

TAIL RECURSION TRANSFORMATION

Alessandro Coglio

Kestrel Institute

2016

Associative Binary Operator

old function: $f(\bar{x}) \triangleq \underline{\text{if}} \ a(\bar{x}) \ \underline{\text{then}} \ b(\bar{x}) \ \underline{\text{else}} \ c(\bar{x}) * f(\bar{d}(\bar{x}))$ some binary operator
 $\bar{x} = (x_1, \dots, x_n) \quad \bar{d}(\bar{x}) = (d_1(\bar{x}), \dots, d_n(\bar{x})) \quad n > 0$

$$\boxed{\tau_f} \quad \neg a(\bar{x}) \Rightarrow \mu_f(\bar{d}(\bar{x})) <_f \mu_f(\bar{x})$$

$$f(\bar{x}) = \underbrace{c(\bar{x}) * \left(c(\bar{d}(\bar{x})) * \left(c(\bar{d}^2(\bar{x})) * \dots * \left(c(\bar{d}^{m-1}(\bar{x})) * b(\bar{d}^m(\bar{x})) \right) \dots \right) \right)}_{\text{expansion of } f(\bar{x})} \quad m \geq 0 \quad \forall j \in \{0, \dots, m-1\}. \neg a(\bar{d}^j(\bar{x})) \quad a(\bar{d}^m(\bar{x}))$$

condition: $\boxed{ASC} \quad u * (v * w) = (u * v) * w$ — associativity $\Rightarrow (\mathcal{U}, *)$ is a semigroup

expansion of $f(\bar{x}) = \left(\dots \left(\left(c(\bar{x}) * c(\bar{d}(\bar{x})) \right) * c(\bar{d}^2(\bar{x})) \right) * \dots * c(\bar{d}^{m-1}(\bar{x})) \right) * b(\bar{d}^m(\bar{x}))$

initially, if $\neg a(\bar{x})$,
set $r := c(\bar{x})$ and
update $\bar{x} := \bar{d}(\bar{x})$

then, while $\neg a(\bar{x})$,
update $r := r * c(\bar{x})$
and $\bar{x} := \bar{d}(\bar{x})$

finally, when $a(\bar{x})$,
return $r * b(\bar{x})$

if $a(\bar{x})$ initially,
return $b(\bar{x})$

calculate $f(\bar{x})$ when $m > 0$

calculate $f(\bar{x})$ when $m = 0$



new function: $f'(\bar{x}, r) \triangleq \underline{\text{if}} \ a(\bar{x}) \ \underline{\text{then}} \ r * b(\bar{x}) \ \underline{\text{else}} \ f'(\bar{d}(\bar{x}), r * c(\bar{x}))$ — tail-recursive

$$\mu_{f'}(\bar{x}, r) \triangleq \mu_f(\bar{x}) \quad <_{f'} \triangleq <_f$$

relation between f and f' : $f(\bar{x}) = \underline{\text{if}} \ a(\bar{x}) \ \underline{\text{then}} \ b(\bar{x}) \ \underline{\text{else}} \ f'(\bar{d}(\bar{x}), c(\bar{x}))$

$\vdash \boxed{\tau_{f'}} \quad \neg a(\bar{x}) \Rightarrow \mu_{f'}(\bar{d}(\bar{x}), r * c(\bar{x})) <_{f'} \mu_{f'}(\bar{x}, r) \quad \text{--- } f' \text{ terminates}$

$$\neg a(\bar{x}) \xrightarrow{\tau_f} \mu_f(\bar{d}(\bar{x})) <_f \mu_f(\bar{x})$$

$$\delta_{\mu_{f'}} \parallel \delta_{\mu_f} \parallel \delta_{\mu_{f'}}$$

$$\mu_{f'}(\bar{d}(\bar{x}), r * c(\bar{x})) <_{f'} \mu_{f'}(\bar{x}, r)$$

QED

$\vdash \boxed{f'f} \quad f'(\bar{x}, r) = r * f(\bar{x}) \quad \text{--- relation between } f' \text{ and } f \text{ (} f' \text{ in terms of } f \text{)}$

induct f'

base) $a(\bar{x}) \xrightarrow{\delta_{f'}} f'(\bar{x}, r) = r * b(\bar{x}) = r * f(\bar{x})$
 $\delta_f \rightarrow f(\bar{x}) = b(\bar{x})$

step) $a(\bar{x}) \xrightarrow{\delta_f} f(\bar{x}) = c(\bar{x}) * f(\bar{d}(\bar{x}))$
 $\delta_{f'} \rightarrow f'(\bar{x}, r) = f'(\bar{d}(\bar{x}), r * c(\bar{x})) = (r * c(\bar{x})) * f(\bar{d}(\bar{x})) \stackrel{ASC}{=} r * (c(\bar{x}) * f(\bar{d}(\bar{x}))) = r * f(\bar{x})$
 $\mathbb{IH} \rightarrow f'(\bar{d}(\bar{x}), r * c(\bar{x})) = (r * c(\bar{x})) * f(\bar{d}(\bar{x}))$

QED

$\vdash \boxed{ff'} \quad f(\bar{x}) = \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), c(\bar{x})) \quad \text{--- relation between } f \text{ and } f' \text{ (} f \text{ in terms of } f' \text{)}$

$$f(\bar{x}) \xrightarrow{\delta_f} \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } c(\bar{x}) * f(\bar{d}(\bar{x})) = \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), c(\bar{x}))$$

$$f'f \xrightarrow[\bar{x} := \bar{d}(\bar{x})]{r := c(\bar{x})} f'(\bar{d}(\bar{x}), c(\bar{x})) = c(\bar{x}) * f(\bar{d}(\bar{x}))$$

QED

wrapper : $\tilde{f}(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), c(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x})$

Associative Binary Operator with Left Identity

conditions $\left\{ \begin{array}{l} \boxed{\text{ASC}} \quad u * (v * w) = (u * v) * w \quad \text{— associativity} \\ \boxed{\text{LI}} \quad b(\bar{x}) * u = u \quad \text{— left identity} \end{array} \right.$

$$f(\bar{x}) = c(\bar{x}) * \left(c(\bar{d}(\bar{x})) * \left(c(\bar{d}^2(\bar{x})) * \dots * \left(c(\bar{d}^{m-1}(\bar{x})) * b(\bar{d}^m(\bar{x})) \right) \dots \right) \right) \quad \text{— expansion of } f, \text{ as before}$$

$\Rightarrow //$

$$b(\bar{x}) * \left(c(\bar{x}) * \left(c(\bar{d}(\bar{x})) * \left(c(\bar{d}^2(\bar{x})) * \dots * \left(c(\bar{d}^{m-1}(\bar{x})) * b(\bar{d}^m(\bar{x})) \right) \dots \right) \right) \right)$$

ASC $\Rightarrow //$

$$\left(\dots \left(\left(\left(b(\bar{x}) * c(\bar{x}) \right) * c(\bar{d}(\bar{x})) \right) * c(\bar{d}^2(\bar{x})) \right) * \dots * c(\bar{d}^{m-1}(\bar{x})) \right) * b(\bar{d}^m(\bar{x}))$$

start with $r := b(\bar{x})$ instead of $r := c(\bar{x})$ \nearrow one more update $r := r * c(\bar{x})$ than before, no need to update $\bar{x} := \bar{d}(\bar{x})$ initially

\nwarrow no initial split on $a(\bar{x})$, because r starts as $b(\bar{x})$ instead of $c(\bar{x})$

\Downarrow

new function: $f'(\bar{x}, r) \triangleq \text{if } a(\bar{x}) \text{ then } r * b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), r * c(\bar{x}))$ — as in associative-only case

