not orthogonal (but confluent)

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Normalisation?

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Normalisation?

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$
needed reduction may loop on a for

$$\begin{aligned} a &\to a \\ a &\to b \end{aligned}$$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Normalisation?

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$
needed reduction may loop on a for

$$\begin{aligned} a &\to a \\ a &\to b \end{aligned}$$

not orthogonal (but confluent)

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Huet, Lévy 91).

external strategy is needed and normalising
needed reduction bounded by external reduction □

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Bethke, Klop, de Vrijer 99).

symbol needed if contributes to normal form

- all symbols in normal form needed
- $t \rightarrow s$, needed prefix s' of s has needed prefix t' of t origin □

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Bethke, Klop, de Vrijer 99).

symbol needed if contributes to normal form

- all symbols in normal form needed
- $t \to s$, needed prefix s' of s has needed prefix t' of t origin □

Example

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Bethke, Klop, de Vrijer 99).

symbol needed if contributes to normal form

- all symbols in normal form needed
- $t \to s$, needed prefix s' of s has needed prefix t' of t origin □

Example

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Bethke, Klop, de Vrijer 99).

symbol needed if contributes to normal form

- all symbols in normal form needed
- $t \to s$, needed prefix s' of s has needed prefix t' of t origin □

Example

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$

Needed

Definition (Huet, Lévy 91)

redex needed if residual contracted in any reduction to normal form

Theorem (Huet, Lévy 91)

needed reduction is normalising strategy

Proof. (Bethke, Klop, de Vrijer 99).

symbol needed if contributes to normal form

- all symbols in normal form needed
- $t \to s$, needed prefix s' of s has needed prefix t' of t origin □

Example

$f(\Omega, Ia, b) \to f(\Omega, a, b) \to c$

External strategy

Idea

- non-needed if erased; $f(a)$ if $f(x) \to b$

External strategy

Idea

- non-needed if erased; $f(a)$ if $f(x) \to b$
- to be erased must be below some redex-pattern; a below $f(x)$

External strategy

Idea

- non-needed if erased; $f(a)$ if $f(x) \to b$
- to be erased must be below some redex-pattern; a below $f(x)$
- never below a redex-pattern then definitely needed; e.g. head

External strategy

Idea

- non-needed if erased; $f(a)$ if $f(x) \to b$
- to be erased must be below some redex-pattern; a below $f(x)$
- never below a redex-pattern then definitely needed; e.g. head

Example

in term $f(a, a)$ for rule

$$\begin{array}{rcl} a & \to & b \\ f(b, x) & \to & c \end{array}$$

second a not below redex-pattern, but residual after one step is: $f(a, a) \to f(b, a) \to c$

External strategy

Definition

symbol outer if not below redex-pattern
redex-pattern outer if its head symbol is
symbol external if outer and descendant (if any) external again
redex-pattern external if outer and residual (if any) external again

External strategy

Definition

redex-pattern external if outer and residual (if any) external again

Lemma

may witness non-externality by reduction at parallel positions

Proof.

reduction R witnessing non-externality of f occurring in t

1. if step above f truncate before 1st such in R; if redex-pattern
   - overlaps f then no descendant after in R; contradiction
   - above f then shorter witness to non-externality
2. if step strictly below f drop last such in R; shorter witness □

External strategy

Definition

redex-pattern external if outer and residual (if any) external again

Lemma

may witness non-externality by reduction at parallel positions

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

External strategy

Definition

redex-pattern external if outer and residual (if any) external again

Lemma

may witness non-externality by reduction at parallel positions

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

Theorem

external origin of external prefix is external; contains external redex

Proof.

by lemma and orthogonality (for $s'[\ell]_p$) □

External strategy

Definition

redex-pattern external if outer and residual (if any) external again

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

Theorem

external origin of external prefix is external; contains external redex

Definition

external strategy: if $t = f(\vec{t})$ head normal form then recurse on $\vec{t}$, otherwise $t \twoheadrightarrow \ell^\sigma$ and contract step in external origin of ℓ

External normalisation

Theorem

external reduction has random descent

Proof.

if $s\ {}_e\!\!\leftarrow t \rightarrow_e u$ then $s = u$, or $s \rightarrow_e r\ {}_e\!\!\leftarrow u$ □

External normalisation

Theorem

external reduction has random descent

Definition

distance $d(t)$ #external steps to normal form, else ∞

External normalisation

Theorem

external reduction has random descent

Definition

distance $d(t)$ #external steps to normal form, else ∞

Lemma

if $t \to s$ then $d(t) \geqslant d(s)$

Proof.

by induction on $d(t)$; interesting case t has but is not normal form:
$s \;{}_e\!\leftarrow t \to u$ and $s \twoheadrightarrow r \;{}_e\!\leftarrow u$ (by orthogonality)
$d(t) = d(s) + 1$, $d(s) \geq d(r)$ by IH, $d(r) + 1 = d(u)$ □

External normalisation

Theorem

external reduction has random descent

Definition

distance $d(t)$ #external steps to normal form, else ∞

Lemma

if $t \to s$ then $d(t) \geqslant d(s)$

Theorem

external strategy is (hyper-)normalising

Proof.

by lemma; if $t \to_e s$ then $d(t) > d(s)$ if t has normal form □

Bounding needed by external reductions

Lemma

if $t \to_n s$ *then* $d(t) > d(s)$ *if* t *has normal form*

Proof.

