Linear-algebraic λ-calculus: higher-order and confluence.

Pablo Arrighi*, Gilles Dowek†

5.12.06, Q-Day III, University of Glasgow

*arrighi@imag.fr, IMAG Laboratories, University of Grenoble
†gilles.dowek@polytechnique.fr, LIX, Ecole polytechnique & INRIA

Roadmap

1. Language & semantics
2. Confluence
3. Future works
4. Extras

Roadmap

1. Language & semantics
2. Confluence
3. Future works
4. Extras

A language with

Computation

$\mathbf{t} ::= \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}]$ (B)

Linear algebra.

$\mathbf{t}+\mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v}$ (E)

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}$ (F)

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0},\quad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0}$ (A)

A minimal language with

Computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u} + \mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u} + \mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u} + \beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

A minimal language with

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\ \mathbf{t}) \qquad \mid$

$$\lambda \mathbf{x}.\mathbf{t}\ \mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$$
$$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$$

$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$$

$$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v}),$$
$$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t}),$$
$$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$$
$$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$$
$$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$$

A minimal language with

Higher-order computation

$$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\ \mathbf{t}) \quad \mid$$

$$\lambda \mathbf{x}.\mathbf{t}\ \mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$$

Linear algebra.

$$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$$

$$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$$
$$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$$

$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$$

$$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\ \mathbf{u}) + (\mathbf{t}\ \mathbf{v}),$$
$$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\ \mathbf{t}) + (\mathbf{v}\ \mathbf{t}),$$
$$\mathbf{t}\ \alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\ \mathbf{u}),$$
$$\alpha.\mathbf{u}\ \mathbf{t} \longrightarrow \alpha.(\mathbf{u}\ \mathbf{t}),$$
$$\mathbf{0}\ \mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\ \mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$$

A minimal language with

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

A minimal language with

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad \mid$

$\lambda\mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$

$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$

$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$

$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$

$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v}),$

$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t}),$

$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$

$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$

$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

A minimal language with

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

. . . and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

. . . and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\ \mathbf{t}) \qquad \mid$

$\lambda \mathbf{x}.\mathbf{t}\ \mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\ (\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\ \mathbf{u}) + (\mathbf{t}\ \mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\ \mathbf{t} \longrightarrow (\mathbf{u}\ \mathbf{t}) + (\mathbf{v}\ \mathbf{t}),$
$\mathbf{t}\ \alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\ \mathbf{u}),$
$\alpha.\mathbf{u}\ \mathbf{t} \longrightarrow \alpha.(\mathbf{u}\ \mathbf{t}),$
$\mathbf{0}\ \mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\ \mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

... and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda\mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v}),$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t}),$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

. . . and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad\qquad \mid$

$$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{u} \longrightarrow \mathbf{t}[\mathbf{u}/\mathbf{x}] \qquad (B)$$

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$$
$$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$$

$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u},$$
$$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$$

$$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v}),$$
$$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t}),$$
$$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u}),$$
$$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t}),$$
$$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$$

Linearity

As such this is not even linear:

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \longrightarrow^* \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Instead of

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Copying vs cloning. . .

We must restrict (B) to **base vectors**.

But who are they?

Linearity

As such this is not even linear:

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \longrightarrow^{*} \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Instead of

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Copying vs cloning...

We must restrict (B) to **base vectors**.

But who are they?

Linearity

As such this is not even linear:

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \longrightarrow^* \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Instead of

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Copying vs cloning...
We must restrict (B) to **base vectors**.
But who are they?

Linearity

As such this is not even linear:

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \longrightarrow^* \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Instead of

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}$$

Copying vs cloning...
We must restrict (B) to **base vectors**.
But who are they?

Higher-order

Fixed points, quantum control, black-box algos...
Higher-order usually means

$$\lambda \mathbf{x}.\mathbf{t}\ \lambda \mathbf{y}.\mathbf{u} \longrightarrow \mathbf{t}[\lambda \mathbf{y}.\mathbf{u}/\mathbf{x}]$$

Hence base vectors are
- abstractions (i.e. terms of the form $\lambda \mathbf{y}.\mathbf{u}$);
- variables (by some kind of recurrence).

Machine description interpretation, LISP quotes...

Higher-order

Fixed points, quantum control, black-box algos...
Higher-order usually means

$$\lambda \mathbf{x}.\mathbf{t}\ \lambda \mathbf{y}.\mathbf{u} \longrightarrow \mathbf{t}[\lambda \mathbf{y}.\mathbf{u}/\mathbf{x}]$$

Hence base vectors are
- abstractions (i.e. terms of the form $\lambda \mathbf{y}.\mathbf{u}$);
- variables (by some kind of recurrence).