$\vdash \boxed{f'f} \quad f'(\bar{x}, r) = r * f(\bar{x})$ — as in associative-only case

$\vdash \boxed{ff'} \quad f(\bar{x}) = f'(\bar{x}, b(\bar{x}))$
 $f'f \xrightarrow{r := b(\bar{x})} f'(\bar{x}, b(\bar{x})) = b(\bar{x}) * f(\bar{x}) \stackrel{\text{LI}}{=} f(\bar{x})$
 QED

wrapper: $\tilde{f}(\bar{x}) \triangleq f'(\bar{x}, b(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x})$

Associative Binary Operator with Right Identity

conditions $\left\{ \begin{array}{l} \boxed{\text{ASC}} \quad u * (v * w) = (u * v) * w \quad \text{— associativity} \\ \boxed{\text{RI}} \quad u * b(\bar{x}) = u \quad \text{— right identity} \end{array} \right.$

$$f(\bar{x}) = c(\bar{x}) * \left(c(\bar{d}(\bar{x})) * \left(c(\bar{d}^2(\bar{x})) * \dots * \left(c(\bar{d}^{m-1}(\bar{x})) * b(\bar{d}^m(\bar{x})) \right) \dots \right) \right) \quad \text{— expansion of } f, \text{ as before}$$

ASC \longrightarrow ||

$$\left(\dots \left(\left(c(\bar{x}) * c(\bar{d}(\bar{x})) \right) * c(\bar{d}^2(\bar{x})) \right) * \dots * c(\bar{d}^{m-1}(\bar{x})) \right) * b(\bar{d}^m(\bar{x}))$$

if $m > 0$ \longrightarrow ||

$$\dots \left(\left(c(\bar{x}) * c(\bar{d}(\bar{x})) \right) * c(\bar{d}^2(\bar{x})) \right) * \dots * c(\bar{d}^{m-1}(\bar{x}))$$

if $m > 0$,
finally return r
instead of $r * b(\bar{x})$

if $m = 0$,
initially return $b(\bar{x})$,
as in associative-only case



new function: $f'(\bar{x}, r) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), r * c(\bar{x}))$ $\mu_{f'}, \prec_{f'}, \tau_{f'}$ as before

$\vdash \boxed{f'f} \quad f'(\bar{x}, r) = r * f(\bar{x})$

induct f' $\left\{ \begin{array}{l} \text{base) } \begin{array}{l} \delta_{f'} \nearrow f'(\bar{x}, r) = r \stackrel{\text{RI}}{=} r * b(\bar{x}) = r * f(\bar{x}) \\ \delta_f \searrow f(\bar{x}) = b(\bar{x}) \end{array} \end{array} \right.$

step) $\begin{array}{l} \delta_f \nearrow f(\bar{x}) = c(\bar{x}) * f(\bar{d}(\bar{x})) \\ \tau_{a(\bar{x})} \searrow f'(\bar{x}, r) = f'(\bar{d}(\bar{x}), r * c(\bar{x})) = (r * c(\bar{x})) * f(\bar{d}(\bar{x})) \stackrel{\text{ASC}}{=} r * (c(\bar{x}) * f(\bar{d}(\bar{x}))) = r * f(\bar{x}) \end{array}$

IH $\nearrow f'(\bar{d}(\bar{x}), r * c(\bar{x})) = (r * c(\bar{x})) * f(\bar{d}(\bar{x}))$

QED

$\vdash \boxed{ff'} \quad f(\bar{x}) = \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), c(\bar{x})) \quad \text{— as in associative-only case}$

wrapper: $\tilde{f}(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), c(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x}) \quad \text{— as in associative-only case}$

Associative Binary Operator with Left and Right Identity

$$\text{conditions} \left\{ \begin{array}{ll} \boxed{\text{ASC}} & u * (v * w) = (u * v) * w \quad - \text{associativity} \\ \boxed{\text{LI}} & b(\bar{x}) * u = u \quad - \text{left identity} \\ \boxed{\text{RI}} & u * b(\bar{x}) = u \quad - \text{right identity} \end{array} \right. \Rightarrow (U, *, b_0) \text{ is a monoid}$$

$$\left(\begin{array}{l} \vdash \boxed{b.\text{const}} \quad b(\bar{x}) = b(\bar{y}) \\ \quad \left(\begin{array}{l} b(\bar{x}) = b(\bar{x}) * b(\bar{y}) \\ \quad \text{RI} \quad \quad \quad \text{LI} \end{array} \right) \Rightarrow b_0 \triangleq b(\bar{x}) \quad - \text{constant value of } b \\ \text{QED} \end{array} \right)$$

$$f(\bar{x}) = c(\bar{x}) * \left(c(\bar{d}(\bar{x})) * \left(c(\bar{d}^2(\bar{x})) * \dots * \left(c(\bar{d}^{m-1}(\bar{x})) * b(\bar{d}^m(\bar{x})) \right) \dots \right) \right) \quad - \text{expansion of } f, \text{ as before}$$

$\begin{array}{l} \text{ASC} \\ \text{LI} \\ \text{RI} \end{array} \rightarrow \parallel$

$$\dots \left(\left(\left(b(\bar{x}) * c(\bar{x}) \right) * c(\bar{d}(\bar{x})) \right) * c(\bar{d}^2(\bar{x})) \right) * \dots * c(\bar{d}^{m-1}(\bar{x}))$$

start with $r := b(\bar{x})$ \uparrow LI no initial $\bar{x} := \bar{d}(\bar{x})$ \uparrow LI no final $r * b(\bar{x})$ \uparrow RI no initial split on $a(\bar{x})$ \uparrow LI

} "combine" LI and RI

\Downarrow

new function: $f'(\bar{x}, r) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(\bar{d}(\bar{x}), r * c(\bar{x}))$ — as in associative-with-right-identity case

$$\vdash \boxed{f'f} \quad f'(\bar{x}, r) = r * f(\bar{x}) \quad - \text{as in associative-with-right-identity case}$$

$$\vdash \boxed{ff'} \quad f(\bar{x}) = f'(\bar{x}, b(\bar{x})) \quad - \text{as in associative-with-left-identity case}$$

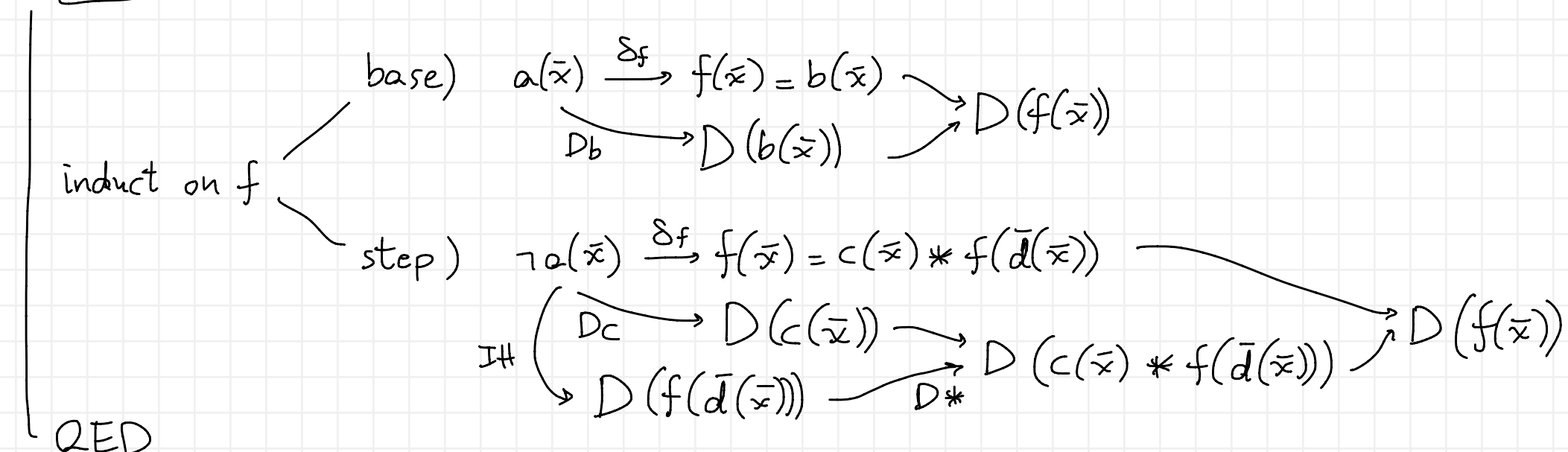
$$\text{wrapper: } \tilde{f}(\bar{x}) \triangleq f(\bar{x}, b(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x}) \quad - \text{as in associative-with-left-identity case}$$

