1 Overview

Shorthand for code .
Path = https://github.com/cchiw
Exs = *Path*/latte/
DATm = *Path*/DATm
Diderot_Dev = *Path*/Diderot-Dev
Vis15 = vis15

Shorthand for Text .
[Doc] = *Exs*/writeup/paper.pdf
[dissertation] = Chiw'17 dissertation
[AST paper] = Chiw'17 ICSE-AST paper

1.1 New and extended syntax

1.1.1 Composition

Functionality: Field Composition
Syntax: "compose" and "$\circ$"
field#k(d_1)[α]$\times$ field#k(d_0)[d_1] $\rightarrow$ field#k(d_1)[α]
Branch: *Diderot_Dev*
Text: EIN IR design, rewriting rules, and resolved bugs listed in [Doc]
Issues: none
Examples: Check out programs in *Exs*/fn_composition
- *function name syntax* can use "compose" (B*/observ.diderot)
- *unicode syntax* can use "$\circ$"(X1/observ.diderot,*Exs*/fn_composition/X2/t.diderot)
- *chains composition operator* can apply composition operator to other operators and to itself (X2/*.diderot)
- *solved bugs* Copies of the programs with solved bugs are in *Path*/B*

DATm command: python3 cte.py 1 36

1.1.2 Concatenation

Functionality: Concatenation
Syntax: "concat()"
field#k(d)[α] $\times$ field#k(d)[α] $\rightarrow$ field#k(d)[2,α]
Branch: *Diderot_Dev*
Text: Future Work chapter in [dissertation]. Details provided in [Doc]
Issues: None
Examples: Check out programs in *Exs*/fn_concatenation
DATm command: python3 cte.py 1 37
Future work Use syntax "[","]"

1.1.3 Matrix Inverse

Functionality: Matrix Inverse
Syntax: "inv()"
field#k(d)[i,j] $\rightarrow$ field#k(d)[i,j]
tensor[i,j] $\rightarrow$ tensor[i,j]
Branch: *Diderot_Dev*
Text: Design chapter in [dissertation]
Issues: None
Examples: Check out programs in *Exs*/fn_matrixInverse
DATm command: python3 cte.py 1 8

1.1.4 Swap and Selection

Functionality: Field assignment based on integer selector
Syntax: "swap()"
fty = field#k(d)[α]
int $\times$ fty $\times$ fty $\rightarrow$ fty
int $\times$ fty $\times$ fty$\times$ .... $\rightarrow$ fty

int × fty × fty × fty × fty × fty × fty → fty
Selection-id The first argument is an integer that serves to select a field. *i.e.* id=1 chooses the first field argument
Field Arguments The function accepts 2-6 field arguments.
Branch: *Diderot_Dev*
Text: none
Issues: Selection id is clamped when outside range (*i.e.* id=-6 → id=1) instead of throwing an error
Examples: Check out programs in *Exs*/fn_selection

- *Set Selection id* in Update method (X2)
- *Set Selection id* in Global Initialization (X1)

Future work: Hard-coded code generation needs to be more general to support n field arguments

1.1.5 Clerp and Clamp

Functionality: Clerp. Clamp and Lerp all in one
Syntax: "clerp()"
tensor[i] × tensor[i] × real → tensor[i]
tensor[i] × tensor[i] × real× real× real → tensor[i]
Branch: *Diderot_Dev* & *Vis15*
Text: none
Issues: none
Examples: Check out *Exs*/fn_clerp/clerp3.diderot

Functionality: Apply clamp to arbituary-sized tensors
Syntax: "clamp()"
tty = tensor[α]
tty × tty × tty → tty
Branch: *Diderot_Dev* & *Vis15*
Text: none
Issues: none
Examples: Check out *Exs*/fn_clerp/clamp.diderot

1.1.6 Max and Min

Functionality: Maximal and Minimum between fields
Syntax: MaxF(), MinF()
fty = field#k(d)[α]
fty × fty → fty
Branch: *Diderot_Dev*
Text: none
Issues: syntax "Max()" instead of "MaxF()
Examples: Check out programs in *Exs*/fn_min_max

1.1.7 Other

Unsupported functionality:

- radial basis functions
- absolute value
- sign (positive/negative)
- if statement

1.2 Tools

1.2.1 DATm

Use: **Test operators on and between tensors/image data based on correctness**
Tool: *DATm*:Diderot's Automated Testing tool
Branch: Any
Text: Testing chapter in [dissertation],[AST paper]
Issues: Needs a tutorial, but in lieu of one I added a Q& A
Q & A: See [Doc]