

Formulating Functional Formalisms for Fun

Brent Yorgey

Haskell Symposium
6 September 2024

~~Formulating Functional Formalisms for Fun~~ The Joy of Building (Functional) Languages

Brent Yorgey

Haskell Symposium
6 September 2024

1999

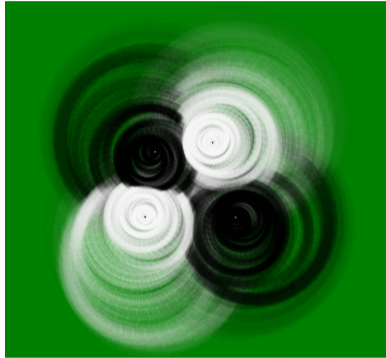
1. setlang





More History

2. Context Free Art



Chalk, by chris, CC-BY 3.0

<https://www.contextfreeart.org/gallery/view.php?id=256>



master ▾

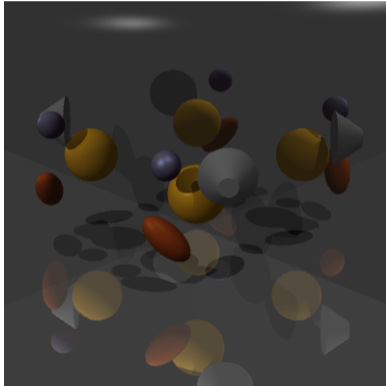
context-free / src-common / cfdgimpl.cpp

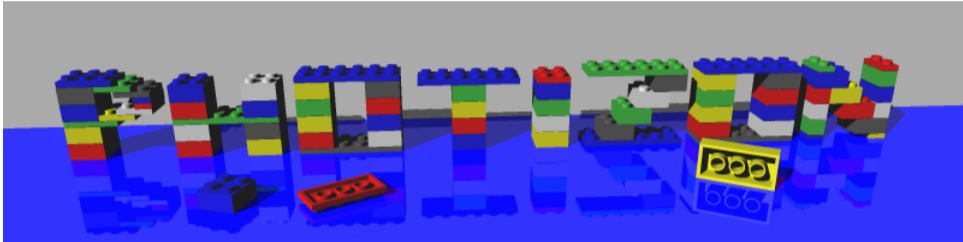
Code

Blame

```
406     }
407 }
408
409 void
410 CFDGImpl::rulesLoaded()
411 {
412     // thanks to by Brent Yorgey
413     std::vector<double> weightsums( m_shapeTypes.size(), 0.0 );
414     std::vector<double> percentweightsums( m_shapeTypes.size(), 0.0 );
415     std::vector<double> unitweightsums( m_shapeTypes.size(), 0.0 );
416     std::vector<int> rulecounts( m_shapeTypes.size(), 0 );
417     std::vector<int> weightTypes( m_shapeTypes.size(), 0 );
418
419     // first pass: sum all the weights for each shape type
420     for (auto&& r: mRules) {
421         if (r->weightType == ASTRule::PercentWeight) {
```

3. Photizon





4. c2bef

```
222p35*89+*11p>133p                >33g1+33p   22g33g- v>22g33g%#v_v
o                                     >|
2                                     v,,,,, ,,,,,.g22"is prime."<
1                                     >   v^                                     <
^_@#-g11g22p22+1g22,*25<,,,,,,,,,,,,,.g22"is not prime."<
```

2007



5. adventure

```
bear_hangout = location "Bear hangout" (  
  "You are standing at the end of a road.  A passage leads back to the " ++  
  "northwest." )  
  !!! [ SE -*-> cond  
    [ isHere bear ==> (  
      do msg ( "The bear is very annoyed that you would be " ++  
        "so presumptuous as to try and walk right " ++  
        "by it.  He tells you so by tearing your " ++  
        "head off." )  
      die  
    )  
    , always ==> stay  
  ]  
  , NW --> se_nw_road  
  , SW --> hidden_area[]  
]  
@@@ [ bear ]
```