Restriction of Operator Properties to a Domain

$D \subseteq U$ - domain

conditions	\boxed{Db}	$a(\bar{x}) \Rightarrow D(b(\bar{x}))$	- domain closure for b
	\boxed{Dc}	$\neg a(\bar{x}) \Rightarrow D(c(\bar{x}))$	- domain closure for c
	$\boxed{D*}$	$D(u) \wedge D(v) \Rightarrow D(u * v)$	- domain closure for $*$
	\boxed{ASC}	$D(u) \wedge D(v) \wedge D(w) \Rightarrow u * (v * w) = (u * v) * w$	- associativity
	\boxed{LI}	$a(\bar{x}) \wedge D(u) \Rightarrow b(\bar{x}) * u = u$	- left identity (optional)
	\boxed{RI}	$a(\bar{x}) \wedge D(u) \Rightarrow u * b(\bar{x}) = u$	- right identity (optional)

$\vdash \boxed{Df}$ $D(f(\bar{x}))$ - domain closure for f



$Db \wedge Dc \wedge D* \Rightarrow$ all the operands of $*$ in the expansion of $f(\bar{x})$ are in $D \Rightarrow ASC$ and RI apply

but LI applies only if $b(\bar{x})$ is such that $a(\bar{x})$ holds \Rightarrow we cannot use $b(\bar{x})$ for any \bar{x} in general



calculate, from any \bar{x} , some $b(\bar{x})$ such that $a(\bar{x})$ holds (i.e. go to the "bottom" of the recursion of f):

$$\beta(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } \beta(d(\bar{x}))$$

$$\mu_\beta(\bar{x}) \triangleq \mu_f(\bar{x})$$

$$<_\beta \triangleq <_f$$

$$\vdash [\tau_\beta] \neg a(\bar{x}) \Rightarrow \mu_\beta(d(\bar{x})) <_\beta \mu_\beta(\bar{x})$$

$$\begin{array}{c} \neg a(\bar{x}) \xrightarrow{\tau_f} \mu_f(d(\bar{x})) <_f \mu_f(\bar{x}) \\ \delta_{\mu_f} \parallel \quad \delta_{<_f} \parallel \quad \parallel \delta_{\mu_\beta} \\ \mu_\beta(d(\bar{x})) <_\beta \mu_\beta(\bar{x}) \end{array}$$

QED

$$\vdash [D\beta] D(\beta(\bar{x}))$$

$$\begin{array}{l} \text{induct on } \beta \\ \begin{array}{l} \text{base) } a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = b(\bar{x}) \xrightarrow{D_b} D(b(\bar{x})) \rightarrow D(\beta(\bar{x})) \\ \text{step) } \neg a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = \beta(d(\bar{x})) \xrightarrow{I\#} D(\beta(d(\bar{x}))) \rightarrow D(\beta(\bar{x})) \end{array} \end{array}$$

QED

$$LI \Rightarrow \vdash [LI\beta] D(u) \Rightarrow \beta(\bar{x}) * u = u$$

$$\begin{array}{l} \text{induct on } \beta \\ \begin{array}{l} \text{base) } a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = b(\bar{x}) \xrightarrow{D(u)}_{LI} b(\bar{x}) * u = u \rightarrow \beta(\bar{x}) * u = u \\ \text{step) } \neg a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = \beta(d(\bar{x})) \xrightarrow{D(u)}_{I\#} \beta(d(\bar{x})) * u = u \rightarrow \beta(\bar{x}) * u = u \end{array} \end{array}$$

QED

$$RI \Rightarrow \vdash [RI\beta] D(u) \Rightarrow u * \beta(\bar{x}) = u$$

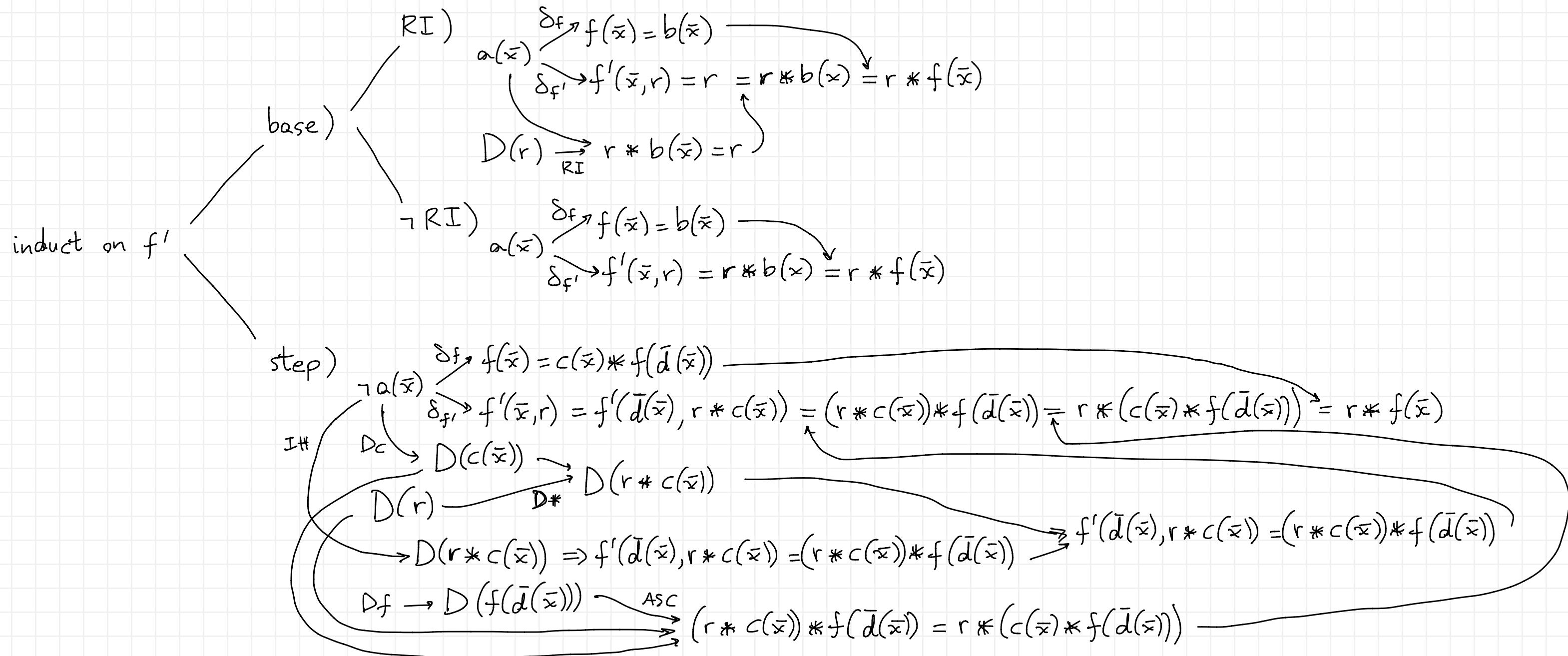
$$\begin{array}{l} \text{induct on } \beta \\ \begin{array}{l} \text{base) } a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = b(\bar{x}) \xrightarrow{D(u)}_{RI} u * b(\bar{x}) = u \rightarrow u * \beta(\bar{x}) = u \\ \text{step) } \neg a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = \beta(d(\bar{x})) \xrightarrow{D(u)}_{I\#} u * \beta(d(\bar{x})) = u \rightarrow u * \beta(\bar{x}) = u \end{array} \end{array}$$