by induction on $d(t)$; interesting case t has but is not normal form:
$s \;{}_e\!\leftarrow t \to_n u$ and $s \twoheadrightarrow s' \to_n r' \twoheadrightarrow r \;{}_e\!\leftarrow u$
$d(t) = d(s) + 1$, $d(s) \geq d(s') > d(r') \geq d(r)$ by IH,
$d(r) + 1 = d(u)$ □

Bounding needed by external reductions

Lemma

if $t \to_n s$ then $d(t) > d(s)$ if t has normal form

Proof.

by induction on $d(t)$; interesting case t has but is not normal form:
$s \; {}_e\!\leftarrow t \to_n u$ and $s \twoheadrightarrow s' \to_n r' \twoheadrightarrow r \; {}_e\!\leftarrow u$
$d(t) = d(s) + 1$, $d(s) \geq d(s') > d(r') \geq d(r)$ by IH,
$d(r) + 1 = d(u)$ □

Example

external > needed; term $f(a)$, rules $a \to b$, $f(x) \to g(x,x)$; $3 > 2$

Bounding needed by external reductions

Lemma

if $t \to_n s$ then $d(t) > d(s)$ if t has normal form

Proof.

by induction on $d(t)$; interesting case t has but is not normal form:
$s \ {}_e\!\leftarrow t \to_n u$ and $s \twoheadrightarrow s' \to_n r' \twoheadrightarrow r \ {}_e\!\leftarrow u$
$d(t) = d(s) + 1$, $d(s) \geq d(s') > d(r') \geq d(r)$ by IH,
$d(r) + 1 = d(u)$ □

Theorem

needed strategy is (hyper-)normalising

Proof.

by lemmata distance never increases, decreases by needed steps □

Needed prefix vs. external prefix

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

Needed prefix vs. external prefix

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

Definition (Bethke, Klop, de Vrijer 99)

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then needed origin of needed prefix s' of s is: if p in s' then $s'[\ell^\tau]_p$ with τ mgu of $s'|_p$ and r, otherwise s'

Needed prefix vs. external prefix

Definition

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then external origin of external prefix s' of s is: if p in s' then $s'[\ell]_p$, otherwise s'

Definition (Bethke, Klop, de Vrijer 99)

if $t = C[\ell^\sigma]_p \to C[r^\sigma]_p = s$ then needed origin of needed prefix s' of s is: if p in s' then $s'[\ell^\tau]_p$ with τ mgu of $s'|_p$ and r, otherwise s'

Example

$f(f(a)) \to f(a) \to a$ for rule $f(x) \to x$

external origin: $f(f(a)) \to f(a) \to a$

needed origin: $f(f(a)) \to f(a) \to a$

Normalisation by head needeness

Definition

term t in head normal form if not $t \twoheadrightarrow \ell^{\sigma}$

head needed: residual contracted in reductions to head normal form

Normalisation by head needeness

Definition

term t in head normal form if not $t \twoheadrightarrow \ell^\sigma$

head needed: residual contracted in reductions to head normal form

Example

$f(a)$ in head normal form for $a \to b$, not for $a \to b$, $f(b) \to f(b)$

Normalisation by head neededness

Definition

term t in head normal form if not $t \twoheadrightarrow \ell^\sigma$

head needed: residual contracted in reductions to head normal form

Definition

head external reduction: if $t \twoheadrightarrow \ell^\sigma$ contract step in external origin ℓ

Theorem

head needed reduction is (hyper-)head-normalising strategy

Proof.

head distance $hd(t)$ #head external steps to hnf

if $t \to s$ then $hd(t) \geq hd(s)$; if $t \to_{hn} s$ then $hd(t) > hd(s)$ □

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Idea

- random descent only requires redexes have unique residuals

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Idea

- random descent only requires redexes have unique residuals
- so include redexes below linear (in rhs) variables

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Idea

- random descent only requires redexes have unique residuals
- so include redexes below linear (in rhs) variables
-

$$\begin{array}{rcl}(\lambda x.M(x))N & \to & M(N)\\ \lambda x.Mx & \to & M\end{array}$$

both linear in function (left argument @), function body (M)

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Definition

spine: if head normal form recur, else head spine
head spine: recur on left

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Definition

spine: if head normal form recur, else head spine
head spine: recur on left

Example

$\lambda x.I(KIx)x \to \lambda x.KIxx \to \lambda x.Ix \to I$

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Definition

spine: if head normal form recur, else head spine
head spine: recur on left

Example

$\lambda x.I(KIx)x \to \lambda x.KIxx \to \lambda x.Ix \to I$

Theorem

spine reduction is (hyper-)normalising strategy

Non-orthogonal left-linear systems

Example

$\lambda\beta\eta$-calculus; $\lambda x.I(KIx)x$ no external redex

Definition

spine: if head normal form recur, else head spine
head spine: recur on left

Example

$\lambda x.I(KIx)x \to \lambda x.KIxx \to \lambda x.Ix \to I$

Theorem

spine reduction is (hyper-)normalising strategy

Proof.

spine distance $ds(t)$ #spine steps to nf □

Conclusion

- (head) external strategy having random descent ⇒ distance
- distance ⇒ prove normalisation by induction on distance
- generalises from external prefixes to linear prefixes (non-ortho)