Course 3: Nested and Mutual Algorithms in Coq

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

LORIA (Nancy), TU & WPI (Vienna), CNRS

PC'19, Herrsching, September 20, 2019

Introduction

- What is nested recursion (McCarthy's F91)

$$\texttt{let rec f91}\ n = \texttt{if}\ n > 100\ \texttt{then}\ n - 10$$
$$\texttt{else f91(f91(}n + 11\texttt{))}$$

- Why are nested (mutual) algorithms more difficult:
  - need semantic info. for termination
  - hard to separate correctness and termination
- Mutual can hyde nesting (Knuth's F91)

$$\begin{aligned}
&\texttt{let rec k91}\ x \ = \\
&\quad \texttt{if}\ x \leqslant a\ \texttt{then}\ \texttt{k91}^c(x + d)\ \texttt{else}\ x - b \\
&\texttt{and}\ \texttt{k91}^n\ x \ = \\
&\quad \texttt{if}\ n = 0\ \texttt{then}\ x\ \texttt{else}\ \texttt{k91}^{n-1}\ (f\ x)
\end{aligned}$$

- Another Example
  - Paulson's normalization

Cheating: `f91` for smarties

- Suppose we already now the full spec of `f91`

$$\begin{aligned}\mathbb{S}_{\mathtt{f91}}\ n\ r := {} & 100 < n \wedge r = n - 10 \\ \vee {} & n \leqslant 100 \wedge r = 91\end{aligned}$$

- And a decreasing measure: $n \mapsto 101 - n$
- Then extracting `f91` is easy:

```
Definition f91_sem_full n : {r | S_f91 n r}.
Proof.
  induction on n as f91 with measure (101 - n).
  ...
Defined.
```

- But are we that smart?

Methodically: `f91` for the dummies

$$\texttt{let rec f91}\ n = \texttt{if}\ n > 100\ \texttt{then}\ n - 10\ \texttt{else}\ \texttt{f91}(\texttt{f91}(n+11))$$

- The graph $\mathbb{G}_{\texttt{f91}} : \texttt{nat} \to \texttt{nat} \to \texttt{Prop}$ from the algo.

$$\frac{n > 100}{\mathbb{G}_{\texttt{f91}}\ n\ (n-10)} \qquad \frac{n \leqslant 100 \qquad \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \qquad \mathbb{G}_{\texttt{f91}}\ m\ r}{\mathbb{G}_{\texttt{f91}}\ n\ r}$$

Methodically: `f91` for the dummies

$$\texttt{let rec f91 } n = \texttt{if } n > 100 \texttt{ then } n - 10 \texttt{ else f91}(\texttt{f91}(n+11))$$

- The graph $\mathbb{G}_{\texttt{f91}} : \texttt{nat} \to \texttt{nat} \to \texttt{Prop}$ from the algo.

$$\frac{n > 100}{\mathbb{G}_{\texttt{f91}}\ n\ (n-10)} \qquad \frac{n \leqslant 100 \qquad \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \qquad \mathbb{G}_{\texttt{f91}}\ m\ r}{\mathbb{G}_{\texttt{f91}}\ n\ r}$$

- $\mathbb{G}_{\texttt{f91}}$ is functional (induction/inversion)

$$\forall n\, r_1\, r_2,\ \mathbb{G}_{\texttt{f91}}\ n\ r_1 \to \mathbb{G}_{\texttt{f91}}\ n\ r_2 \to r_1 = r_2$$

Methodically: `f91` for the dummies

$$\texttt{let rec f91}\ n = \texttt{if}\ n > 100\ \texttt{then}\ n - 10\ \texttt{else f91(f91(}n + 11\texttt{))}$$

- The graph $\mathbb{G}_{\texttt{f91}} : \texttt{nat} \to \texttt{nat} \to \texttt{Prop}$ from the algo.

$$\frac{n > 100}{\mathbb{G}_{\texttt{f91}}\ n\ (n-10)} \qquad \frac{n \leqslant 100 \qquad \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \qquad \mathbb{G}_{\texttt{f91}}\ m\ r}{\mathbb{G}_{\texttt{f91}}\ n\ r}$$

- $\mathbb{G}_{\texttt{f91}}$ is functional (induction/inversion)

$$\forall n\, r_1\, r_2,\ \mathbb{G}_{\texttt{f91}}\ n\ r_1 \to \mathbb{G}_{\texttt{f91}}\ n\ r_2 \to r_1 = r_2$$

- The domain $\mathbb{D}_{\texttt{f91}}$ as a bar predicate:

$$\frac{n > 100}{\mathbb{D}_{\texttt{f91}}\ n} \qquad \frac{n \leqslant 100 \qquad \mathbb{D}_{\texttt{f91}}\ (n+11) \qquad \forall m, \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \to \mathbb{D}_{\texttt{f91}}\ m}{\mathbb{D}_{\texttt{f91}}\ n}$$

Methodically: f91 for the dummies

$$\texttt{let rec f91}\ n = \texttt{if}\ n > 100\ \texttt{then}\ n - 10\ \texttt{else}\ \texttt{f91}(\texttt{f91}(n+11))$$

- The graph $\mathbb{G}_{\texttt{f91}} : \texttt{nat} \to \texttt{nat} \to \texttt{Prop}$ from the algo.

$$\frac{n > 100}{\mathbb{G}_{\texttt{f91}}\ n\ (n-10)} \qquad \frac{n \leqslant 100 \qquad \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \qquad \mathbb{G}_{\texttt{f91}}\ m\ r}{\mathbb{G}_{\texttt{f91}}\ n\ r}$$

- $\mathbb{G}_{\texttt{f91}}$ is functional (induction/inversion)

$$\forall n\, r_1\, r_2,\ \mathbb{G}_{\texttt{f91}}\ n\ r_1 \to \mathbb{G}_{\texttt{f91}}\ n\ r_2 \to r_1 = r_2$$

- The domain $\mathbb{D}_{\texttt{f91}}$ as a bar predicate:

$$\frac{n > 100}{\mathbb{D}_{\texttt{f91}}\ n} \qquad \frac{n \leqslant 100 \qquad \mathbb{D}_{\texttt{f91}}\ (n+11) \qquad \forall m, \mathbb{G}_{\texttt{f91}}\ (n+11)\ m \to \mathbb{D}_{\texttt{f91}}\ m}{\mathbb{D}_{\texttt{f91}}\ n}$$

- and then $\texttt{f91\_full} : \forall n,\ \mathbb{D}_{\texttt{f91}}\ n \to \{r \mid \mathbb{G}_{\texttt{f91}}\ n\ r\}$