QED

new function: $f'(\bar{x}, r) \triangleq \begin{cases} \text{if } a(\bar{x}) \text{ then } r & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) \\ \text{if } a(\bar{x}) \text{ then } r * b(\bar{x}) & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) \end{cases} \begin{matrix} \Leftarrow RI \\ \Leftarrow \neg RI \end{matrix}$

— as when $D = \mathcal{U}$

$\vdash \boxed{f'f} \quad D(r) \Rightarrow f'(\bar{x}, r) = r * f(\bar{x})$



QED

β and LI are not used in f' and $f'f$

$$\begin{array}{l}
LI \Rightarrow \vdash \boxed{ff'} \quad f(\bar{x}) = f'(\bar{x}, \beta(\bar{x})) \\
\left\{ \begin{array}{l}
D\beta \rightarrow D(\beta(\bar{x})) \xrightarrow{\quad} f'(\bar{x}, \beta(\bar{x})) = \beta(\bar{x}) * f(\bar{x}) = f(\bar{x}) \\
f'f \xrightarrow[r := \beta(\bar{x})]{} f'(\bar{x}, \beta(\bar{x})) = \beta(\bar{x}) * f(\bar{x}) = f(\bar{x}) \\
Df \rightarrow D(f(\bar{x})) \xrightarrow{LI\beta} f'(\bar{x}, \beta(\bar{x})) = \beta(\bar{x}) * f(\bar{x}) = f(\bar{x})
\end{array} \right. \\
QED
\end{array}$$

$$\text{wrapper: } \tilde{f}(\bar{x}) \triangleq f'(\bar{x}, \beta(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x}) \quad - \text{ as when } D = \mathcal{U}$$

$$\begin{array}{l}
\neg LI \Rightarrow \vdash \boxed{ff'} \quad f(\bar{x}) = \underline{\text{if}} \, a(\bar{x}) \, \underline{\text{then}} \, b(\bar{x}) \, \underline{\text{else}} \, f'(\bar{d}(\bar{x}), c(\bar{x})) \\
\left\{ \begin{array}{l}
f(\bar{x}) \stackrel{\delta_f}{=} \underline{\text{if}} \, a(\bar{x}) \, \underline{\text{then}} \, b(\bar{x}) \, \underline{\text{else}} \, c(\bar{x}) * f(\bar{d}(\bar{x})) = \underline{\text{if}} \, a(\bar{x}) \, \underline{\text{then}} \, b(\bar{x}) \, \underline{\text{else}} \, f'(\bar{d}(\bar{x}), c(\bar{x})) \\
f'f \xrightarrow[\bar{x} := \bar{d}(\bar{x})]{r := c(\bar{x})} D(c(\bar{x})) \Rightarrow f'(\bar{d}(\bar{x}), c(\bar{x})) = c(\bar{x}) * f(\bar{d}(\bar{x})) \\
Dc \rightarrow \neg a(\bar{x}) \Rightarrow D(c(\bar{x})) \xrightarrow{\quad} f'(\bar{d}(\bar{x}), c(\bar{x})) = c(\bar{x}) * f(\bar{d}(\bar{x}))
\end{array} \right. \\
QED
\end{array}$$

$$\text{wrapper: } \tilde{f}(\bar{x}) \triangleq \underline{\text{if}} \, a(\bar{x}) \, \underline{\text{then}} \, b(\bar{x}) \, \underline{\text{else}} \, f'(\bar{d}(\bar{x}), c(\bar{x})) \Rightarrow \vdash \boxed{f\tilde{f}} \quad f(\bar{x}) = \tilde{f}(\bar{x}) \quad - \text{ as when } D = \mathcal{U}$$

β is used only if LI holds ; only $LI\beta$ is used here, not $RI\beta$

$$LI \wedge RI \Rightarrow$$

$$\vdash \boxed{b.\text{const}} \quad a(\bar{x}) \wedge a(\bar{y}) \Rightarrow b(\bar{x}) = b(\bar{y})$$

$$\begin{array}{l} \left(\begin{array}{l} a(\bar{x}) \xrightarrow{Db} D(b(\bar{x})) \xrightarrow{RI} b(\bar{x}) * b(\bar{y}) = b(\bar{x}) \\ a(\bar{y}) \xrightarrow{Db} D(b(\bar{y})) \xrightarrow{LI} b(\bar{x}) * b(\bar{y}) = b(\bar{y}) \end{array} \right) \rightarrow b(\bar{x}) = b(\bar{y}) \end{array}$$

QED

$$\vdash \boxed{\beta b} \quad a(\bar{y}) \Rightarrow \beta(\bar{x}) = b(\bar{y})$$

induct on β

$$\begin{array}{l} \text{base) } \begin{array}{l} a(\bar{x}) \xrightarrow{\delta\beta} \beta(\bar{x}) = b(\bar{x}) \\ a(\bar{y}) \xrightarrow{b.\text{const}} b(\bar{x}) = b(\bar{y}) \end{array} \rightarrow \beta(\bar{x}) = b(\bar{y}) \\ \text{step) } \begin{array}{l} \neg a(\bar{x}) \xrightarrow{\delta\beta} \beta(\bar{x}) = \beta(\bar{d}(\bar{x})) \\ a(\bar{y}) \xrightarrow{\neq H} \beta(\bar{d}(\bar{x})) = b(\bar{y}) \end{array} \rightarrow \beta(\bar{x}) = b(\bar{y}) \end{array}$$

QED

$$\vdash \boxed{\beta.\text{const}} \quad \beta(\bar{x}) = \beta(\bar{y})$$

induct on β

$$\begin{array}{l} \text{base) } \begin{array}{l} a(\bar{x}) \xrightarrow{\delta\beta} \beta(\bar{x}) = b(\bar{x}) \\ \xrightarrow[\substack{\bar{x} := \bar{y} \\ \bar{y} := \bar{x}}]{\beta b} \beta(\bar{y}) = b(\bar{x}) \end{array} \rightarrow \beta(\bar{x}) = \beta(\bar{y}) \\ \text{step) } \begin{array}{l} \neg a(\bar{x}) \xrightarrow{\delta\beta} \beta(\bar{x}) = \beta(\bar{d}(\bar{x})) \\ \xrightarrow{\neq H} \beta(\bar{d}(\bar{x})) = \beta(\bar{y}) \end{array} \rightarrow \beta(\bar{x}) = \beta(\bar{y}) \end{array}$$

