RChain COOPERATIVE

The Space Calculus in context

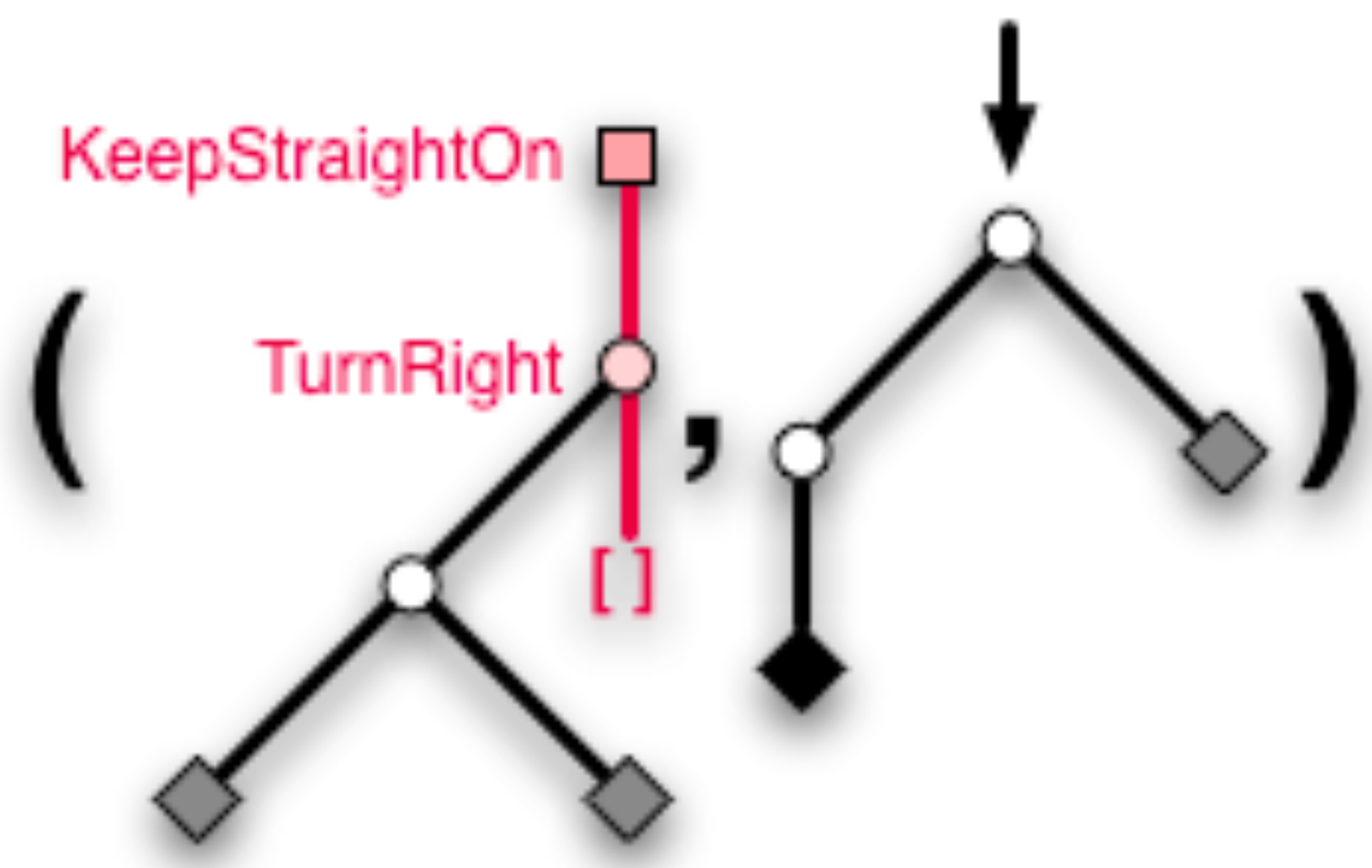
Lucius Gregory Meredith

RChain Research

Space-calculus in context

Background and motivations from maths and computer science

The notion of location as a splitting has a long history in mathematics.
From Dedekind cuts to Conway games this idea has born a great deal of fruit.
Huet's idea of the zipper should be seen in this light.