Machine description interpretation, LISP quotes...

Higher-order

Fixed points, quantum control, black-box algos...
Higher-order usually means

$$\lambda \mathbf{x}.\mathbf{t}\ \lambda \mathbf{y}.\mathbf{u} \longrightarrow \mathbf{t}[\lambda \mathbf{y}.\mathbf{u}/\mathbf{x}]$$

Hence base vectors are
- abstractions (i.e. terms of the form $\lambda \mathbf{y}.\mathbf{u}$);
- variables (by some kind of recurrence).
Machine description interpretation, LISP quotes...

Higher-order

Fixed points, quantum control, black-box algos...
Higher-order usually means

$$\lambda \mathbf{x}.\mathbf{t}\ \lambda \mathbf{y}.\mathbf{u} \longrightarrow \mathbf{t}[\lambda \mathbf{y}.\mathbf{u}/\mathbf{x}]$$

Hence base vectors are
- abstractions (i.e. terms of the form $\lambda \mathbf{y}.\mathbf{u}$);
- variables (by some kind of recurrence).

Machine description interpretation, LISP quotes...

Higher-order

Fixed points, quantum control, black-box algos...
Higher-order usually means

$$\lambda \mathbf{x}.\mathbf{t}\ \lambda \mathbf{y}.\mathbf{u} \longrightarrow \mathbf{t}[\lambda \mathbf{y}.\mathbf{u}/\mathbf{x}]$$

Hence base vectors are
- abstractions (i.e. terms of the form $\lambda \mathbf{y}.\mathbf{u}$);
- variables (by some kind of recurrence).

Machine description interpretation, LISP quotes...

A minimal language. . . and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda\mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

(*) b an abstraction or a variable.

Linear algebra.

$\mathbf{t}+\mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u} \qquad (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v})$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t})$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Roadmap

1. Language & semantics
2. Confluence
3. Future works
4. Extras

Linearity

. . . is a matter for confluence!

$$\begin{array}{l} \lambda x.(x\,x)(u+v) \not\longrightarrow^* u\,u + u\,v + v\,u + v\,v \\ \quad\downarrow \\ \lambda x.(x\,x)u + \lambda x.(x\,x)v \longrightarrow u\,u + v\,v \end{array}$$

The restriction forbids the above banch.
Is it now confluent?

Linearity

. . . is a matter for confluence!

$$\begin{array}{l}\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \not\longrightarrow^{*} \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}\\ \quad\downarrow\\ \lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v}\end{array}$$

The restriction forbids the above banch.
Is it now confluent?

Linearity

. . . is a matter for confluence!

$$\begin{array}{l} \lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})(\mathbf{u}+\mathbf{v}) \not\longrightarrow^* \mathbf{u}\,\mathbf{u}+\mathbf{u}\,\mathbf{v}+\mathbf{v}\,\mathbf{u}+\mathbf{v}\,\mathbf{v} \\ \quad\downarrow \\ \lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{u}+\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{x})\mathbf{v} \longrightarrow \mathbf{u}\,\mathbf{u}+\mathbf{v}\,\mathbf{v} \end{array}$$

The restriction forbids the above banch.
Is it now confluent?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow \ldots$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$

$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms. These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$

$$\downarrow_*$$

$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**. But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow\ \infty$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$
$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms. These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$
$$\downarrow_*$$
$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**. But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow$ ∞

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$

$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms. These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$

$$\downarrow_*$$

$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**. But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow$ ∞

$$\mathbf{Yb} \equiv \lambda\mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda\mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$
$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms.
These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$
$$\downarrow_*$$
$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**.
But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow \ \infty$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$

$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms.

These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$

$$\downarrow_*$$

$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**.

But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow \ \infty$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$
$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms. These are again a matter for confluence!

$$\begin{array}{l} \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} \\ \ \downarrow_* \\ \mathbf{0} \end{array}$$

High school teacher says we must restrict (F) to **finite vectors**. But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow\ \infty$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\,\lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$
$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms. These are again a matter for confluence!

$$\begin{array}{l} \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} \\ \ \downarrow_* \\ \mathbf{0} \end{array}$$

High school teacher says we must restrict (F) to **finite vectors**.
But who are they?