$\texttt{f91\_full} : \forall n, \mathbb{D}_{\texttt{f91}}\ n \to \{r \mid \mathbb{G}_{\texttt{f91}}\ n\ r\}$

$$
\begin{array}{l}
\texttt{Let f91\_full} : \forall n, \mathbb{D}_{\texttt{f91}}\ n \to \{r \mid \mathbb{G}_{\texttt{f91}}\ n\ r\}. \\
\quad \texttt{refine (fix loop } n\ D_n\ \{\texttt{struct } D_n\} := \\
\quad\quad \texttt{match } 100 <_? n \texttt{ as } r \texttt{ return } 100 <_? n = r \to \_ \texttt{ with} \\
\quad\quad\quad \mid \texttt{true} \ \Rightarrow \texttt{fun } E \Rightarrow \texttt{exist } \_\ (n-10)\ \mathbb{G}_1^? \\
\quad\quad\quad \mid \texttt{false} \Rightarrow \texttt{fun } E \Rightarrow \texttt{let } (f_1, H_1) := \texttt{loop } (n+11)\ \mathbb{T}_1^? \texttt{ in} \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \texttt{let } (f_2, H_2) := \texttt{loop } f_1\ \mathbb{T}_2^? \\
\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad \texttt{in exist } \_\ f_2\ \mathbb{G}_2^? \\
\quad \texttt{end eq\_refl}). \\
\texttt{Proof.} \\
\quad \text{provide proofs for } \mathbb{T}_1^?, \mathbb{T}_2^?, \mathbb{G}_1^?, \mathbb{G}_2^? \\
\texttt{Qed.}
\end{array}
$$

$$\mathbb{G}_1^? // \ldots, D_n : \mathbb{D}_{\texttt{f91}}\ n, E : 100 <_? n = \texttt{true} \vdash \mathbb{G}_{\texttt{f91}}\ n\ (n-10)$$

$$\mathbb{T}_1^? // \ldots, D_n : \mathbb{D}_{\texttt{f91}}\ n, E : 100 <_? n = \texttt{false} \vdash \mathbb{D}_{\texttt{f91}}\ (n+11)$$

$$\ldots$$

Induction/Recursion for f91

- From `f91_full` we simulate the IR scheme

$$
\begin{array}{l}
\texttt{Inductive}\ \mathbb{D}_{\texttt{f91}} : \texttt{nat} \to \texttt{Prop} := \\
\quad \mid \texttt{d\_f91\_0} : \forall n,\ 100 < n \to \mathbb{D}_{\texttt{f91}}\ n \\
\quad \mid \texttt{d\_f91\_1} : \forall n,\ n \leqslant 100 \to \forall D, \mathbb{D}_{\texttt{f91}}\ (\texttt{f91}\ (n+11)\ D) \\
\qquad\qquad\qquad\qquad\qquad\quad \to \mathbb{D}_{\texttt{f91}}\ n \\
\texttt{with Fixpoint f91}\ n\ (D : \mathbb{D}_{\texttt{f91}}\ n) := \\
\ \texttt{match}\ D\ \texttt{with} \\
\quad \mid \texttt{d\_f91\_0}\ n\ H_n \qquad\qquad \Rightarrow n - 10 \\
\quad \mid \texttt{d\_f91\_1}\ n\ H_n\ D_1\ D_2 \quad \Rightarrow \texttt{f91}\ (\texttt{f91}\ (n+11)\ D_1)\ D_2 \\
\ \texttt{end.}
\end{array}
$$

Induction/Recursion for f91

- From `f91_full` we simulate the IR scheme

$$
\begin{array}{l}
\texttt{Inductive}\ \mathbb{D}_{\texttt{f91}} : \texttt{nat} \to \texttt{Prop} := \\
\quad \mid \texttt{d\_f91\_0} : \forall n,\ 100 < n \to \mathbb{D}_{\texttt{f91}}\ n \\
\quad \mid \texttt{d\_f91\_1} : \forall n,\ n \leqslant 100 \to \forall D, \mathbb{D}_{\texttt{f91}}\ (\texttt{f91}\ (n+11)\ D) \\
\qquad\qquad\qquad\qquad\qquad\quad \to \mathbb{D}_{\texttt{f91}}\ n \\
\texttt{with Fixpoint f91}\ n\ (D : \mathbb{D}_{\texttt{f91}}\ n) := \\
\ \ \texttt{match}\ D\ \texttt{with} \\
\quad \mid \texttt{d\_f91\_0}\ n\ H_n \qquad\qquad \Rightarrow n - 10 \\
\quad \mid \texttt{d\_f91\_1}\ n\ H_n\ D_1\ D_2 \ \ \Rightarrow \texttt{f91}\ (\texttt{f91}\ (n+11)\ D_1)\ D_2 \\
\ \ \texttt{end.}
\end{array}
$$

- constructors, fixpoint equations

Induction/Recursion for f91

- From f91_full we simulate the IR scheme

```
Inductive 𝔻f91 : nat → Prop :=
   | d_f91_0 : ∀n, 100 < n → 𝔻f91 n
   | d_f91_1 : ∀n, n ⩽ 100 → ∀D, 𝔻f91 (f91 (n + 11) D)
                              → 𝔻f91 n
with Fixpoint f91 n (D : 𝔻f91 n) :=
  match D with
   | d_f91_0 n Hn          ⇒ n − 10
   | d_f91_1 n Hn D1 D2    ⇒ f91 (f91 (n + 11) D1) D2
  end.
```

- constructors, fixpoint equations
- proof irrelevance (PIRR) : $\texttt{f91}\ n\ D_n = \texttt{f91}\ n\ D'_n$

Induction/Recursion for f91

- From `f91_full` we simulate the IR scheme

$$
\begin{array}{l}
\texttt{Inductive } \mathbb{D}_{\texttt{f91}} : \texttt{nat} \to \texttt{Prop} := \\
\quad \mid \texttt{d\_f91\_0} : \forall n,\ 100 < n \to \mathbb{D}_{\texttt{f91}}\ n \\
\quad \mid \texttt{d\_f91\_1} : \forall n,\ n \leqslant 100 \to \forall D, \mathbb{D}_{\texttt{f91}}\ (\texttt{f91}\ (n+11)\ D) \\
\qquad\qquad\qquad\qquad\qquad\quad \to \mathbb{D}_{\texttt{f91}}\ n \\
\texttt{with Fixpoint f91 } n\ (D : \mathbb{D}_{\texttt{f91}}\ n) := \\
\ \ \texttt{match } D \texttt{ with} \\
\quad \mid \texttt{d\_f91\_0}\ n\ H_n \qquad\quad \Rightarrow n - 10 \\
\quad \mid \texttt{d\_f91\_1}\ n\ H_n\ D_1\ D_2 \ \ \Rightarrow \texttt{f91}\ (\texttt{f91}\ (n+11)\ D_1)\ D_2 \\
\ \ \texttt{end.}
\end{array}
$$

