Implementing the Solo Calculus

A REPL environment for evaluation of Solo Calculus expressions

Lassiter, Adam T.

Department of Computer Science
University of Bath

February 19, 2018

Rationale

- Recent decline in yearly single-core performance increase of processors has led to a need to think seriously about parallelism in even everyday applications.
- From a theoretical viewpoint, one may wish to exhibit useful properties for multiple processes, such as equivalence of computation.
- What does a parallel λ-calculus look like?

State of the Art

Solo Calculus

- A calculus of communicating mobile processes.
- Based on Fusion Calculus, based on π-calculus, inspired by λ-calculus...
  - All of these have been fairly well studied so far.
  - There exist interpreters for λ-calculus expressions, but not for any other.
  - The suitability of the alternatives for a teaching tool is poor due to the syntax being more complicated.

State of the Art

Solo Calculus

$$
\begin{array}{lcll}
\text{Solo } \alpha & := & u\,\tilde{x} & \text{(input)} \\
 & & \overline{u}\,\tilde{x} & \text{(output)} \\
 & & & \\
\text{Agent } P & := & 0 & \text{(inaction)} \\
 & & \alpha & \text{(solo)} \\
 & & Q \mid R & \text{(composition)} \\
 & & (x)\,Q & \text{(scope)} \\
 & & Q\{x/y\} & \text{(match)} \\
 & & !\,P & \text{(replication)}
\end{array}
$$

State of the Art

Solo Calculus

$$\overline{x}\, yz \mid !(uv)(x\, uv \mid \overline{u}\, v) \rightarrow$$

$$\overline{x}\, yz \mid (uv)(x\, uv \mid \overline{u}\, v) \mid !(uv)(x\, uv \mid \overline{u}\, v) \rightarrow$$

$$(uv)(\overline{u}\, v)\{y/u,\, z/v\} \mid !(uv)(x\, uv \mid \overline{u}\, v) \rightarrow$$

$$\overline{y}\, z \mid !(uv)(x\, uv \mid \overline{u}\, v)$$

State of the Art

Solo Diagrams

- A canonical representation of calculus expressions.
- There exists an obvious analog between these algebraic ‘diagrams’ and traditional graphical diagrams.
- Research led to easier calculation of replicated terms in calculi expressions.