Infinity

Untyped λ-calculus + linear algebra $\Rightarrow\ \infty$

$$\mathbf{Yb} \equiv \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))\ \lambda \mathbf{x}.(\mathbf{b} + (\mathbf{x}\,\mathbf{x}))$$
$$\mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb}$$

But whoever says infinity says trouble says... indefinite forms.
These are again a matter for confluence!

$$\mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b} + \mathbf{Yb} - \mathbf{Yb} \longrightarrow \mathbf{b}$$
$$\downarrow_*$$
$$\mathbf{0}$$

High school teacher says we must restrict (F) to **finite vectors**.
But who are they?

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors...

...could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Finite vectors. . .

. . . could be:

- finite vector norm? hard to define.
- normalizable terms? undecidable.
- normal terms? OK.

Comp. Sci "normal" = Maths "normal" in this calculus!

Hiding indefinite forms

We restrict (F) to $\mathbf{u}$ normal.
Now

$$\mathbf{Yb} - \mathbf{Yb} \not\longrightarrow \mathbf{0}$$

But

$$\begin{array}{l} \lambda \mathbf{x}.(\mathbf{x}\,\mathbf{y} - \mathbf{x}\,\mathbf{y})\,\lambda \mathbf{y}.\mathbf{Yb} \longrightarrow^{*} \mathbf{0} \\ \quad \downarrow_{*} \\ \mathbf{Yb} - \mathbf{Yb} \end{array}$$

Hiding indefinite forms

We restrict (F) to $\mathbf{u}$ closed normal.
Now

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{y} - \mathbf{x}\,\mathbf{y})\lambda \mathbf{y}.\mathbf{Yb} \not\longrightarrow^* \mathbf{0}$$

Hiding indefinite forms

We restrict (F) to $\mathbf{u}$ closed normal.
Now

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{y} - \mathbf{x}\,\mathbf{y})\lambda \mathbf{y}.\mathbf{Yb} \not\longrightarrow^* \mathbf{0}$$

But

$$\begin{array}{l}(\lambda \mathbf{y}.\mathbf{Yb})(\lambda \mathbf{x}.\mathbf{x} - \lambda \mathbf{x}.\mathbf{x}) \longrightarrow^* \mathbf{Yb} - \mathbf{Yb} \\ \downarrow_* \\ \mathbf{0}\end{array}$$

Hiding indefinite forms

We restrict (F) to $\mathbf{u}$ closed normal.
Now

$$\lambda \mathbf{x}.(\mathbf{x}\,\mathbf{y} - \mathbf{x}\,\mathbf{y})\lambda \mathbf{y}.\mathbf{Y}\mathbf{b} \not\longrightarrow^* \mathbf{0}$$

But

$$\begin{array}{l}(\lambda \mathbf{y}.\mathbf{Y}\mathbf{b})(\lambda \mathbf{x}.\mathbf{x} - \lambda \mathbf{x}.\mathbf{x}) \longrightarrow^* \mathbf{Y}\mathbf{b} - \mathbf{Y}\mathbf{b} \\ \downarrow_* \\ \mathbf{0}\end{array}$$

We restrict (A) to $\mathbf{u}$ closed normal, $\mathbf{u} + \mathbf{v}$ closed normal.

A minimal language... and its semantics.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

(*) $\mathbf{b}$ an abstraction or a variable.
(**) $\mathbf{u}$ closed normal.
(***) $\mathbf{u}+\mathbf{v}$ closed normal.
(****) $\mathbf{u}$ closed normal.

Linear algebra.

$\mathbf{t}+\mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}\,(**)\ (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v})\,(***)$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t})\,(***)$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})\,(****)$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})\,(****)$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Proof architecture.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \quad \mid$

$\lambda\mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

Non terminating. Define $B^{||}$.

Parallel moves lemma.

Terminating. AC extentions.

Critical pairs lemma.

-Automatic for non cond. N.

-By hands for F, A.

Parameterized on scalars.

Now the two commute.

The union is confluent.

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}\,(**)\ (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v})\,(***)$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t})\,(***)$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})\,(****)$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})\,(****)$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Proof architecture.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda\mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad \mid$

$\lambda\mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

Non terminating. Define $B^{||}$.
Parallel moves lemma.
Terminating. AC extentions.
Critical pairs lemma.
-Automatic for non cond. N.
-By hands for F, A.
Parameterized on scalars.
Now the two commute.
The union is confluent.

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha\times\beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u}+\alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}\,(**)\ (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u})+(\mathbf{t}\,\mathbf{v})\,(***)$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t})+(\mathbf{v}\,\mathbf{t})\,(***)$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})\,(****)$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})\,(****)$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Proof architecture.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \,|\, \lambda \mathbf{x}.\mathbf{t} \,|\, (\mathbf{t}\,\mathbf{t}) \qquad |$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

Non terminating. Define $B^{||}$.
Parallel moves lemma.
Terminating. AC extentions.
Critical pairs lemma.
-Automatic for non cond. N.
-By hands for F, A.
Parameterized on scalars.
Now the two commute.
The union is confluent.