- constructors, fixpoint equations
- proof irrelevance (PIRR) : $\texttt{f91}\ n\ D_n = \texttt{f91}\ n\ D'_n$
- a Type-bounded dependent elimination scheme
  - for PIRR predicates $P : \forall n, \mathbb{D}_{\texttt{f91}}\ n \to \texttt{Type}$

$$HP_{\text{pirr}} : \forall n\, D_n\, D'_n,\ P\ \ n\ \ D_n \to P\ \ n\ \ D'_n$$

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Study `f91` as a partial function

- First, partial correctness by induction (PIRR):

$$\forall n\,(D_n : \mathbb{D}_{\texttt{f91}}\ n),\ \mathbb{S}_{\texttt{f91}}\ n\ (\texttt{f91}\ n\ D_n)$$

Study `f91` as a partial function

- First, partial correctness by induction (PIRR):

$$\forall n\,(D_n : \mathbb{D}_{\mathtt{f91}}\ n),\ \mathbb{S}_{\mathtt{f91}}\ n\ (\mathtt{f91}\ n\ D_n)$$

- then totality/termination:
  - with constructors for $\mathbb{D}_{\mathtt{f91}}$
  - by induction on $101 - n$
  - we get $\mathtt{f91\_terminates} : \forall n, \mathbb{D}_{\mathtt{f91}}\ n$

Study `f91` as a partial function

- First, partial correctness by induction (PIRR):

$$\forall n\,(D_n : \mathbb{D}_{\texttt{f91}}\ n),\ \mathbb{S}_{\texttt{f91}}\ n\ (\texttt{f91}\ n\ D_n)$$

- then totality/termination:
  - with constructors for $\mathbb{D}_{\texttt{f91}}$
  - by induction on $101 - n$
  - we get `f91_terminates` : $\forall n,\ \mathbb{D}_{\texttt{f91}}\ n$
- total fun. as `fun` $n \Rightarrow$ `f91` n (`f91_terminates` n)

Study `f91` as a partial function

- First, partial correctness by induction (PIRR):

$$\forall n\,(D_n : \mathbb{D}_{\texttt{f91}}\ n),\ \mathbb{S}_{\texttt{f91}}\ n\ (\texttt{f91}\ n\ D_n)$$

- then totality/termination:
  - with constructors for $\mathbb{D}_{\texttt{f91}}$
  - by induction on $101 - n$
  - we get $\texttt{f91\_terminates} : \forall n, \mathbb{D}_{\texttt{f91}}\ n$
- total fun. as $\texttt{fun}\ n \Rightarrow\ \texttt{f91}\ n\ (\texttt{f91\_terminates}\ n)$
- Extraction as intended
  - with `nat` as unary/Peano natural numbers...

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Larry Paulson's normalization (1985)

$$
\begin{array}{l}
\texttt{let rec nm } e = \texttt{match } e \texttt{ with} \\
\quad \begin{array}{|ll}
\alpha & \to \alpha \\
\omega(\alpha, y, z) & \to \omega(\alpha, \texttt{nm } y, \texttt{nm } z) \\
\omega(\omega(a, b, c), y, z) & \to \texttt{nm}(\omega(a, \texttt{nm}(\omega(b, y, z)), \\
 & \qquad\qquad \texttt{nm}(\omega(c, y, z))))
\end{array}
\end{array}
$$

- Expressions in $\Omega : b, x, y ::= \alpha \mid \omega\ b\ x\ y$
  - α is atomic expression
  - $\omega\ b\ x\ y$ denotes "if b then x else y"

Larry Paulson's normalization (1985)

$$
\begin{array}{l}
\texttt{let rec nm } e = \texttt{match } e \texttt{ with} \\
\quad \begin{array}{lll}
\mid \alpha & \to & \alpha \\
\mid \omega(\alpha, y, z) & \to & \omega(\alpha, \texttt{nm } y, \texttt{nm } z) \\
\mid \omega(\omega(a, b, c), y, z) & \to & \texttt{nm}(\omega(a, \texttt{nm}(\omega(b, y, z)), \\
 & & \qquad\qquad \texttt{nm}(\omega(c, y, z))))
\end{array}
\end{array}
$$

- Expressions in $\Omega : b, x, y ::= \alpha \mid \omega\ b\ x\ y$
  - α is atomic expression
  - $\omega\ b\ x\ y$ denotes "$\texttt{if}\ b\ \texttt{then}\ x\ \texttt{else}\ y$"
- Interest of this algorithm:
  - recurring example (Giesl 97, B&C 05...)
  - has nested recursion but still compact
  - idealized but meaningful

Inductive capture of $\mathbb{D} : \Omega \to \texttt{Prop}$

- Using the graph $\mathbb{G} : \Omega \to \Omega \to \texttt{Prop}$

$$\frac{}{\mathbb{G}\ \alpha\ \alpha} \qquad \frac{\mathbb{G}\ y\ n_y \qquad\qquad \mathbb{G}\ z\ n_z}{\mathbb{G}\ (\omega\ \alpha\ y\ z)\ (\omega\ \alpha\ n_y\ n_z)}$$

$$\frac{\mathbb{G}\ (\omega\ b\ y\ z)\ n_b \qquad \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \qquad \mathbb{G}\ (\omega\ a\ n_b\ n_c)\ n_a}{\mathbb{G}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)\ n_a}$$

Inductive capture of $\mathbb{D} : \Omega \to \mathtt{Prop}$

- Using the graph $\mathbb{G} : \Omega \to \Omega \to \mathtt{Prop}$

$$\frac{}{\mathbb{G}\ \alpha\ \alpha} \qquad \frac{\mathbb{G}\ y\ n_y \qquad \mathbb{G}\ z\ n_z}{\mathbb{G}\ (\omega\ \alpha\ y\ z)\ (\omega\ \alpha\ n_y\ n_z)}$$

$$\frac{\mathbb{G}\ (\omega\ b\ y\ z)\ n_b \qquad \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \qquad \mathbb{G}\ (\omega\ a\ n_b\ n_c)\ n_a}{\mathbb{G}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)\ n_a}$$

- Define $\mathbb{D} = \mathtt{fun}\ e \mapsto \exists n, \mathbb{G}\ e\ n$ inductively by:

$$\frac{}{\mathbb{D}\ \alpha} \qquad \frac{\mathbb{D}\ y \qquad \mathbb{D}\ z}{\mathbb{D}\ (\omega\ \alpha\ y\ z)}$$

$$\frac{\begin{array}{c}\mathbb{D}\ (\omega\ b\ y\ z) \qquad \mathbb{D}\ (\omega\ c\ y\ z) \\ \forall n_b\, n_c, \mathbb{G}\ (\omega\ b\ y\ z)\ n_b \to \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \to \mathbb{D}\ (\omega\ a\ n_b\ n_c)\end{array}}{\mathbb{D}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)}$$