QED

$b_0 \triangleq \beta(\bar{x})$ — constant value of β , i.e. of b under a

$$\vdash \boxed{LI_0} \quad D(u) \Rightarrow b_0 * u = u$$

$$\begin{array}{l} D(u) \xrightarrow{LI\beta} \beta(\bar{x}) * u = u \xrightarrow{\delta b_0} b_0 * u = u \\ \text{QED} \end{array}$$

$$\vdash \boxed{RI_0} \quad D(u) \Rightarrow u * b_0 = u$$

$$\begin{array}{l} D(u) \xrightarrow{RI\beta} u * \beta(\bar{x}) = u \xrightarrow{\delta b_0} u * b_0 = u \\ \text{QED} \end{array}$$

$$\vdash \boxed{Db_0} \quad D(b_0)$$

$$\begin{array}{l} D\beta \rightarrow D(\beta(\bar{x})) \xrightarrow{\delta b_0} D(b_0) \\ \text{QED} \end{array}$$

$$D * \wedge Db_0 \wedge ASC \wedge LI_0 \wedge RI_0$$

\Downarrow
 $(D, *, b_0)$ is a monoid

Guards

old function: $f(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } c(\bar{x}) * f(\bar{d}(\bar{x}))$

$$\gamma_{\bar{d}}(\bar{x}) = \gamma_{d_1}(\bar{x}) \wedge \dots \wedge \gamma_{d_n}(\bar{x})$$

$$\boxed{\sqrt{f}} \quad \gamma_{\gamma_f}(\bar{x}) \wedge [\gamma_f(\bar{x}) \Rightarrow \gamma_a(\bar{x}) \wedge [a(\bar{x}) \Rightarrow \gamma_b(\bar{x})] \wedge [\neg a(\bar{x}) \Rightarrow \gamma_c(\bar{x}) \wedge \gamma_{\bar{d}}(\bar{x}) \wedge \gamma_f(\bar{d}(\bar{x})) \wedge \gamma_*(c(\bar{x}), f(\bar{d}(\bar{x})))]]]$$

$D \subseteq \mathcal{U}$ - domain

conditions $\begin{cases} \boxed{GD} & \gamma_D = \mathcal{U} & - D \text{ always well-defined} \\ \boxed{G*} & \gamma_* \geq D \times D & - * \text{ well-defined at least over } D \end{cases}$

new function: $f'(\bar{x}, r) \triangleq \begin{cases} \text{if } a(\bar{x}) \text{ then } r & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) & \Leftarrow RI \\ \text{if } a(\bar{x}) \text{ then } r * b(\bar{x}) & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) & \Leftarrow \neg RI \end{cases}$ - as before

$$\gamma_{f'}(\bar{x}, r) \triangleq [\gamma_f(\bar{x}) \wedge D(r)]$$

$\vdash \boxed{\sqrt{f'}} \quad \omega_{f'}(\bar{x}, r)$

RI)

$$\omega_{f'}(\bar{x}, r) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \cancel{\gamma_D(r)} \wedge [\cancel{\gamma_f(\bar{x})} \wedge D(r) \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge [\neg a(\bar{x}) \Rightarrow \cancel{\gamma_{\bar{d}}(\bar{x})} \wedge \cancel{\gamma_c(\bar{x})} \wedge \gamma_*(r, c(\bar{x})) \wedge \gamma_{f'}(\bar{d}(\bar{x})) \wedge D(r * c(\bar{x}))]]]$$

Annotations: \sqrt{f} (over γ_{γ_f}), GD (over γ_D), \sqrt{f} (over γ_f), \sqrt{f} (over γ_a), \sqrt{f} (over $\gamma_{\bar{d}}$), \sqrt{f} (over γ_c), $G*$ (over γ_*), \sqrt{f} (over $\gamma_{f'}$), $D_c \rightarrow D(c(\bar{x}))$ (under γ_c), D_* (over $D(r * c(\bar{x}))$).

$\neg RI)$

$$\omega_{f'}(\bar{x}, r) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \cancel{\gamma_D(r)} \wedge [\cancel{\gamma_f(\bar{x})} \wedge D(r) \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge [a(\bar{x}) \Rightarrow \cancel{\gamma_b(\bar{x})} \wedge \gamma_*(r, b(\bar{x}))]] \wedge [\neg a(\bar{x}) \Rightarrow \dots]]]$$

Annotations: \sqrt{f} (over γ_{γ_f}), GD (over γ_D), \sqrt{f} (over γ_f), \sqrt{f} (over γ_a), \sqrt{f} (over γ_b), $G*$ (over γ_*), $D_b \rightarrow D(b(\bar{x}))$ (under γ_b), \sqrt{f} (over $\gamma_{f'}$), $\neg RI$ (over the second part of the formula), as with RI (under the second part of the formula).

QED

$\beta(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } \beta(d(\bar{x}))$ — as before

$$\gamma_\beta(\bar{x}) \triangleq \gamma_f(\bar{x})$$

$$\vdash \boxed{\sqrt{\beta}} \quad \omega_\beta(\bar{x})$$

$$\omega_\beta(\bar{x}) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \left[\cancel{\gamma_f(\bar{x})} \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge \left[\cancel{a(\bar{x})} \Rightarrow \cancel{\gamma_b(\bar{x})} \right] \wedge \left[\cancel{\neg a(\bar{x})} \Rightarrow \cancel{\gamma_{\bar{d}}(\bar{x})} \wedge \cancel{\gamma_f(d(\bar{x}))} \right] \right]$$

QED

wrapper : $\tilde{f}(\bar{x}) \triangleq \begin{cases} f'(\bar{x}, \beta(\bar{x})) \\ \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } f'(d(\bar{x}), c(\bar{x})) \end{cases} \quad \begin{matrix} \Leftarrow \text{LI} \\ \Leftarrow \neg \text{LI} \end{matrix} \quad \text{— as before}$

$$\gamma_{\tilde{f}}(\bar{x}) \triangleq \gamma_f(\bar{x})$$

$$\vdash \boxed{\sqrt{\tilde{f}}} \quad \omega_{\tilde{f}}(\bar{x})$$

$$\text{LI) } \omega_{\tilde{f}}(\bar{x}) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \left[\cancel{\gamma_f(\bar{x})} \xRightarrow{\delta_{\gamma_f}} \cancel{\gamma_\beta(\bar{x})} \wedge \cancel{\gamma_{f'}(\bar{x}, \beta(\bar{x}))} \right]$$

$\neg \text{LI) }$

$$\omega_{\tilde{f}}(\bar{x}) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \left[\cancel{\gamma_f(\bar{x})} \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge \left[\cancel{a(\bar{x})} \Rightarrow \cancel{\gamma_b(\bar{x})} \right] \wedge \left[\cancel{\neg a(\bar{x})} \Rightarrow \cancel{\gamma_{\bar{d}}(\bar{x})} \wedge \cancel{\gamma_c(\bar{x})} \wedge \cancel{\gamma_{f'}(d(\bar{x}), c(\bar{x}))} \right] \right]$$

QED

if \tilde{f} is not generated, a proof like this establishes that the body of \tilde{f} is guard-verified under $\gamma_f(\bar{x})$:
this theorem is useful to guard-verify terms where a call to f is replaced with the (instantiated) body of \tilde{f}