Linear algebra.

$\mathbf{t} + \mathbf{t} \,|\, \alpha.\mathbf{t} \,|\, \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}\,(**)\ (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v})\,(***)$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t})\,(***)$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})\,(****)$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})\,(****)$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Proof architecture.

Higher-order computation

$\mathbf{t} ::= \quad \mathbf{x} \mid \lambda \mathbf{x}.\mathbf{t} \mid (\mathbf{t}\,\mathbf{t}) \qquad \mid$

$\lambda \mathbf{x}.\mathbf{t}\,\mathbf{b} \longrightarrow \mathbf{t}[\mathbf{b}/\mathbf{x}]\,(*) \qquad (B)$

Non terminating. Define $B^{||}$.
Parallel moves lemma.
Terminating. AC extentions.
Critical pairs lemma.
-Automatic for non cond. N.
-By hands for F, A.
Parameterized on scalars.
Now the two commute.
The union is confluent.

Linear algebra.

$\mathbf{t} + \mathbf{t} \mid \alpha.\mathbf{t} \mid \mathbf{0}$

$1.\mathbf{u} \longrightarrow \mathbf{u},\ 0.\mathbf{u} \longrightarrow \mathbf{0},\ \alpha.\mathbf{0} \longrightarrow \mathbf{0},$
$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u},\ \alpha.(\beta.\mathbf{u}) \longrightarrow \alpha \times \beta.\mathbf{u},$
$\alpha.(\mathbf{u}+\mathbf{v}) \longrightarrow \alpha.\mathbf{u} + \alpha.\mathbf{v} \qquad (E)$

$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\mathbf{u} \longrightarrow (\alpha+1).\mathbf{u}\,(**)$
$\alpha.\mathbf{u}+\beta.\mathbf{u} \longrightarrow (\alpha+\beta).\mathbf{u}\,(**)\ (F)$

$\mathbf{t}\,(\mathbf{u}+\mathbf{v}) \longrightarrow (\mathbf{t}\,\mathbf{u}) + (\mathbf{t}\,\mathbf{v})\,(***)$
$(\mathbf{u}+\mathbf{v})\,\mathbf{t} \longrightarrow (\mathbf{u}\,\mathbf{t}) + (\mathbf{v}\,\mathbf{t})\,(***)$
$\mathbf{t}\,\alpha.\mathbf{u} \longrightarrow \alpha.(\mathbf{t}\,\mathbf{u})\,(****)$
$\alpha.\mathbf{u}\,\mathbf{t} \longrightarrow \alpha.(\mathbf{u}\,\mathbf{t})\,(****)$
$\mathbf{0}\,\mathbf{u} \longrightarrow \mathbf{0}, \qquad \mathbf{u}\,\mathbf{0} \longrightarrow \mathbf{0} \qquad (A)$

Roadmap

1. Language & semantics
2. Confluence
3. Future works
4. Extras

Motivations/Future works/Open problem

Quantum computation, computability in linear algebraic structure.

Typing: for unitarity? for developping novel logics.

Investigate the connection with linear λ-calculus, linear logic.

Open problem: find a model.

Motivations/Future works/Open problem

Quantum computation, computability in linear algebraic structure.

Typing: for unitarity? for developping novel logics.

Investigate the connection with linear λ-calculus, linear logic.

Open problem: find a model.

Motivations/Future works/Open problem

Quantum computation, computability in linear algebraic structure.

Typing: for unitarity? for developping novel logics.

Investigate the connection with linear λ-calculus, linear logic.

Open problem: find a model.

Roadmap

1. Language & semantics
2. Confluence
3. Future works
4. Extras

Encoding booleans

$$\mathbf{true} \equiv \lambda x.\lambda y.x$$
$$\mathbf{false} \equiv \lambda x.\lambda y.y$$

Encoding the Not gate

$$\mathbf{Not} \equiv \lambda y.\Big((y * \mathbf{false}) * \mathbf{true}\Big)$$

Encoding the Phase gate

$$\mathbf{Phase} \equiv \lambda y.\left(\left(\left(y * \lambda x.(e^{i\frac{\pi}{4}}.\mathsf{true})\right) * \lambda x.\mathsf{false}\right) * \_\right)$$

where _ stands for dead code.

