APL: Sheet 6

We have lots of new things to play with this week! Effects :D

Is everyone okay with the concept of effects? I found them vague for the longest time, but basically

effects = impure operations
e.g. IO or state

CBN vs. CBV

Since I am forgetful, especially with initialisms I'll put this here so I won't forget:

CBN = variables are terms
CBV = variables are values

① Best aquaint ourselves with our new toys ⇒ time for some trees and reductions

By this point in the course, if you struggle with derivation trees or reductions PLEASE let us know.

The term e is well typed.

Now in this derivation, we are not given a type to verify, we must work out the correct type. Luckily inversion helps with this. I will leave the ty abstract, then bind it when I find out.

We are also given no env, so must show well-typedness for empty env.

P.T.O

$\tau = Num$
$\sigma = Num \to Num$
$\Gamma = \{f : Num \to Num\}$
$\sigma_2 = Num$

I like to list my bindings/unknowns here

Makes it easy to see what still needs resolved.

$$
\dfrac{
  \dfrac{}{\Gamma \vdash f : \sigma_2 \to \tau}\,\text{VAR}
  \qquad
  \dfrac{
    \dfrac{}{\Gamma \vdash f : Num \to Num}\,\text{VAR}
    \qquad
    \dfrac{2 \in \mathbb{N}}{\Gamma \vdash num[2] : Num}\,\text{Num}
  }{\Gamma \vdash f(num[2]) : \tau}\,\text{APP}
}{
  \dfrac{\Gamma \vdash f(f(num[2])) : \tau}{\vdash \lambda f : Num \to Num . f(f(num[2])) : \sigma \to \tau}\,\text{LAM}
}\,\text{APP}
$$

$$
\dfrac{
  \dfrac{\vdash \lambda f : Num \to Num . f(f(num[2])) : \sigma \to \tau}{\vdash e_1 : \sigma \to \tau}\,\text{def. } e_1
  \qquad
  \dfrac{\vdash print('na'; u) : \sigma}{\vdash e_2 : \sigma}\,\text{def } e_2
}{
  \dfrac{\vdash e_1(e_2) : \tau}{\vdash e : \tau}\,\text{def. } e
}\,\text{APP}
$$

:: See sudden

sudden:

$\Gamma_2 = x : Num$

$$
\dfrac{
  'batman' \in \Sigma^*
  \qquad
  \dfrac{
    \dfrac{}{\Gamma \vdash x : Num}\,\text{VAR}
    \qquad
    \dfrac{1 \in \mathbb{N}}{\Gamma \vdash num[1] : Num}\,\text{Num}
  }{\Gamma \vdash plus(x; num[1]) : Num}\,\text{PLUS}
}{\Gamma \vdash print('batman'; plus(x; num[1])) : Num}\,\text{PRINT}
$$

$$
\dfrac{\Gamma \vdash print('batman'; plus(x; num[1])) : Num}{\vdash \lambda x : Num . print('batman'; plus(x; num[1])) : Num \to Num}\,\text{LAM}
$$

$$
\dfrac{
  'na' \in \Sigma^*
  \qquad
  \dfrac{\vdash \lambda x : Num . print('batman'; plus(x; num[1])) : Num \to Num}{\vdash u : Num \to Num}\,\text{def. } u
}{\vdash print('na'; u) : Num \to Num}\,\text{PRINT}
$$

new rule! exciting!

CBN reduction:

This is what we have been doing so far in the course.

$e_1(e_2)$

$\equiv$ def e_1

$(\lambda f:\mathrm{Num}\to\mathrm{Num}.\, f(f(\mathrm{num}[2])))\, e_2$

$\mapsto$ D-BETA

$f(f(\mathrm{num}[2]))\,[e_2/f]$

$\equiv$ def subst

$e_2(e_2(\mathrm{num}[2]))$

$\equiv$ def e_2

$\mathrm{print}(\text{'na'}; u)(\mathrm{print}(\text{'na'}; u)(\mathrm{num}[2]))$

$\mapsto$ D-P-PRINT

last week lots of you were mislabelling these. $\mapsto$ ONLY when you use a dynamics elimination

'na'

$u(\mathrm{print}(\text{'na'}; u)(\mathrm{num}[2]))$

$\equiv$ def u

$(\lambda x:\mathrm{Num}.\, \mathrm{print}(\text{'batman'}; \mathrm{plus}(x; \mathrm{num}[1])))$
$(\mathrm{print}(\text{'na'}; u)(\mathrm{num}[2]))$

$\mapsto$ D-BETA

$\mathrm{print}(\text{'batman'}; \mathrm{plus}((\mathrm{print}(\text{'na'}; u)(\mathrm{num}[2])); \mathrm{num}[1]))$

$\mapsto$ D-P-PRINT

'batman'

$\mathrm{plus}((\mathrm{print}(\text{'na'}; u)(\mathrm{num}[2])); \mathrm{num}[1])$

$\mapsto$ D-PLUS-1 / D-APP-1 / ~~D-P-PRINT~~

'na'

$\mathrm{plus}(u(\mathrm{num}[2]); \mathrm{num}[1])$

$\equiv$ def u

$\mathrm{plus}((\lambda x:\mathrm{Num}.\, \mathrm{print}(\text{'batman'}; \mathrm{plus}(x; \mathrm{num}[1])))(\mathrm{num}[2])$
$; \mathrm{num}[1])$

$\mapsto$ D-PLUS-1 / BETA

$\mathrm{plus}(\mathrm{print}(\text{'batman'}; \mathrm{plus}(\mathrm{num}[2]; \mathrm{num}[1]))$
$; \mathrm{num}[1])$

$\mapsto$ D-PLUS-1 / D-P-PRINT

'batman'