And it generalizes to a wide range of functors (data type constructors). Formally, the zipper-based notion of location, for a given data type T, is $\partial T \times T$ where ∂T is the type of 1-holed T-contexts.

Space-calculus in context

The domain equation

The idea is to apply this notion of location to a process calculus so that names really are locations.

We begin with the domain equation which generates the rho-calculus.

$$P[X] = 1 + X \times P[X] + X \times X \times P[X] + P[X] \times P[X] + X$$
$$RP = P[RP]$$

But now we want not quoted processes but locations, $\partial P[X] \times P[X]$

$$LP = P[\partial P[LP] \times P[LP]]$$

This was the equation i wrote down in 2008, but couldn't find a solution for until 2018

Space-calculus in context

Background and motivations from physics and chemistry

Oh, wait, why do we want an equation like this?

The intuition in the Einstein's field equations is that — when dynamics are taken into account — the metric is recursively intertwined with the stress-energy tensor.

$$R_{\mu\nu} - \frac{1}{2} R g_{\mu\nu} + \Lambda g_{\mu\nu} = \kappa T_{\mu\nu}.$$

We can get a discrete version of this idea using a process calculus where where the dynamics links location and synchronization.

To make this work, we need one additional ingredient, which we borrow from chemistry: the idea of a catalyst. So, a reduction can only happen in the presence of a catalyst.

In the calculus presented below, the enabling entity is more of a co-factor than a catalyst, but there are versions of the calculus where it is strictly a catalyst.

Space-calculus in context

Background and motivations from computer science, physics, and biology

```
P,Q ::= 0                   stopped process
     |  for( y <- x )P      input-guarded continuation
     |  x!( Q )             output
     |  P|Q                 parallel composition
     |  *x                  deref

x,y,… ::= @P

*y{@Q/y} = Q
```

COMM `for( x <- y )P | x!( Q ) -> P{@Q/y}`

COMM_K `for( x <- y )P | x!( Q ) -> P{@K[Q]/y}` Has applications to protocol interoperability and error correction

$\mathrm{COMM}_{K1} \circ \mathrm{COMM}_{K2}$ `for( x <- y )P | x!( Q ) -> P{@K2[K1[Q]]/y}` composes

But is static. The context is not programmable.

Space-calculus in context

The central proposal

```
P,Q ::= 0                      stopped process
     |  U( x )                 location update
     |  for( y <- x )P         input-guarded continuation
     |  x!( Q )                output
     |  P|Q                    parallel composition
     |  *x                     deref
     |  COMM( K )              situation catalyst

x,y ::= @<K,Q>
K   ::= [] | for( x <- y )K | x!( K ) | P|K
*y{@<K,Q>/y} = K[Q]

COMM( K ) | for( x <- y )P | x!( Q ) -> P{@<K,Q>/y}
U( x ) | *@<K,Q> -> COMM( K ) | x!( Q )
```

Space-calculus in context

A simple calculation

A simple calculation shows….

```
for(@<K,Q> <- @<[],0>)*@<K,Q> | @<[],0>!(P) | COMM(K1)
-> *@<K,Q>{@<K',P>/@<K,Q>} = K1[P]
```

If K1 is of the form

```
for(@<K,Q> <- @<[],0>)*@<K,Q> | @<[],0>!([]) | COMM(K2) | …
```

If K2 is of the form

```
for(@<K,Q> <- @<[],0>)*@<K,Q> | @<[],0>!([]) | COMM(K3) | …
```

then P moves through contexts K1, K2, K3, …

Space-calculus in context

Some desiderata and future work

While this represents a discrete version of the intuitions underlying GR it is decidedly not quantum. As far as my understanding goes, there are no non-local correlations.

However, we do have about 2/3 of an encoding of quantum mechanics in rho-calculus and so it is conceivable that this encoding could be composed with the techniques developed here.