6. species

```
> octopi = cycle `o` nonEmpty linOrd  
> take 10 $ labeled octopi  
[0,1,3,14,90,744,7560,91440,1285200,20603520]  
> take 10 $ unlabeled octopi  
[0,1,2,3,5,7,13,19,35,59]
```

7. GHC





















































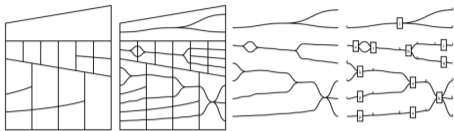
8. diagrams

diagrams

```
hilbert 0 = empty
hilbert n = hilbert' (n-1) # reflectY <> vrule 1
           <> hilbert (n-1) <> hrule 1
           <> hilbert (n-1) <> vrule (-1)
           <> hilbert' (n-1) # reflectX
where
  hilbert' m = hilbert m # rotateBy (1/4)

diagram :: Diagram B
diagram = strokeT (hilbert 6) # lc silver
           # opacity 0.3
```

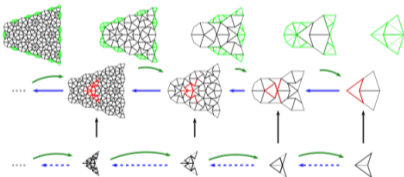
	Andy Gill andygill	✓2FA		Private	Member	2 teams	Orders	
	Mel Zuser bacchomato	✓2FA		Private	Member	2 teams	Orders	
	Brad Hardy brh29	✗2FA		Private	Member	1 team	Orders	
	Daniel Bergey bergey	✓2FA		Public	Owner	4 teams	Orders	
	Brent Yorgey byorgey	✓2FA		Public	Owner	7 teams	Orders	
	Christopher Chalmers rchalmers	✓2FA		Public	Member	2 teams	Orders	
	Deepak Jois deepakjois	✓2FA		Public	Member	1 team	Orders	
	Daniel Wagner drwtt	✓2FA		Private	Member	1 team	Orders	
	Ryan Yates ryy@rbrb	✓2FA		Public	Owner	5 teams	Orders	
	Gabor Greif ggreif	✓2FA		Private	Member	1 team	Orders	
	Niklas Haas haasn	✓2FA		Public	Member	2 teams	Orders	
	Hans Höglund haanshoglund	✓2FA		Public	Member	1 team	Orders	
	idontgetoutmuch	✓2FA		Public	Member	1 team	Orders	
	Igor Moreno igormoreno	✓2FA		Private	Member	1 team	Orders	
	Jan Bracker jbracker	✓2FA		Public	Member	2 teams	Orders	
	Jeffrey Rosenbluth jeffre Rosenbluth	✓2FA		Public	Owner	5 teams	Orders	
	John Lato john.lato	✗2FA		Private	Member	3 teams	Orders	
	Lally Singh laly	✓2FA		Private	Owner	0 teams	Orders	
	Tapesh Mandal mandatapesh	✗2FA		Private	Member	1 team	Orders	
	MathnerD14	✓2FA		Private	Member	3 teams	Orders	