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\Big(\big((y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_\Big) * \mathbf{true}$$

$$\big((\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\big) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\Big(\big((y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_\Big) * \mathbf{true}$$

$$\big((\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\big) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\Big(\big((y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_\Big) * \mathbf{true}$$

$$\big((\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_$$

$$\big(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\big) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\left(\left(\left(y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_\right) * \mathbf{true}$$

$$\left(\left(\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_$$

$$\left(\left((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_$$

$$\left(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\right) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\left(\left(\left(y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_\right) * \mathbf{true}$$

$$\left(\left(\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_$$

$$\left(\left((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})\right) * \lambda x.\mathbf{false}\right) * \_$$

$$\left(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\right) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Running the Phase gate

Phase $*$ **true** yields

$$\lambda y.\Big(\big((y * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\big) * \_\Big) * \mathbf{true}$$

$$\Big((\mathbf{true} * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\Big) * \_$$

$$\Big(((\lambda x.\lambda y.x) * \lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true})) * \lambda x.\mathbf{false}\Big) * \_$$

$$\Big(\lambda x.\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \lambda x.\mathbf{false}\Big) * \_$$

$$\lambda x.(e^{i\frac{\pi}{4}}.\mathbf{true}) * \_$$

$$e^{i\frac{\pi}{4}}.\mathbf{true}$$

Encoding the Hadamard gate

$$\textbf{Hadamard} \equiv \lambda y.\Big(\Big(\big(y * \lambda x.(\textbf{false} - \textbf{true})\big) * \lambda x.(\textbf{false} + \textbf{true})\Big) * \_\Big)$$

where _ stands for dead code.

Encoding tensors

$$\otimes \equiv \lambda x.\lambda y.\lambda f.((f * x) * y)$$

$$\pi_{\mathbf{1}} \equiv \lambda x.\lambda y.x$$

$$\pi_{\mathbf{2}} \equiv \lambda x.\lambda y.y$$

$$\bigotimes \equiv \lambda f.\lambda g.\lambda x.\Big(\Big(\otimes * (f * (\pi_1 * x))\Big) * (g * (\pi_2 * x))\Big)$$

E.g. $\mathbf{H}^{\otimes \mathbf{2}} \equiv ((\bigotimes \textbf{Hadamard}) * \textbf{Hadamard})$

Encoding the CNOT gate

$$\mathbf{Cnot} \equiv$$

$$\lambda x. \Bigg(\bigg(\otimes * (\pi_1 * x) \bigg) * \bigg(\Big((\pi_1 * x) * \big(\mathbf{Not} * (\pi_2 * x) \big) \Big) * (\pi_2 * x) \bigg) \Bigg)$$

Expressing Deutsch's algorithm parametrically

$$\mathbf{Deutsch} \equiv$$

$$\lambda f.\left(\mathbf{H}^{\otimes 2} * \left(f * \left(\mathbf{H}^{\otimes 2} * ((\otimes \mathbf{false}) * \mathbf{true})\right)\right)\right)$$

Trends

Van Tonder/ Valiron & Selinger's quantum λ-calculus: Uses the linear λ-calculus framework. Heterogeneous. Quantum resources (treated as linear) and classical resources (treated as nonlinear):

$$(\lambda_q) \qquad t ::= x \,|\, \lambda x.t \,|\, (t \;\; t) \,|\, c \,|\, !t \,|\, \lambda !x.t$$

$$c ::= 0 \,|\, 1 \,|\, H \,|\, CNOT \,|\, P$$

(together with the well-formedness rules of the classical linear λ calculus)

$$(\mathcal{R}_q) \qquad (\lambda x.t \;\; s) \xrightarrow{\beta} t[s/x]$$

$$H \;\; 0 \longrightarrow 0 + 1$$

$$H \;\; 1 \longrightarrow 0 - 1$$

$$\vdots$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 = p \qquad p \text{ plus } S(q) = S(p) \text{ plus } q$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 \longrightarrow p \qquad p \text{ plus } S(q) \longrightarrow S(p) \text{ plus } q$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 \longrightarrow p \qquad p \text{ plus } S(q) \longrightarrow S(p) \text{ plus } q$$

$$2 + 2 \equiv S(S(0)) \text{ plus } S(S(0))$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 \longrightarrow p \qquad p \text{ plus } S(q) \longrightarrow S(p) \text{ plus } q$$

$$2 + 2 \equiv S(S(S(0))) \text{ plus } S(0)$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 \longrightarrow p \qquad p \text{ plus } S(q) \longrightarrow S(p) \text{ plus } q$$

$$2 \text{ plus } 2 \equiv S(S(S(S(0)))) \text{ plus } 0$$

Term Rewrite Systems

These consist in

- a finite set of rules $l \longrightarrow r$, each interpreted as follows
- $t = t[\sigma l]_p$ should be rewritten into a term $t' = t[\sigma r]_p$.