$\mathrm{plus}(\mathrm{plus}(\mathrm{num}[2]; \mathrm{num}[1]); \mathrm{num}[1])$

$\mapsto$ D-PLUS-1 / D-PLUS (2+1=3)

$\mathrm{plus}(\mathrm{num}[3]; \mathrm{num}[1])$

$\mapsto$ D-PLUS (3+1=4)

$\mathrm{num}[4]$

$\Rightarrow$ result = 4
effect =
na batman
na batman

when in doubt, check

$$\dfrac{}{\text{print}(\text{'na'}, u) \xrightarrow{\text{'na'}} u}\ \text{D-P-PRINT}$$

$$\dfrac{}{\text{print}(\text{'na'}; u)(\text{num}[2]) \xrightarrow{\text{'na'}} u(\text{num}[2])}\ \text{D-APP-1}$$

$$\dfrac{}{\text{plus}((\text{print}(\text{'na'}; u)(\text{num}[2])); \text{num}[1])}\ \text{D-PLUS-1}$$

$$\xrightarrow{\text{'na'}}$$

$$\text{plus}(u(\text{num}[2]); \text{num}[1])$$

CBV reduction: (new)

I'd expect the effect to be different.

$e_1(e_2)$

This time, I can only use D-BETA when the arg is a val, so I am forced to do work on e_2.

$\equiv$ def. e_2 which I can only do after e_1 is a val, which it is

$e_1(\text{print}(\text{'na'}; u))$

$\mapsto$ D-APP-2/D-LAM/PRINT by VAL-LAM

'na'

$e_1(u)$

$\equiv$ def. e_1

$(\lambda f:\text{Num} \to \text{Num}.\, f(f(\text{num}[2])))(u)$

$\equiv$ def u

$(\lambda f:\text{Num} \to \text{Num}.\, f(f(\text{num}[2])))$
$(\lambda x:\text{Num}.\, \text{print}(\text{'batman'}; \text{plus}(x; \text{num}[1])))$

This time our arg is a val (by VAL-LAM)

$\mapsto$ D-BETA

$(\lambda x:\text{Num}.\, \text{print}(\text{'batman'}; \text{plus}(x; \text{num}[1])))$
$((\lambda x:\text{Num}.\, \text{print}(\text{'batman'}; \text{plus}(x; \text{num}[1])))$
$(\text{num}[2]))$

↦ D-APP-2 / D-BETA

(λx:Num. print('batman'; plus (x; num[1])))
(print('batman'; plus (num[2]; num[1])))

↦ D-APP-2 / D-P-PRINT

'batman'

(λx:Num. print('batman'; plus (x; num[1])))
(plus (num[2]; num[1]))

↦ D-APP-2 / PLUS (2+1=3)

(λx:Num. print('batman'; plus (x; num[1])))
(num[3]

finally, it is a value so we can contract

↦ D-BETA (num[3] val)

print('batman'; plus (num[3]; num[1]))

↦

'batman'

plus (num[3]; num[1])

ah and now we see our effects have differed: we have 'nabatmanbatman' and no more prints) :o

↦ PLUS

num[4] but of course the result is unaffected

⇒ result = 4
effect = nabatmanbatman

② "Do we have to prove progress / preservation?" is certainly the question on everyones lips, but maybe not for the right reasons lol.

Well, lets think about what has changed: the dynamics

⇒ If the proof relies on dynamics we need to make some adjustment.

Preservation

If $\vdash e : \tau$ and $e \mapsto e'$ then $\vdash e' : \tau$

Now, this is a proof by induction on the dynamics ($\mapsto$) ⇒ the proof would need redoing with the new rules and adjustments to the rule. Having said that, we can certainly use the old proof to our advantage.

It must be proved for the new cases:

D-INL
D-INR
D-APP-2

The changed rules also need to be checked to see if the adjustments check the proof.

Progress

If $\vdash e : \tau$ then either e val or $e \mapsto e'$ for some e'

This is a proof by induction on the statics, so the structure won't change, but we need to conclude either a value or a transition, so our conclusions may need adjusted.

Some transition conclusions may need adjusted.

Algol

① We need to show derivability

A reminder:

Derivable means that the derivation of the rule ends in its premise

Hint says to do them in order so lets go!

$$\dfrac{① \ \vdots}{\Gamma \vdash_\Sigma m_1 \ ok} \quad \leftarrow \text{what we have} \rightarrow \quad \dfrac{\vdots \ ②}{\Gamma, x{:}Nat \vdash_\Sigma m_2 \ ok}$$

$$\dfrac{\vdots}{\Gamma \vdash_\Sigma \{x \leftarrow m_1 ; m_2\} \ ok} \quad \leftarrow \text{what we need to do}$$

Process for this is just writing the goal at the bottom of the page and iteratively applying rules till hopefully we hit ① or ②.

Oh cool this is de notation!

We'll need to do some desugaring before we play the derivation tree game

$$\dfrac{\dfrac{\dfrac{\vdots \ ①}{\Gamma \vdash_\Sigma m_1 \ ok}}{\Gamma \vdash_\Sigma cmd(m_1) : Cmd}\,\text{CMD} \qquad \dfrac{\vdots \ ②}{\Gamma, x{:}Nat \vdash_\Sigma m_2 \ ok}}{\Gamma \vdash_\Sigma bnd(cmd(m_1) ; x.m_2)}\,\text{BIND}$$

$$\equiv \text{def} \equiv$$

$$\Gamma \vdash_\Sigma \{x \leftarrow m_1 ; m_2\} \ ok$$

Well that was fun and pretty mechanical (hopefully my highlighting makes that clear).

The rest seem similar, except you must remember that we have our previous proofs and weakening at our disposal. I'll do these live if there is demand.