Inductive capture of $\mathbb{D} : \Omega \to \texttt{Prop}$

- Using the graph $\mathbb{G} : \Omega \to \Omega \to \texttt{Prop}$

$$\frac{}{\mathbb{G}\ \alpha\ \alpha} \qquad \frac{\mathbb{G}\ y\ n_y \qquad \mathbb{G}\ z\ n_z}{\mathbb{G}\ (\omega\ \alpha\ y\ z)\ (\omega\ \alpha\ n_y\ n_z)}$$

$$\frac{\mathbb{G}\ (\omega\ b\ y\ z)\ n_b \qquad \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \qquad \mathbb{G}\ (\omega\ a\ n_b\ n_c)\ n_a}{\mathbb{G}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)\ n_a}$$

- Define $\mathbb{D} = \texttt{fun}\ e \mapsto \exists n, \mathbb{G}\ e\ n$ inductively by:

$$\frac{}{\mathbb{D}\ \alpha} \qquad \frac{\mathbb{D}\ y \qquad \mathbb{D}\ z}{\mathbb{D}\ (\omega\ \alpha\ y\ z)}$$

$$\frac{\begin{array}{c}\mathbb{D}\ (\omega\ b\ y\ z) \qquad \mathbb{D}\ (\omega\ c\ y\ z) \\ \forall n_b\, n_c, \mathbb{G}\ (\omega\ b\ y\ z)\ n_b \to \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \to \mathbb{D}\ (\omega\ a\ n_b\ n_c)\end{array}}{\mathbb{D}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)}$$

- Define `nm_full` by structural induction on $D_e : \mathbb{D}\ e$

Def. of nm_full : $\forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}$

$$
\begin{array}{l}
\texttt{Let}\ \ \texttt{nm\_full} : \forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}. \\
\texttt{refine(fix loop}\ e\ D_e\ \{\texttt{struct}\ D_e\} := \\
\quad \texttt{match}\ e\ \texttt{as}\ e'\ \texttt{return}\ \mathbb{D}\ e' \to \{n \mid \mathbb{G}\ e'\ n\}\ \texttt{with} \\
\quad\quad \begin{array}{lll}
\mid \alpha & \Rightarrow \texttt{fun}\ D \Rightarrow & \quad\ \texttt{exist}\ \_\ \alpha\ \mathbb{G}_0^? \\
\mid \omega\ \alpha\ y\ z & \Rightarrow \texttt{fun}\ D \Rightarrow & \texttt{let}\ (n_y, H_y) := \texttt{loop}\ y\ \mathbb{T}_y^?\ \texttt{in} \\
 & & \texttt{let}\ (n_z, H_z) := \texttt{loop}\ z\ \mathbb{T}_z^? \\
 & & \texttt{in}\ \ \texttt{exist}\ \_\ (\omega\ \alpha\ n_y\ n_z)\ \mathbb{G}_1^? \\
\mid \omega\ (\omega\ a\ b\ c)\ y\ z & \Rightarrow \texttt{fun}\ D \Rightarrow & \texttt{let}\ (n_b, H_b) := \texttt{loop}\ (\omega\ b\ y\ z)\ \mathbb{T}_b^?\ \texttt{in} \\
 & & \texttt{let}\ (n_c, H_c) := \texttt{loop}\ (\omega\ c\ y\ z)\ \mathbb{T}_c^?\ \texttt{in} \\
 & & \texttt{let}\ (n_a, H_a) := \texttt{loop}\ (\omega\ a\ n_b\ n_c)\ \mathbb{T}_a^? \\
 & & \texttt{in}\ \ \texttt{exist}\ \_\ n_a\ \mathbb{G}_2^?
\end{array} \\
\quad \texttt{end}\ D_e);\ \texttt{simpl in}\ *. \\
\texttt{Proof.}\ \ \text{of certificates}\ \mathbb{T}_y^?, \mathbb{T}_z^?, \mathbb{T}_b^?, \mathbb{T}_c^?, \mathbb{T}_a^?\ \text{and post-conditions}\ \mathbb{G}_0^?, \mathbb{G}_1^?, \mathbb{G}_2^?\ \ \texttt{Qed.}
\end{array}
$$

Def. of nm_full : $\forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}$

```
Let  nm_full : ∀e, 𝔻 e → {n | 𝔾 e n}.
refine(fix loop e Dₑ {struct Dₑ} :=
 match e as e' return 𝔻 e' → {n | 𝔾 e' n} with
   | α               ⇒ fun D ⇒      exist _ α 𝔾₀?
   | ω α y z         ⇒ fun D ⇒ let (n_y, H_y) := loop y 𝕋_y? in
                               let (n_z, H_z) := loop z 𝕋_z?
                               in  exist _ (ω α n_y n_z) 𝔾₁?
   | ω (ω a b c) y z ⇒ fun D ⇒ let (n_b, H_b) := loop (ω b y z) 𝕋_b? in
                               let (n_c, H_c) := loop (ω c y z) 𝕋_c? in
                               let (n_a, H_a) := loop (ω a n_b n_c) 𝕋_a?
                               in  exist _ n_a 𝔾₂?
 end Dₑ); simpl in ∗.
```

Proof. of certificates $\mathbb{T}^?_y, \mathbb{T}^?_z, \mathbb{T}^?_b, \mathbb{T}^?_c, \mathbb{T}^?_a$ and post-conditions $\mathbb{G}^?_0, \mathbb{G}^?_1, \mathbb{G}^?_2$ Qed.

- use of dependent pattern matching

Def. of `nm_full` : $\forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}$

$$
\begin{array}{l}
\texttt{Let}\ \ \texttt{nm\_full} : \forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}. \\
\texttt{refine(fix loop}\ e\ D_e\ \{\texttt{struct}\ D_e\} := \\
\quad \texttt{match}\ e\ \texttt{as}\ e'\ \texttt{return}\ \mathbb{D}\ e' \to \{n \mid \mathbb{G}\ e'\ n\}\ \texttt{with} \\
\quad\quad \begin{array}{lll}
\mid \alpha & \Rightarrow \texttt{fun}\ D \Rightarrow & \quad\ \texttt{exist}\ \_\ \alpha\ \mathbb{G}_0^? \\
\mid \omega\ \alpha\ y\ z & \Rightarrow \texttt{fun}\ D \Rightarrow & \texttt{let}\ (n_y, H_y) := \texttt{loop}\ y\ \mathbb{T}_y^?\ \texttt{in} \\
 & & \texttt{let}\ (n_z, H_z) := \texttt{loop}\ z\ \mathbb{T}_z^? \\
 & & \texttt{in}\ \ \texttt{exist}\ \_\ (\omega\ \alpha\ n_y\ n_z)\ \mathbb{G}_1^? \\
\mid \omega\ (\omega\ a\ b\ c)\ y\ z & \Rightarrow \texttt{fun}\ D \Rightarrow & \texttt{let}\ (n_b, H_b) := \texttt{loop}\ (\omega\ b\ y\ z)\ \mathbb{T}_b^?\ \texttt{in} \\
 & & \texttt{let}\ (n_c, H_c) := \texttt{loop}\ (\omega\ c\ y\ z)\ \mathbb{T}_c^?\ \texttt{in} \\
 & & \texttt{let}\ (n_a, H_a) := \texttt{loop}\ (\omega\ a\ n_b\ n_c)\ \mathbb{T}_a^? \\
 & & \texttt{in}\ \ \texttt{exist}\ \_\ n_a\ \mathbb{G}_2^?
\end{array} \\
\quad \texttt{end}\ D_e);\ \texttt{simpl in}\ *. \\
\texttt{Proof.}\ \ \text{of certificates}\ \mathbb{T}_y^?, \mathbb{T}_z^?, \mathbb{T}_b^?, \mathbb{T}_c^?, \mathbb{T}_a^?\ \text{and post-conditions}\ \mathbb{G}_0^?, \mathbb{G}_1^?, \mathbb{G}_2^?\ \ \texttt{Qed.}
\end{array}
$$