Remarks:

- unambiguous so long as it is confluent. Preferably terminating.
- $l \longrightarrow r$ sometimes viewed as an oriented form of $l = r$.

E.g.

$$p \text{ plus } 0 \longrightarrow p \qquad p \text{ plus } S(q) \longrightarrow S(p) \text{ plus } q$$

$$2 + 2 \equiv S(S(S(S(0)))) \equiv 4$$

An algorithm

The algorithm for expressing vectors as linear combinations of the unknown:

- implemented elegantly as a TRS;
- shown terminating and confluent;
- provided a field, defines the notion of vectorial space.

An algorithm

The algorithm for expressing vectors as linear combinations of the unknown:

- implemented elegantly as a TRS;
- shown terminating and confluent;
- provided a field, defines the notion of vectorial space.

An algorithm

The algorithm for expressing vectors as linear combinations of the unknown:

- implemented elegantly as a TRS;
- shown terminating and confluent;
- provided a field, defines the notion of vectorial space.

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

We want to develop:

$$3.(\mathbf{x} + \mathbf{y}) \longrightarrow 3.\mathbf{x} + 3.\mathbf{y}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$

We want to develop:

$$3.(\mathbf{x}+\mathbf{y}) \longrightarrow 3.\mathbf{x}+3.\mathbf{y}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$

But factor:

$$3.\mathbf{x}+1.\mathbf{x} \longrightarrow (3+1).\mathbf{x}$$

$$3.(2.\mathbf{x}) \longrightarrow (3\times 2).\mathbf{x}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\mu).\mathbf{u}$$

But factor:

$$3.\mathbf{x}+1.\mathbf{x} \longrightarrow (3+1).\mathbf{x}$$
$$3.(2.\mathbf{x}) \longrightarrow (3\times 2).\mathbf{x}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\mu).\mathbf{u}$$

We want to get rid of neutral elements:

$$\mathbf{x}+\mathbf{0} \longrightarrow \mathbf{x}$$

$$0.\mathbf{x} \longrightarrow \mathbf{0}$$

$$1.\mathbf{x} \longrightarrow \mathbf{x}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\mu).\mathbf{u}$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u}$$
$$1.\mathbf{u} \longrightarrow \mathbf{u}$$
$$0.\mathbf{u} \longrightarrow \mathbf{0}$$

We want to get rid of neutral elements:

$$\mathbf{x}+\mathbf{0} \longrightarrow \mathbf{x}$$
$$0.\mathbf{x} \longrightarrow \mathbf{0}$$
$$1.\mathbf{x} \longrightarrow \mathbf{x}$$

Let us seek a TRS for presenting numbers as linear combinations of unknowns:

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\mu).\mathbf{u}$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u}$$
$$1.\mathbf{u} \longrightarrow \mathbf{u}$$
$$0.\mathbf{u} \longrightarrow \mathbf{0}$$

Modulo AC(+)

Consider:

$3.\mathbf{x} + 1.\mathbf{x} \longrightarrow (3+1).\mathbf{x}$

But: $3.\mathbf{x} + 1.\mathbf{x} \longrightarrow 3.\mathbf{x} + \mathbf{x}$

We need:

$$\lambda.\mathbf{u} + \mathbf{u} \longrightarrow (\lambda + 1).\mathbf{u}$$

An algorithm

$$\lambda.(\mathbf{u}+\mathbf{v}) \longrightarrow \lambda.\mathbf{u}+\lambda.\mathbf{v}$$
$$\lambda.\mathbf{u}+\mu.\mathbf{u} \longrightarrow (\lambda+\mu).\mathbf{u}$$
$$\lambda.(\mu.\mathbf{u}) \longrightarrow (\lambda\mu).\mathbf{u}$$
$$\mathbf{u}+\mathbf{0} \longrightarrow \mathbf{u}$$
$$1.\mathbf{u} \longrightarrow \mathbf{u}$$
$$0.\mathbf{u} \longrightarrow \mathbf{0}$$
$$\lambda.\mathbf{u}+\mathbf{u} \longrightarrow (\lambda+1).\mathbf{u}$$
$$\mathbf{u}+\mathbf{u} \longrightarrow (1+1).\mathbf{u}$$
$$\lambda.\mathbf{0} \longrightarrow \mathbf{0}.$$

Call it $\mathcal{R}$.