Decomposition of the Old Function

$f(\bar{x}) \triangleq \underline{\text{if}}\ t_1\ \underline{\text{then}}\ t_2\ \underline{\text{else}}\ t_3$ t_1, t_2, t_3 terms without let f occurs in t_3 , not in t_1 or t_2

$$a \triangleq \lambda \bar{x}. t_1$$

$$b \triangleq \lambda \bar{x}. t_2$$

all calls to f in t_3 must be identical: $f(s_1, \dots, s_n)$ s_1, \dots, s_n terms not containing f $n > 0$

$$d_i \triangleq \lambda \bar{x}. s_i$$

$$\tilde{t}_3 \triangleq t_3[f(s_1, \dots, s_n)/r], \text{ } r \text{ fresh variable} \begin{cases} r \in FV(\tilde{t}_3), \text{ otherwise } f \text{ would not be recursive} \\ \tilde{t}_3 \neq r, \text{ otherwise } f \text{ would be already tail-recursive} \end{cases}$$

$$C \triangleq \{(c, *) \in (\mathcal{U}^n \rightarrow \mathcal{U}) \times (\mathcal{U} \times \mathcal{U} \rightarrow \mathcal{U}) \mid \tilde{t}_3 = c(\bar{x}) * r\} \text{ — candidates for } c \text{ and } *$$

$$C = \emptyset, \text{ e.g. } \tilde{t}_3 \equiv g(x_1, x_2, r)$$

$$|C| = 1, \text{ e.g. } \tilde{t}_3 \equiv g(x_1, r), C = \{(id, g)\}$$

$$|C| > 1, \text{ e.g. } \tilde{t}_3 \equiv g(h(x_1), r), C = \{(h, g), (id, \lambda(q, r). g(h(q), r))\}$$

exactly one $(c, *) \in C$ for each term s such that $r \notin FV(s)$ and $FV(\tilde{t}_3[s/q]) \subseteq \{q, r\}$, q fresh variable
 s includes all occurrences of \bar{x} in \tilde{t}_3

special case: $FV(\tilde{t}_3) = \{r\}$, i.e. \bar{x} do not occur in \tilde{t}_3

\Rightarrow any s would do, but then $*$ may ignore its first argument, making ASC, LI, RI less likely

when two such terms s and s' are one a subterm of the other, one is not always better than the other, e.g.:

$$\tilde{t}_3 \equiv -x_1 + r \begin{cases} s \equiv x_1 \Rightarrow * = \lambda(q, r). -q + r \text{ is not associative} \\ s' \equiv -x_1 \Rightarrow * = \lambda(q, r). q + r \text{ is associative} \end{cases} \Rightarrow \text{larger term is better}$$

$$\tilde{t}_3 \equiv -(-x_1) + r \begin{cases} s \equiv x_1 \Rightarrow * = \lambda(q, r). -(-q) + r \text{ is associative} \\ s' \equiv -x_1 \Rightarrow * = \lambda(q, r). -q + r \text{ is not associative} \end{cases} \Rightarrow \text{smaller term is better}$$

Special Case: Ground Base Value

old function: $f(\bar{x}) \triangleq \text{if } t_1 \text{ then } t_2 \text{ else } t_3$ — as in discussion about decomposition

$FV(t_2) = \emptyset$ — ground base value $\Rightarrow b \triangleq \lambda \bar{x}. b_0$, $b_0 \in \mathcal{U} \Rightarrow f(\bar{x}) = \text{if } a(\bar{x}) \text{ then } b_0 \text{ else } c(\bar{x}) * f(\bar{d}(\bar{x}))$

$\vdash \boxed{\beta_0} \beta(\bar{x}) = b_0$

induct on β

- base) $a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = b_0$
- step) $\neg a(\bar{x}) \xrightarrow{\delta_\beta} \beta(\bar{x}) = \beta(\bar{d}(\bar{x})) = b_0$
 $\text{IH} \rightarrow \beta(\bar{d}(\bar{x})) = b_0$

QED

$\vdash \boxed{D b_0} D(b_0)$

$D\beta \rightarrow D(\beta(\bar{x})) \xrightarrow{\beta_0} D(b_0)$

QED

LI $\Rightarrow \vdash \boxed{LI_0} D(u) \Rightarrow b_0 * u = u$

$D(u) \xrightarrow{LI\beta} \beta(\bar{x}) * u = u \xrightarrow{\beta_0} b_0 * u = u$

QED

RI $\Rightarrow \vdash \boxed{RI_0} D(u) \Rightarrow u * b_0 = u$

$D(u) \xrightarrow{RI\beta} u * \beta(\bar{x}) = u \xrightarrow{\beta_0} u * b_0 = u$

QED

$D * \wedge D b_0 \wedge ASC \wedge LI_0 \wedge RI_0 \Rightarrow (D, *, b_0) \text{ is a monoid}$

f' and f'/f are the same as before (they do not use β and LI)

LI $\Rightarrow \vdash \boxed{ff'_0} f(\bar{x}) = f'(\bar{x}, b_0)$

$D b_0 \rightarrow D(b_0) \xrightarrow{r := b_0} f'(\bar{x}, b_0) = b_0 * f(\bar{x}) = f(\bar{x})$

$f'/f \xrightarrow{r := b_0} f'(\bar{x}, b_0) = b_0 * f(\bar{x}) = f(\bar{x})$

$Df \rightarrow D(f(\bar{x})) \xrightarrow{LI_0} f'(\bar{x}, b_0) = b_0 * f(\bar{x}) = f(\bar{x})$

QED

wrapper: $\tilde{f}(\bar{x}) \triangleq f'(\bar{x}, b_0) \Rightarrow \vdash \boxed{f\tilde{f}} f(\bar{x}) = \tilde{f}(\bar{x})$ — as before

$\alpha(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } \bar{x} \text{ else } \alpha(d(\bar{x}))$ $\alpha: \mathcal{U}^n \rightarrow \mathcal{U}^n$ — calculate \bar{x}_0 such that $a(\bar{x}_0)$ holds, from any \bar{x}

$$\mu_\alpha(\bar{x}) \triangleq \mu_f(\bar{x}) \quad \angle_\alpha \triangleq \angle_f$$

$$\begin{array}{l} \vdash \boxed{\tau_\alpha} \quad \neg a(\bar{x}) \Rightarrow \mu_\alpha(d(\bar{x})) <_\alpha \mu_\alpha(\bar{x}) \\ \quad \neg a(\bar{x}) \xrightarrow{\tau_f} \mu_f(d(\bar{x})) <_f \mu_f(\bar{x}) \\ \quad \quad \delta_{\mu_\alpha} \parallel \quad \delta_{\angle_\alpha} \parallel \quad \parallel \delta_{\mu_\alpha} \\ \quad \mu_\alpha(d(\bar{x})) <_\alpha \mu_\alpha(\bar{x}) \\ \text{QED} \end{array}$$

$$\begin{array}{l} \vdash \boxed{a\alpha} \quad a(\alpha(\bar{x})) \\ \text{induct on } \alpha \begin{cases} \text{base) } a(\bar{x}) \xrightarrow{\delta_\alpha} \alpha(\bar{x}) = \bar{x} \rightarrow a(\alpha(\bar{x})) \\ \text{step) } \neg a(\bar{x}) \xrightarrow{\delta_\alpha} \alpha(\bar{x}) = \alpha(d(\bar{x})) \xrightarrow{\text{IH}} a(\alpha(d(\bar{x}))) \rightarrow a(\alpha(\bar{x})) \end{cases} \\ \text{QED} \end{array}$$

$$\begin{array}{l} \sqrt{f} \Rightarrow \vdash \boxed{\gamma_f \alpha} \quad \gamma_f(\bar{x}) \Rightarrow \gamma_f(\alpha(\bar{x})) \\ \text{induct on } \alpha \begin{cases} \text{base) } a(\bar{x}) \xrightarrow{\delta_\alpha} \alpha(\bar{x}) = \bar{x} \xrightarrow{\gamma_f(\bar{x})} \gamma_f(\alpha(\bar{x})) \\ \text{step) } \neg a(\bar{x}) \xrightarrow{\delta_\alpha} \alpha(\bar{x}) = \alpha(d(\bar{x})) \xrightarrow{\text{IH}} \gamma_f(\alpha(d(\bar{x}))) \rightarrow \gamma_f(\alpha(\bar{x})) \\ \quad \left(\gamma_f(\bar{x}) \xrightarrow{\sqrt{f}} \gamma_f(d(\bar{x})) \xrightarrow{\text{IH}} \gamma_f(\alpha(d(\bar{x}))) \right) \end{cases} \\ \text{QED} \end{array}$$