- use of dependent pattern matching
- LC (i.e. proof obligations) separated from CC

Def. of nm_full : $\forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}$

$$
\begin{array}{l}
\texttt{Let nm\_full} : \forall e, \mathbb{D}\ e \to \{n \mid \mathbb{G}\ e\ n\}. \\
\texttt{refine(fix loop}\ e\ D_e\ \{\texttt{struct}\ D_e\} := \\
\quad \texttt{match}\ e\ \texttt{as}\ e'\ \texttt{return}\ \mathbb{D}\ e' \to \{n \mid \mathbb{G}\ e'\ n\}\ \texttt{with} \\
\quad\quad \mid \alpha \quad \Rightarrow \texttt{fun}\ D \Rightarrow \quad \texttt{exist}\ \_\ \alpha\ \mathbb{G}^?_0 \\
\quad\quad \mid \omega\ \alpha\ y\ z \quad \Rightarrow \texttt{fun}\ D \Rightarrow \texttt{let}\ (n_y, H_y) := \texttt{loop}\ y\ \mathbb{T}^?_y\ \texttt{in} \\
\quad\quad\quad\quad \texttt{let}\ (n_z, H_z) := \texttt{loop}\ z\ \mathbb{T}^?_z \\
\quad\quad\quad\quad \texttt{in}\ \ \texttt{exist}\ \_\ (\omega\ \alpha\ n_y\ n_z)\ \mathbb{G}^?_1 \\
\quad\quad \mid \omega\ (\omega\ a\ b\ c)\ y\ z \Rightarrow \texttt{fun}\ D \Rightarrow \texttt{let}\ (n_b, H_b) := \texttt{loop}\ (\omega\ b\ y\ z)\ \mathbb{T}^?_b\ \texttt{in} \\
\quad\quad\quad\quad \texttt{let}\ (n_c, H_c) := \texttt{loop}\ (\omega\ c\ y\ z)\ \mathbb{T}^?_c\ \texttt{in} \\
\quad\quad\quad\quad \texttt{let}\ (n_a, H_a) := \texttt{loop}\ (\omega\ a\ n_b\ n_c)\ \mathbb{T}^?_a \\
\quad\quad\quad\quad \texttt{in}\ \ \texttt{exist}\ \_\ n_a\ \mathbb{G}^?_2 \\
\quad \texttt{end}\ D_e);\ \texttt{simpl in}\ *. \\
\texttt{Proof.}\ \text{of certificates}\ \mathbb{T}^?_y, \mathbb{T}^?_z, \mathbb{T}^?_b, \mathbb{T}^?_c, \mathbb{T}^?_a\ \text{and post-conditions}\ \mathbb{G}^?_0, \mathbb{G}^?_1, \mathbb{G}^?_2\ \ \texttt{Qed.}
\end{array}
$$

- use of dependent pattern matching
- LC (i.e. proof obligations) separated from CC
- LC divided: termination certificates, post-conditions

Introduction
McCarthy's F91
Wise
Dumb
f91_full
IR scheme
Properties
Paulson's normalisation
Inductive domain
nm_full
Logical content
IR scheme
Extraction
Term&correct. postponed
Partial correct. of nm
Totality of nm

Proof obligations (Logical Content)

- Post-conditions by the constructors of $\mathbb{G}$

$$
\begin{array}{l}
\mathbb{G}^?_0 \; // \; \ldots \vdash \mathbb{G}\ \alpha\ \alpha \\
\mathbb{G}^?_1 \; // \; \ldots, H_y : \mathbb{G}\ y\ n_y, H_z : \mathbb{G}\ z\ n_z \vdash \mathbb{G}\ (\omega\ \alpha\ y\ z)\ (\omega\ \alpha\ n_y\ n_z) \\
\mathbb{G}^?_2 \; // \; \ldots, H_b : \mathbb{G}\ (\omega\ b\ y\ z)\ n_b, H_c : \mathbb{G}\ (\omega\ c\ y\ z)\ n_c, \ldots \\
\qquad\qquad \ldots H_a : \mathbb{G}\ (\omega\ a\ n_b\ n_c)\ n_a \vdash \mathbb{G}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)\ n_a
\end{array}
$$

Proof obligations (Logical Content)

- Post-conditions by the constructors of $\mathbb{G}$

$$
\begin{array}{l}
\mathbb{G}_0^? \; // \; \ldots \vdash \mathbb{G}\ \alpha\ \alpha \\
\mathbb{G}_1^? \; // \; \ldots, H_y : \mathbb{G}\ y\ n_y, H_z : \mathbb{G}\ z\ n_z \vdash \mathbb{G}\ (\omega\ \alpha\ y\ z)\ (\omega\ \alpha\ n_y\ n_z) \\
\mathbb{G}_2^? \; // \; \ldots, H_b : \mathbb{G}\ (\omega\ b\ y\ z)\ n_b, H_c : \mathbb{G}\ (\omega\ c\ y\ z)\ n_c, \ldots \\
\qquad \ldots H_a : \mathbb{G}\ (\omega\ a\ n_b\ n_c)\ n_a \vdash \mathbb{G}\ (\omega\ (\omega\ a\ b\ c)\ y\ z)\ n_a
\end{array}
$$

- Termination certificates

$$
\begin{array}{l}
\mathbb{T}_y^? \; // \; \ldots, D : \mathbb{D}\ (\omega\ \alpha\ y\ z) \vdash \mathbb{D}\ y \\
\mathbb{T}_b^? \; // \; \ldots, D : \mathbb{D}\ (\omega\ (\omega\ a\ b\ c)\ y\ z) \vdash \mathbb{D}\ (\omega\ b\ y\ z) \\
\mathbb{T}_a^? \; // \; \ldots, D : \mathbb{D}\ (\omega\ (\omega\ a\ b\ c)\ y\ z), H_b : \mathbb{G}\ (\omega\ b\ y\ z)\ n_b, \ldots \\
\qquad \ldots H_c : \mathbb{G}\ (\omega\ c\ y\ z)\ n_c \vdash \mathbb{D}\ (\omega\ a\ n_b\ n_c)
\end{array}
$$