Definition: scalar rewrite system

These consist in terminating and confluent TRS on a language containing $+$, $\times$, 0, 1 and such that for all closed terms λ, μ, ν:

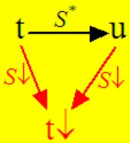
- $0 + \lambda$ and λ,
- $0 \times \lambda$ and 0,
- $1 \times \lambda$ and λ,
- $\lambda \times (\mu + \nu)$ and $(\lambda \times \mu) + (\lambda \times \nu)$,
- $(\lambda + \mu) + \nu$ and $\lambda + (\mu + \nu)$,
- $\lambda + \mu$ and $\mu + \lambda$,
- $(\lambda \times \mu) \times \nu$ and $\lambda \times (\mu \times \nu)$,
- $\lambda \times \mu$ and $\mu \times \lambda$

have the same normal forms and 0 and 1 are normal terms.

Termination and confluence properties

Proposition 1: Assuming $\mathcal{S}$ a confluent and terminating TRS for $\mathbb{K}$, then $\mathcal{R} \cup \mathcal{S}$ is a confluent and terminating TRS for $\mathbb{K}$-vectorial spaces.

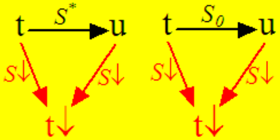
Proof...


t
S*
u
S↓
S↓
t↓

Def. *Subsumption*
A terminating and confluent TRS $\mathcal{S}$ subsumes a TRS $\mathcal{S}_0$ if whenever $t \xrightarrow{\mathcal{S}_0} u$, t and u have the same $\mathcal{S}$-normal form.
Consider $\mathcal{S}_0$:

$$0 + \lambda \longrightarrow \lambda$$
$$0 \times \lambda \longrightarrow 0$$
$$1 \times \lambda \longrightarrow \lambda$$
$$\lambda \times (\mu + \nu) \longrightarrow (\lambda \times \mu) + (\lambda \times \nu)$$

$\mathcal{S}$ subsumes $\mathcal{S}_0$.


t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓

$\mathcal{R}$ terminates.

Proof technique:

Find $|.| :\longrightarrow \mathbb{N}$ such that

$t \longrightarrow r \quad \Rightarrow \quad |t| > |r|$. CiME does it for you.

$\mathcal{S}_0$ terminates.

Same method.

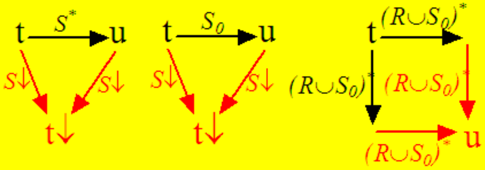
$\mathcal{R} \cup \mathcal{S}$ and $\mathcal{R} \cup \mathcal{S}_0$ terminate.

Proof: $\mathcal{S}$ and $\mathcal{S}_0$ leave $\mathcal{R}$'s polynomial interpretation unchanged.

$\mathcal{R} \cup \mathcal{S}_0$ is confluent.

Proof technique:

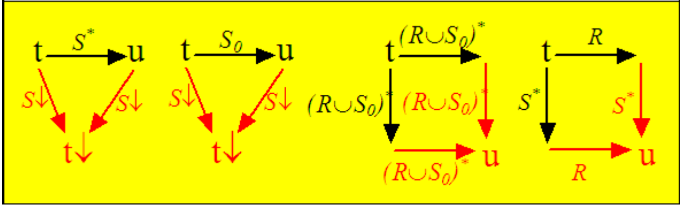
Since it is terminating it is enough to check that it is locally confluent. CiME does it for you.


t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓
t
(R∪S0)*
(R∪S0)*
(R∪S0)*
u
(R∪S0)*

Commutation

A TRS $\mathcal{R}$ commutes with a TRS $\mathcal{R}'$ if whenever $t \xrightarrow{\mathcal{R}} u_1$ and $t \xrightarrow{\mathcal{R}'} u_2$, then there exists w such that $u_1 \xrightarrow{\mathcal{R}'} w$ and $u_2 \xrightarrow{\mathcal{R}} w$.

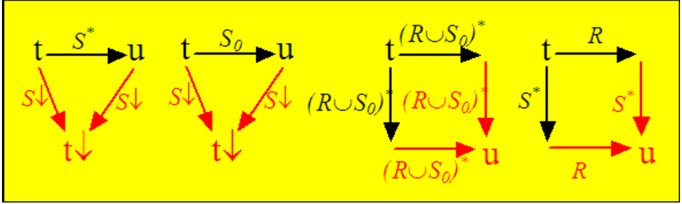
$\mathcal{R}$ commutes with $\mathcal{S}^*$.