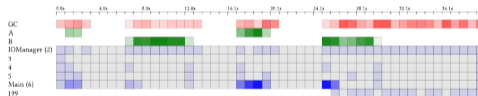
Celia Rubio-Madrigal, Jules Hedges



Dominic Steinitz



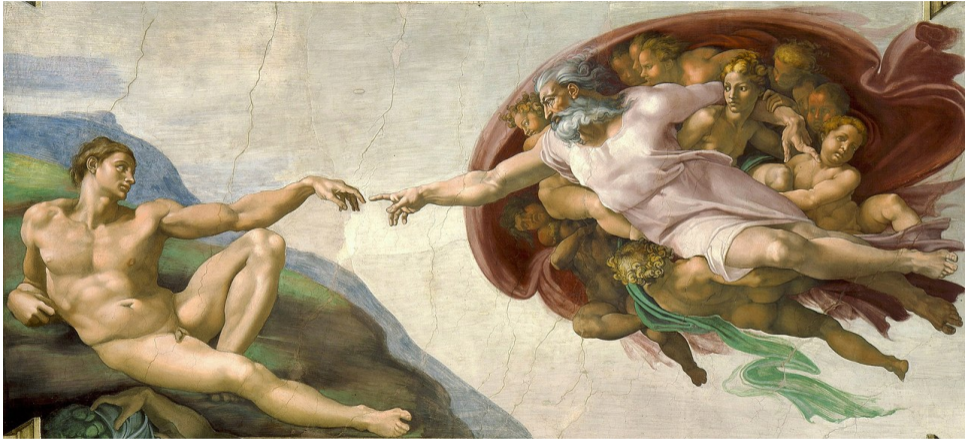
Chris Reade



Edsko de Vries, Duncan Coutts



Building languages and being human

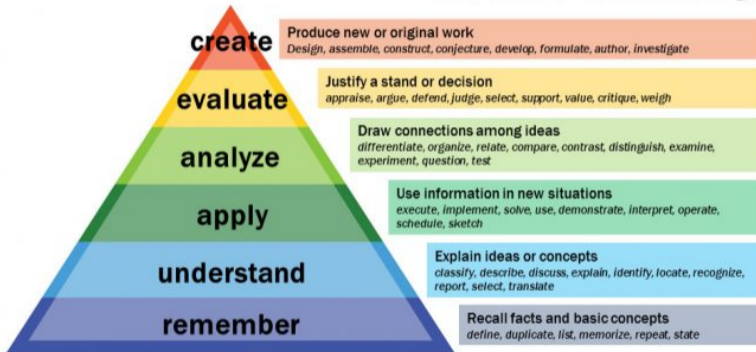


Creativity

Community

Learning

Bloom's Taxonomy



Creating

JOY

Community

Learning

9. Disco



```
||| An example to illustrate the kind of thing you are supposed to do
||| in the exercises below. We have defined the set using a *set
||| comprehension* so that it has the specified elements and the test
||| passes.
```

```
!!! example != {1, 4, 9, 16, 36} -- the test specifying the elements of 'example'
```

```
example : Set(N)
```

```
example = {x^2 | x in {1 .. 6}, x /= 5} -- a set comprehension defining it
```

```
-- Now you try.
```

```
!!! exA != {1, 3, 5, 7, 9, 11, 13, 15}
```

```
exA : Set(N)
```

```
exA = {} -- fill in a set comprehension here!
```

10. Swarm

The screenshot shows a game interface with a top-down view of a world. The world is composed of various terrain types and objects, represented by different colors and symbols. A command console is visible at the bottom, showing a list of commands and their results. The interface also includes a help menu and a status bar.

base (0, 2) Ω

Compendium

- big furnace 0
- big motor 0
- bit (0) 0
- bit (1) 0
- board 0
- boulder 0
- box 0
- branch 0
- camera 0

[F1] Help [F2] Robots [F4] Commands 0:06:dc.f (PAUSED)

0 2 4096 ticks / s

[^p] unpause [^o] step [M-d] debug [^zx] speed [M-.] hide REPL [M-h] hide robots
[!r] history [^c] cancel [M-p] pilot

cmd unit

```
> move; move
> def m2 = move;move end; def m4 = m2;m2 end
> m4
...
```

Machine [base#0]

```
(E. ▶move◀ ; move) ; force m2
```

A big furnace can get even hotter than a normal furnace, and can be used to process quartz into silicon.

Recipes

- big furnace
4 quartz — silicon
- big furnace
1 gold — gold 8

<https://swarm-game.github.io/>

11. Let's build a language together!

<https://github.com/byorgey/HS24-keynote>