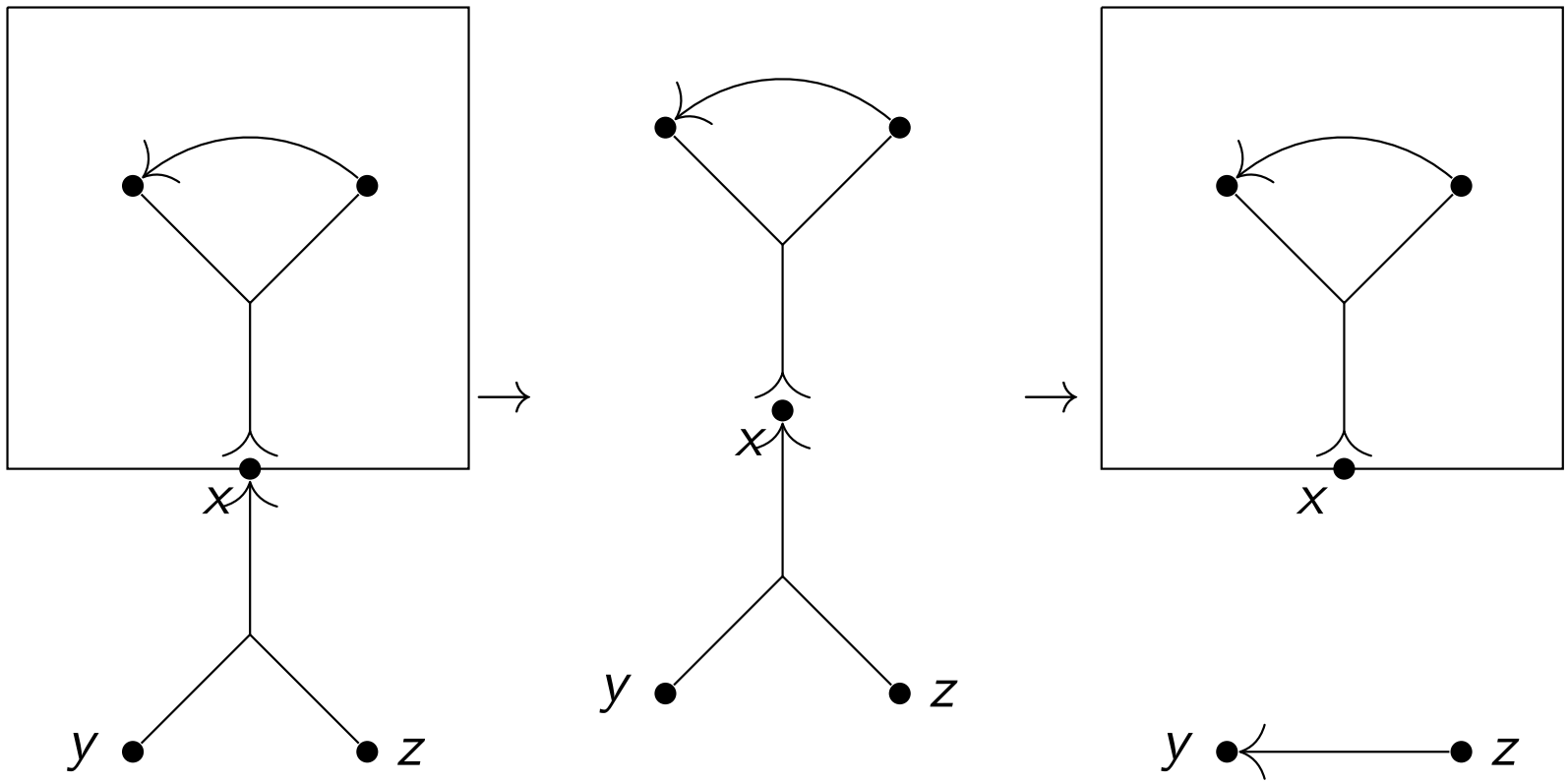
State of the Art

Solo Diagrams

Diagram	$\mathcal{D}$	$:=$	*triple* $(\mathcal{G}, \mathcal{M}, \mathcal{L})$
Graph	$\mathcal{G}$	$:=$	*multiset* $\{\mathcal{E}\}$
Boxes	$\mathcal{M}$	$:=$	*multiset* $\{\mathcal{B}\}$
Label	$\mathcal{L}$	$:=$	*map* $\mathcal{N} \in \mathcal{G} \mapsto \mathcal{N}$
Edge	$\mathcal{E}$	$:=$	*tuple* $(\mathcal{N}, \mathcal{N}_1 \ \ldots \mathcal{N}_n)_{s \in \{i,o\}}$
Box	$\mathcal{B}$	$:=$	*pair* $(\mathcal{G}, \mathit{set}\ \{\mathcal{N}\})$

State of the Art

Solo Diagrams



$\overline{x}\, yz \mid !(uv)(x\ uv \mid \overline{u}\ v) \longrightarrow \overline{x}\, yz \mid (uv)(x\ uv \mid \overline{u}\ v) \mid !(uv)(x\ uv \mid \overline{u}\ v) \longrightarrow \overline{y}\, z \mid !(uv)(x\ uv \mid \overline{u}\ v)$

Methods and Results

- Datatypes implementing Solo Calculus expressions and operations
- Simple Read-Eval-Print-Loop interface for inputting expressions and computing reductions
- Datatypes for Solo Diagrams and (some) operations
- Proof-of-Concept for Diagram visualisation
- Demonstration at the end ...

Plans Moving Forward

- Completion of Solo Diagrams operations (full reduction semantics)
- Possible conversion between Diagrams and Calculus expressions
- Realtime visualisation of Diagrams using Javascript's D3 library

Demonstration Time

Any questions . . . ?