- beware of $\boxed{\text{structural decrease}}$ in term. certificates
  - by the `inversion` tactic
  - or "small inversion" (J.F. Monin)

Simulated IR scheme

$$
\begin{array}{lll}
\texttt{Inductive}\ \ \mathbb{D} : \Omega \to \boxed{\texttt{Prop}} := & & \\
\quad \mid \texttt{d\_nm\_0} & : & \mathbb{D}\ \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z & : & \mathbb{D}\ y \to \mathbb{D}\ z \to \mathbb{D}(\omega\ \alpha\ y\ z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c & : & \mathbb{D}\big(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & & \qquad\quad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)\big) \\
 & \to & \mathbb{D}\big(\omega\ (\omega\ a\ b\ c)\ y\ z\big) \\
\texttt{with Fixpoint}\ \ \texttt{nm}\ e\ (D_e : \mathbb{D}\ e) : \Omega := \texttt{match}\ D_e\ \texttt{with} & & \\
\quad \mid \texttt{d\_nm\_0} & \mapsto & \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z\ D_y\ D_z & \mapsto & \omega\ \alpha\ (\texttt{nm}\ y\ D_y)\ (\texttt{nm}\ z\ D_z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c\ D_a & \mapsto & \texttt{nm}\ \big(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & & \qquad\quad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)\big)\ D_a \\
\ \texttt{end}. & &
\end{array}
$$

Simulated IR scheme

$$
\begin{array}{ll}
\texttt{Inductive}\ \ \mathbb{D} : \Omega \to \boxed{\texttt{Prop}} := & \\
\quad \mid \texttt{d\_nm\_0} & : \ \mathbb{D}\ \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z & : \ \mathbb{D}\ y \to \mathbb{D}\ z \to \mathbb{D}(\omega\ \alpha\ y\ z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c & : \ \mathbb{D}(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & \qquad\qquad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)) \\
 & \to \mathbb{D}(\omega\ (\omega\ a\ b\ c)\ y\ z) \\
\texttt{with Fixpoint}\ \ \texttt{nm}\ e\ (D_e : \mathbb{D}\ e) : \Omega := \texttt{match}\ D_e\ \texttt{with} & \\
\quad \mid \texttt{d\_nm\_0} & \mapsto \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z\ D_y\ D_z & \mapsto \omega\ \alpha\ (\texttt{nm}\ y\ D_y)\ (\texttt{nm}\ z\ D_z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c\ D_a & \mapsto \texttt{nm}\ \big(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & \qquad\qquad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)\big)\ D_a \\
\quad \texttt{end.} &
\end{array}
$$

- The domain $\mathbb{D} : \Omega \to \texttt{Prop}$ is **non-informative**

Simulated IR scheme

$$
\begin{array}{l}
\texttt{Inductive}\ \ \mathbb{D} : \Omega \to \boxed{\texttt{Prop}} := \\
\quad \mid \texttt{d\_nm\_0} \qquad\qquad\qquad\qquad : \mathbb{D}\ \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z \qquad\qquad\qquad : \mathbb{D}\ y \to \mathbb{D}\ z \to \mathbb{D}(\omega\ \alpha\ y\ z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c \quad : \mathbb{D}(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)) \\
\qquad\qquad\qquad\qquad\qquad\qquad \to \mathbb{D}(\omega\ (\omega\ a\ b\ c)\ y\ z) \\
\texttt{with Fixpoint}\ \ \texttt{nm}\ e\ (D_e : \mathbb{D}\ e) : \Omega := \texttt{match}\ D_e\ \texttt{with} \\
\quad \mid \texttt{d\_nm\_0} \qquad\qquad\qquad\qquad \mapsto \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z\ D_y\ D_z \qquad\quad \mapsto \omega\ \alpha\ (\texttt{nm}\ y\ D_y)\ (\texttt{nm}\ z\ D_z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c\ D_a \mapsto \texttt{nm}\ (\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
\qquad\qquad\qquad\qquad\qquad\qquad\qquad\quad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c))\ D_a \\
\ \texttt{end.}
\end{array}
$$

- The domain $\mathbb{D} : \Omega \to \texttt{Prop}$ is **non-informative**
- $\texttt{nm} : \forall e, \mathbb{D}\ e \to \Omega$ is **proof-irrelevant**, i.e. $\texttt{nm}\ x\ D_1 = \texttt{nm}\ x\ D_2$

Simulated IR scheme

$$
\begin{array}{ll}
\texttt{Inductive}\ \mathbb{D} : \Omega \to \boxed{\texttt{Prop}} := & \\
\quad \mid \texttt{d\_nm\_0} & : \mathbb{D}\ \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z & : \mathbb{D}\ y \to \mathbb{D}\ z \to \mathbb{D}(\omega\ \alpha\ y\ z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c & : \mathbb{D}(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & \qquad\qquad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)) \\
 & \to \mathbb{D}(\omega\ (\omega\ a\ b\ c)\ y\ z) \\
\texttt{with Fixpoint}\ \texttt{nm}\ e\ (D_e : \mathbb{D}\ e) : \Omega := \texttt{match}\ D_e\ \texttt{with} & \\
\quad \mid \texttt{d\_nm\_0} & \mapsto \alpha \\
\quad \mid \texttt{d\_nm\_1}\ y\ z\ D_y\ D_z & \mapsto \omega\ \alpha\ (\texttt{nm}\ y\ D_y)\ (\texttt{nm}\ z\ D_z) \\
\quad \mid \texttt{d\_nm\_2}\ a\ b\ c\ y\ z\ D_b\ D_c\ D_a & \mapsto \texttt{nm}\ \big(\omega\ a\ (\texttt{nm}\ (\omega\ b\ y\ z)\ D_b) \\
 & \qquad\qquad (\texttt{nm}\ (\omega\ c\ y\ z)\ D_c)\big)\ D_a \\
\quad \texttt{end.} &
\end{array}
$$

- The domain $\mathbb{D} : \Omega \to \texttt{Prop}$ is **non-informative**
- $\texttt{nm} : \forall e, \mathbb{D}\ e \to \Omega$ is **proof-irrelevant**, i.e. $\texttt{nm}\ x\ D_1 = \texttt{nm}\ x\ D_2$
- Constructors, dep. elim. scheme and fixpoint equations *retrieved*

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Extraction independent of the domain

- In $\texttt{nm}\ e\ (D_e : \mathbb{D}\ e)$ extract. erases $D_e : \mathbb{D}\ e : \texttt{Prop}$

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Extraction independent of the domain