$$b = \lambda \bar{x}. b_0 \Rightarrow \gamma_b \triangleq \lambda \bar{x}. \gamma_{b_0}, \quad \gamma_{b_0} \in \mathcal{U}$$

$$\begin{array}{l} \vdash \boxed{\gamma_{b_0}.const} \quad \gamma_b(\bar{x}) = \gamma_b(\bar{y}) \\ \quad \gamma_b(\bar{x}) \stackrel{\delta_{\gamma_b}}{=} \gamma_{b_0} \stackrel{\delta_{\gamma_b}}{=} \gamma_b(\bar{y}) \\ \text{QED} \end{array}$$

$$\begin{array}{l} \sqrt{f} \Rightarrow \vdash \boxed{Gb} \quad \gamma_f(\bar{x}) \Rightarrow \gamma_b(\bar{x}) \\ \quad \gamma_f(\bar{x}) \xrightarrow{\gamma_f \alpha} \gamma_f(\alpha(\bar{x})) \xrightarrow{\sqrt{f}} \gamma_b(\alpha(\bar{x})) \stackrel{\gamma_{b_0}.const}{=} \gamma_b(\bar{x}) \\ \quad a\alpha \rightarrow a(\alpha(\bar{x})) \xrightarrow{\quad} \gamma_b(\alpha(\bar{x})) \\ \text{QED} \end{array}$$

$$\begin{array}{l} \sqrt{f} \wedge LI \Rightarrow \vdash \boxed{\sqrt{f}} \quad \omega_{\tilde{f}}(\bar{x}) \\ \quad \omega_{\tilde{f}}(\bar{x}) = \cancel{\gamma_{\tilde{f}}(\bar{x})} \wedge \left[\gamma_f(\bar{x}) \Rightarrow \gamma_b(\bar{x}) \wedge \gamma_{f'}(\bar{x}, b_0) \right] \xrightarrow{Gb} \gamma_b(\bar{x}) \xrightarrow{Db_0, \delta_{\gamma_{f'}}} \gamma_{f'}(\bar{x}, b_0) \\ \text{QED} \end{array} \quad - \quad b_0 = b(\bar{x})$$

summary of this special case (ground base value): b_0 can be used instead of β (when LI holds)

note that Db_0 and LI_0 can be proved without using β :

$$\begin{array}{l} \vdash \boxed{Db_0} \quad D(b_0) \quad - \text{alternative proof} \\ \quad a\alpha \rightarrow a(\alpha(\bar{x})) \xrightarrow[\bar{x} := \alpha(\bar{x})]{Db} D(b(\alpha(\bar{x}))) \xrightarrow[\bar{x} := \alpha(\bar{x})]{\delta_b} D(b_0) \\ \text{QED} \end{array}$$

$$\begin{array}{l} LI \Rightarrow \vdash \boxed{LI_0} \quad D(u) \Rightarrow b_0 * u = u \quad - \text{alternative proof} \\ \quad a\alpha \rightarrow a(\alpha(\bar{x})) \xrightarrow[\bar{x} := \alpha(\bar{x})]{LI} b(\alpha(\bar{x})) * u = u \xrightarrow[\bar{x} := \alpha(\bar{x})]{\delta_b} b_0 * u = u \\ \text{QED} \end{array}$$

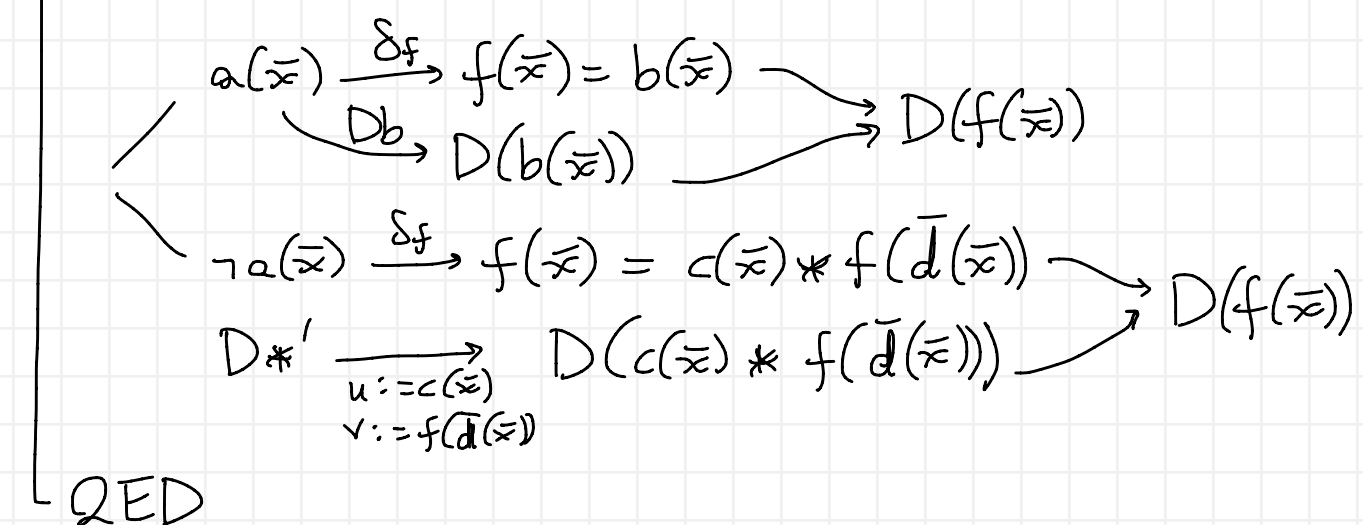
Extension of Operator Associativity and Closure outside the Domain

conditions	\boxed{Db}	$a(\bar{x}) \Rightarrow D(b(\bar{x}))$	— as before
	$\boxed{D*}'$	$D(u * v)$	— unconditional version of $D*$
	\boxed{ASC}'	$u * (v * w) = (u * v) * w$	— unconditional version of ASC
	\boxed{LI}	$a(\bar{x}) \wedge D(u) \Rightarrow b(\bar{x}) * u = u$	— as before, but required (not optional)
	\boxed{RI}	$a(\bar{x}) \wedge D(u) \Rightarrow u * b(\bar{x}) = u$	— as before, and still optional

$(Db \wedge D* \wedge ASC')$ is neither weaker nor stronger than $(Db \wedge Dc \wedge D* \wedge ASC)$

$D*'$ and ASC' may be satisfied when $*$ fixes its arguments to be in D

$\vdash \boxed{Df}$ $D(f(\bar{x}))$ — as before, but slightly different proof



$\beta(\bar{x}) \triangleq \text{if } a(\bar{x}) \text{ then } b(\bar{x}) \text{ else } \beta(\bar{d}(\bar{x}))$ — as before