- In $\texttt{nm}\ e\ (D_e : \mathbb{D}\ e)$ extract. erases $D_e : \mathbb{D}\ e : \texttt{Prop}$
- Hence `Extraction nm` gives the intended term:

$$
\begin{array}{l}
\texttt{let rec nm}\ e = \texttt{match}\ e\ \texttt{with} \\
\quad \mid \alpha \qquad\qquad\quad \to \alpha \\
\quad \mid \omega(x,y,z) \quad\ \ \to \texttt{match}\ x\ \texttt{with} \\
\qquad \mid \alpha \qquad\qquad \to \omega(\alpha, \texttt{nm}\ y, \texttt{nm}\ z) \\
\qquad \mid \omega(a,b,c) \to \texttt{nm}(\omega(a, \texttt{nm}(\omega(b,y,z)), \texttt{nm}(\omega(c,y,z))))
\end{array}
$$

Extraction independent of the domain

- In $\mathtt{nm}\ e\ (D_e : \mathbb{D}\ e)$ extract. erases $D_e : \mathbb{D}\ e : \mathtt{Prop}$
- Hence `Extraction nm` gives the intended term:

$$
\begin{array}{l}
\mathtt{let\ rec\ nm}\ e = \mathtt{match}\ e\ \mathtt{with} \\
\quad \begin{array}{|ll}
\alpha & \to \alpha \\
\omega(x,y,z) & \to \mathtt{match}\ x\ \mathtt{with} \\
\quad \begin{array}{|l} \alpha \\ \omega(a,b,c) \end{array} & \begin{array}{l} \to \omega(\alpha, \mathtt{nm}\ y, \mathtt{nm}\ z) \\ \to \mathtt{nm}(\omega(a, \mathtt{nm}(\omega(b,y,z)), \mathtt{nm}(\omega(c,y,z)))) \end{array}
\end{array}
\end{array}
$$

- The proof term $D_e : \mathbb{D}\ e$
  - has **no impact** on extracted algorithm

Extraction independent of the domain

- In $\mathtt{nm}\ e\ (D_e : \mathbb{D}\ e)$ extract. erases $D_e : \mathbb{D}\ e : \mathtt{Prop}$
- Hence `Extraction nm` gives the intended term:

$$
\begin{array}{l}
\mathtt{let\ rec\ nm}\ e = \mathtt{match}\ e\ \mathtt{with} \\
\quad \mid \alpha \qquad\qquad\quad \to \alpha \\
\quad \mid \omega(x,y,z) \quad\ \to \mathtt{match}\ x\ \mathtt{with} \\
\qquad \mid \alpha \qquad\quad\ \ \to \omega(\alpha, \mathtt{nm}\ y, \mathtt{nm}\ z) \\
\qquad \mid \omega(a,b,c) \to \mathtt{nm}(\omega(a, \mathtt{nm}(\omega(b,y,z)), \mathtt{nm}(\omega(c,y,z))))
\end{array}
$$

- The proof term $D_e : \mathbb{D}\ e$
  - has **no impact** on extracted algorithm
  - great complexity does not matter

Extraction independent of the domain

- In $\mathtt{nm}\ e\ (D_e : \mathbb{D}\ e)$ extract. erases $D_e : \mathbb{D}\ e : \mathtt{Prop}$
- Hence $\mathtt{Extraction\ nm}$ gives the intended term:

$$
\begin{array}{l}
\mathtt{let\ rec\ nm}\ e = \mathtt{match}\ e\ \mathtt{with} \\
\quad \begin{array}{lll}
\mid \alpha & & \to \alpha \\
\mid \omega(x,y,z) & & \to \mathtt{match}\ x\ \mathtt{with} \\
& \mid \alpha & \to \omega(\alpha, \mathtt{nm}\ y, \mathtt{nm}\ z) \\
& \mid \omega(a,b,c) & \to \mathtt{nm}(\omega(a, \mathtt{nm}(\omega(b,y,z)), \mathtt{nm}(\omega(c,y,z))))
\end{array}
\end{array}
$$

- The proof term $D_e : \mathbb{D}\ e$
  - has **no impact** on extracted algorithm
  - great complexity does not matter
  - use high-level tool (lex. prod, WQOs)

Termination postponed after definition

- Proving termination of `nm` at e is a term $D_e : \mathbb{D}\ e$
  - a "meaningful" characterization of $\mathbb{D}\ e$
  - for partial fun.: $P : \Omega \to \texttt{Prop}$ and $P \subseteq \mathbb{D}$
  - for total functions: a proof of $\forall e, \mathbb{D}\ e$

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Termination postponed after definition

- Proving termination of $\mathtt{nm}$ at e is a term $D_e : \mathbb{D}\ e$
  - a "meaningful" characterization of $\mathbb{D}\ e$
  - for partial fun.: $P : \Omega \to \mathtt{Prop}$ and $P \subseteq \mathbb{D}$
  - for total functions: a proof of $\forall e, \mathbb{D}\ e$
- The proof of $P \subseteq \mathbb{D}$ can be provided:
  - after $\mathbb{D} : \Omega \to \mathtt{Prop}$ and $\mathtt{nm} : \forall e,\ \mathbb{D}\ e \to \Omega$ are def'd
  - by any means necessary
  - w/o consequences on extracted code
  - including by adding axioms (if necessary)

Termination postponed after definition

- Proving termination of $\texttt{nm}$ at e is a term $D_e : \mathbb{D}\ e$
  - a "meaningful" characterization of $\mathbb{D}\ e$
  - for partial fun.: $P : \Omega \to \texttt{Prop}$ and $P \subseteq \mathbb{D}$
  - for total functions: a proof of $\forall e, \mathbb{D}\ e$
- The proof of $P \subseteq \mathbb{D}$ can be provided:
  - after $\mathbb{D} : \Omega \to \texttt{Prop}$ and $\texttt{nm} : \forall e,\ \mathbb{D}\ e \to \Omega$ are def'd
  - by any means necessary
  - w/o consequences on extracted code
  - including by adding axioms (if necessary)
- Tools from IR:
  - constructors
  - fixpoint equations

Partial correction postponed after def.

- Partial correction = higher-level charac. of $\texttt{nm}\ e\ D_e$

Partial correction postponed after def.

- Partial correction $=$ higher-level charac. of $\mathtt{nm}\ e\ D_e$
  - another spec/post-condition
  - by induction on $\mathbb{G}\ e\ (\mathtt{nm}\ e\ D_e)$
  - or using dependent elimination on (e, D_e) (IR)

Partial correction postponed after def.

- Partial correction = higher-level charac. of $\texttt{nm}\ e\ D_e$
  - another spec/post-condition
  - by induction on $\mathbb{G}\ e\ (\texttt{nm}\ e\ D_e)$
  - or using dependent elimination on (e, D_e) (IR)
- Partial correction: for meaningful $\mathbb{S}$
  - $\forall e\ (D_e : \mathbb{D}\ e), \mathbb{S}\ e\ (\texttt{nm}\ e\ D_e)$

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Partial correction postponed after def.

- Partial correction = higher-level charac. of $\mathtt{nm}\ e\ D_e$
  - another spec/post-condition
  - by induction on $\mathbb{G}\ e\ (\mathtt{nm}\ e\ D_e)$
  - or using dependent elimination on (e, D_e) (IR)
- Partial correction: for meaningful $\mathbb{S}$
  - $\forall e\ (D_e : \mathbb{D}\ e), \mathbb{S}\ e\ (\mathtt{nm}\ e\ D_e)$
- Tools from IR:
  - dependent elimination
  - fixpoint equations

Partial correction of `nm` on $\mathbb{D}$

- dep. elim. `d_nm_rect` for partial correction

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Partial correction of nm on $\mathbb{D}$

- dep. elim. d_nm_rect for partial correction
- nm_normal : $\forall\, e\ (D_e : \mathbb{D}\ e)$, normal (nm e D_e)
  - the shape $\omega\ (\omega\ \_\ \_\ \_)\ \_\ \_$ is forbidden

Partial correction of $\texttt{nm}$ on $\mathbb{D}$

- dep. elim. `d_nm_rect` for partial correction
- $\texttt{nm\_normal} : \forall\, e\, (D_e : \mathbb{D}\ e),\ \texttt{normal}\,(\texttt{nm}\ e\ D_e)$
  - the shape $\omega\ (\omega\ \_\ \_\ \_)\ \_\ \_$ is forbidden
- $\texttt{nm\_equiv} : \forall\, e\, (D_e : \mathbb{D}\ e),\ e \simeq_\Omega \texttt{nm}\ e\ D_e$
  - the normal form is computationaly equiv.

Partial correction of $\mathtt{nm}$ on $\mathbb{D}$

- dep. elim. `d_nm_rect` for partial correction
- $\mathtt{nm\_normal} : \forall\, e\ (D_e : \mathbb{D}\ e),\ \mathtt{normal}\,(\mathtt{nm}\ e\ D_e)$
  - the shape $\omega\ (\omega\ \_\ \_\ \_)\ \_\ \_$ is forbidden
- $\mathtt{nm\_equiv} : \forall\, e\ (D_e : \mathbb{D}\ e),\ e \simeq_\Omega \mathtt{nm}\ e\ D_e$
  - the normal form is computationaly equiv.
- $\mathtt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e),\ |\mathtt{nm}\ e\ D_e| \leqslant |e|$
  - some "size" $|\cdot| : \Omega \to \mathtt{nat}$ is preserved (Giesl 97)

$$|\alpha| = 1 \qquad |\omega\ x\ y\ z| = |x| \cdot (1 + |y| + |z|)$$

Totality of $\mathbb{D}$ / Termination of $\mathtt{nm}$

$$\mathtt{d\_nm\_total} : \forall e,\ \mathbb{D}\ e$$

- By induction on the size $|e|$

Dominique Larchey-Wendling
https://github.com/DmxLarchey/PC19

Totality of $\mathbb{D}$ / Termination of $\texttt{nm}$

$$\texttt{d\_nm\_total} : \forall e,\ \mathbb{D}\ e$$

- By induction on the size $|e|$
  - we use $\texttt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e),\ |\texttt{nm}\ e\ D_e| \leqslant |e|$

Totality of $\mathbb{D}$ / Termination of `nm`

$$\texttt{d\_nm\_total} : \forall e,\ \mathbb{D}\ e$$

- By induction on the size $|e|$
  - we use $\texttt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e),\ |\texttt{nm}\ e\ D_e| \leqslant |e|$
  - and $|\omega\ x\ y\ z| \leqslant |\omega\ x'\ y'\ z'|$ (monotonic)
    - i.e. when $|x| \leqslant |x'|$, $|y| \leqslant |y'|$, $|z| \leqslant |z'|$

Totality of $\mathbb{D}$ / Termination of $\texttt{nm}$

$$\texttt{d\_nm\_total} : \forall e,\ \mathbb{D}\ e$$

- By induction on the size $|e|$
  - we use $\texttt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e),\ |\texttt{nm}\ e\ D_e| \leqslant |e|$
  - and $|\omega\ x\ y\ z| \leqslant |\omega\ x'\ y'\ z'|$ (monotonic)
    - i.e. when $|x| \leqslant |x'|$, $|y| \leqslant |y'|$, $|z| \leqslant |z'|$
  - and $|\omega\ u\ y\ z| < |\omega\ v\ y\ z|$ when $|u| < |v|$

Totality of $\mathbb{D}$ / Termination of nm

$$\texttt{d\_nm\_total} : \forall e, \mathbb{D}\ e$$

- By induction on the size $|e|$
  - we use $\texttt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e), |\texttt{nm}\ e\ D_e| \leqslant |e|$
  - and $|\omega\ x\ y\ z| \leqslant |\omega\ x'\ y'\ z'|$ (monotonic)
    - i.e. when $|x| \leqslant |x'|$, $|y| \leqslant |y'|$, $|z| \leqslant |z'|$
  - and $|\omega\ u\ y\ z| < |\omega\ v\ y\ z|$ when $|u| < |v|$
  - and $|y| < |\omega\ x\ y\ z|$ and $|z| < |\omega\ x\ y\ z|$
  - & $|\omega\ a\ (\omega\ b\ y\ z)\ (\omega\ c\ y\ z)| < |\omega\ (\omega\ a\ b\ c)\ y\ z|$

Totality of $\mathbb{D}$ / Termination of nm

$$\texttt{d\_nm\_total} : \forall e,\ \mathbb{D}\ e$$

- By induction on the size $|e|$
  - we use $\texttt{nm\_dec} : \forall\, e\ (D_e : \mathbb{D}\ e),\ |\texttt{nm}\ e\ D_e| \leqslant |e|$
  - and $|\omega\ x\ y\ z| \leqslant |\omega\ x'\ y'\ z'|$ (monotonic)
    - i.e. when $|x| \leqslant |x'|$, $|y| \leqslant |y'|$, $|z| \leqslant |z'|$
  - and $|\omega\ u\ y\ z| < |\omega\ v\ y\ z|$ when $|u| < |v|$
  - and $|y| < |\omega\ x\ y\ z|$ and $|z| < |\omega\ x\ y\ z|$
  - & $|\omega\ a\ (\omega\ b\ y\ z)\ (\omega\ c\ y\ z)| < |\omega\ (\omega\ a\ b\ c)\ y\ z|$
- Partial correction / termination indep. of definition

$$\boxed{\texttt{paulson\_nm} : \forall e : \Omega,\ \{n_e : \Omega \mid e \simeq_\Omega n_e \wedge \texttt{normal}\ e\}}$$