$\vdash \boxed{D\beta}$ $D(\beta(\bar{x}))$ — as before, same proof

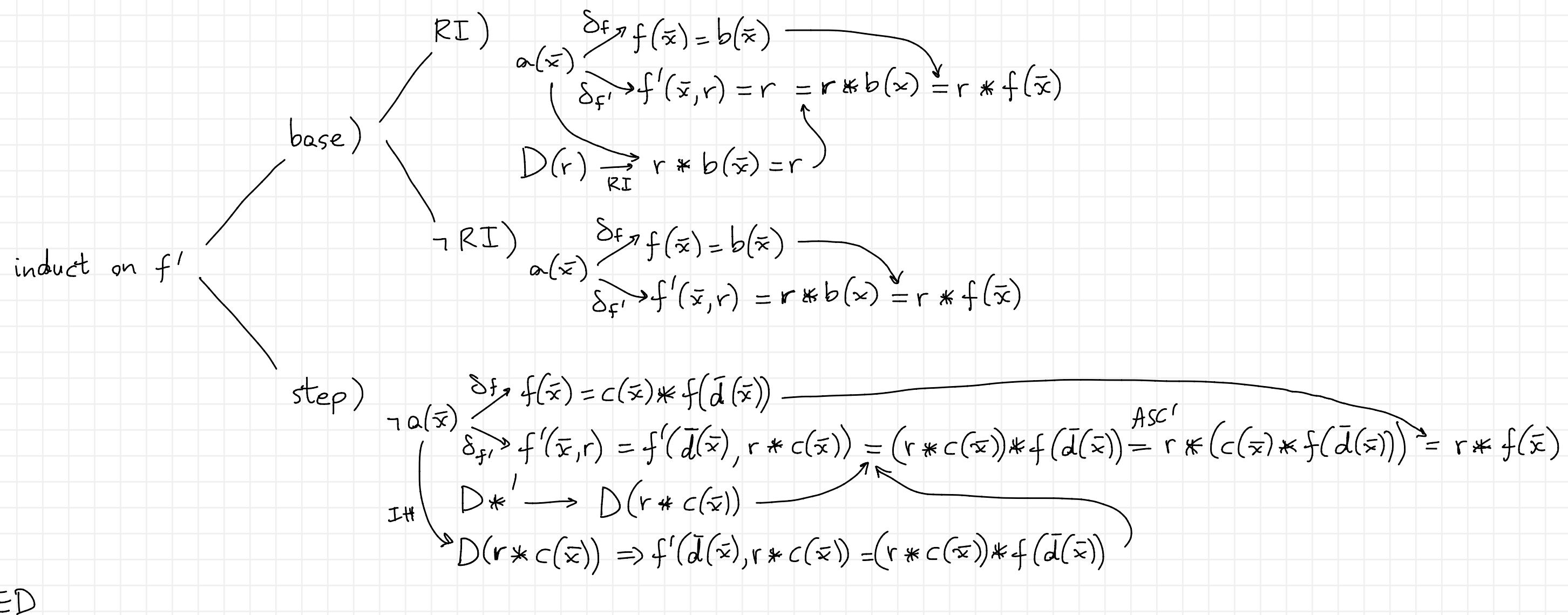
$\vdash \boxed{LI\beta}$ $D(u) \Rightarrow \beta(\bar{x}) * u = u$ — as before, same proof

$RI \Rightarrow \vdash \boxed{RI\beta}$ $D(u) \Rightarrow u * \beta(\bar{x}) = u$ — as before, same proof

$RI \Rightarrow (D, *, b_0)$ is a monoid — as before, same lemmas and proofs

new function: $f'(\bar{x}, r) \triangleq \begin{cases} \text{if } a(\bar{x}) \text{ then } r & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) \\ \text{if } a(\bar{x}) \text{ then } r * b(\bar{x}) & \text{else } f'(\bar{d}(\bar{x}), r * c(\bar{x})) \end{cases} \begin{matrix} \Leftarrow RI \\ \Leftarrow \neg RI \end{matrix}$ — as before

$\vdash \boxed{f'f}$ $D(r) \Rightarrow f'(\bar{x}, r) = r * f(\bar{x})$ — as before, but slightly different proof



$\vdash \boxed{ff'}$ $f(\bar{x}) = f'(\bar{x}, \beta(\bar{x}))$ — as before, same proof

the proof of ff' when LI does not hold needs Dc to use an instance of $f'f$

$\Rightarrow LI$ is required here; otherwise $(Db \wedge Dc \wedge D *' \wedge ASC')$ is stronger (i.e., less satisfiable) than $(Db \wedge Dc \wedge D * \wedge ASC)$

conditions $\begin{cases} \boxed{GD} & \gamma_D = \mathcal{U} & \text{— as before} \\ \boxed{G*} & \gamma_* \supseteq D \times D & \text{— as before} \\ \boxed{GDc} & \gamma_f(\bar{x}) \wedge \neg a(\bar{x}) \Rightarrow D(c(\bar{x})) & \text{— weaker version of } Dc, \text{ conditioned by the guard of } f \end{cases}$

$\gamma_{f'}(\bar{x}, r) \triangleq [\gamma_f(\bar{x}) \wedge D(r)]$ — as before

$\vdash \boxed{\sqrt{f'}}$ $\omega_{f'}(\bar{x}, r)$ — as before, but slightly different proof

RI)

$$\omega_{f'}(\bar{x}, r) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \cancel{\gamma_D(r)} \wedge [\cancel{\gamma_f(\bar{x})} \wedge D(r) \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge [\neg a(\bar{x}) \Rightarrow \cancel{\gamma_{\bar{a}}(\bar{x})} \wedge \cancel{\gamma_c(\bar{x})} \wedge \gamma_*(r, c(\bar{x})) \wedge \gamma_f(\bar{a}(\bar{x})) \wedge D(r * c(\bar{x}))]]$$

Annotations: \sqrt{f} over $\gamma_{\gamma_f}(\bar{x})$ and $\gamma_D(r)$; GD over $\gamma_f(\bar{x})$ and $D(r)$; \sqrt{f} over $\gamma_a(\bar{x})$; \sqrt{f} over $\gamma_{\bar{a}}(\bar{x})$ and $\gamma_c(\bar{x})$; GDc from $D(r)$ to $D(c(\bar{x}))$; $G*$ over $\gamma_*(r, c(\bar{x}))$; \sqrt{f} over $\gamma_f(\bar{a}(\bar{x}))$; $D*'$ over $D(r * c(\bar{x}))$.

\neg RI)

$$\omega_{f'}(\bar{x}, r) = \cancel{\gamma_{\gamma_f}(\bar{x})} \wedge \cancel{\gamma_D(r)} \wedge [\cancel{\gamma_f(\bar{x})} \wedge D(r) \Rightarrow \cancel{\gamma_a(\bar{x})} \wedge [a(\bar{x}) \Rightarrow \cancel{\gamma_b(\bar{x})} \wedge \gamma_*(r, b(\bar{x}))]] \wedge [\neg a(\bar{x}) \Rightarrow \dots]$$

Annotations: \sqrt{f} over $\gamma_{\gamma_f}(\bar{x})$ and $\gamma_D(r)$; GD over $\gamma_f(\bar{x})$ and $D(r)$; \sqrt{f} over $\gamma_a(\bar{x})$; \sqrt{f} over $\gamma_b(\bar{x})$; $G*$ over $\gamma_*(r, b(\bar{x}))$; D_b from $D(r)$ to $D(b(\bar{x}))$; $\sqrt{f} \quad GDc \quad G* \quad D*'$ over the last part; "as with RI" under the last part.

QED

$\gamma_\beta(\bar{x}) \triangleq \gamma_f(\bar{x})$ — as before

$\vdash \boxed{\sqrt{\beta}}$ — as before, same proof

$\gamma_{\tilde{f}}(\bar{x}) \triangleq \gamma_f(\bar{x})$ — as before

$\vdash \boxed{\sqrt{\tilde{f}}}$ — as before, same proof (only LI case)

special case of a ground base value: everything is the same as before