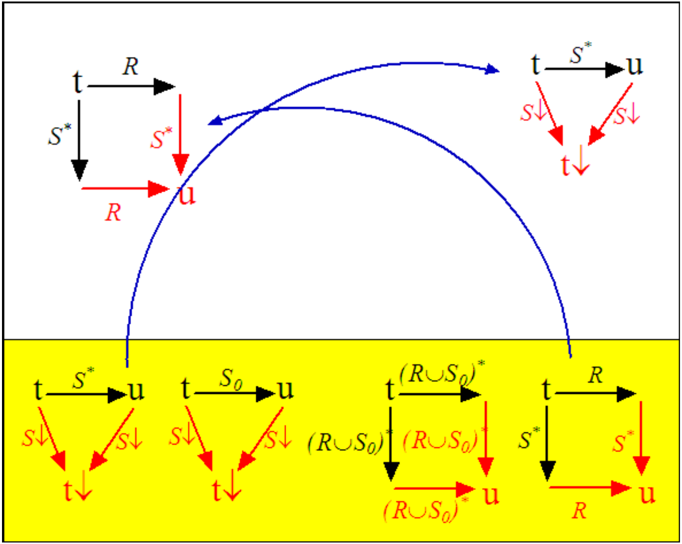

t
S^*
u
$S\downarrow$
$S\downarrow$
t↓
t
S_0
u
$S\downarrow$
$S\downarrow$
t↓
t
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
u
t
R
S^*
S^*
R
u

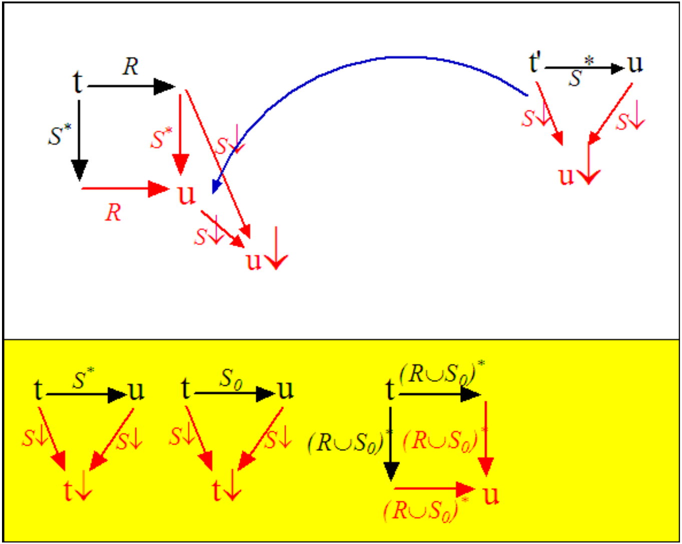
Key Lemma: Let R, S and S_0 be three TRS such that:

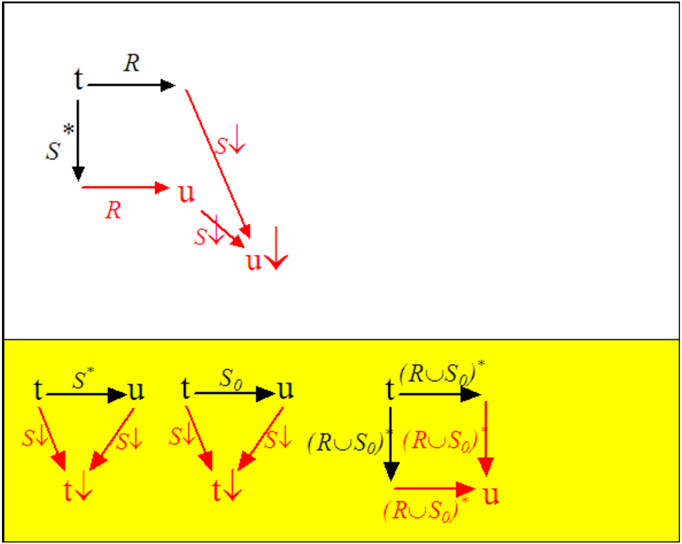
- S is terminating and confluent;
- S subsumes S_0;
- $R \cup S_0$ is confluent;
- R commutes with S^*;
- R terminates.

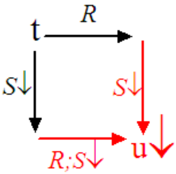
Then $R \cup S$ is confluent.

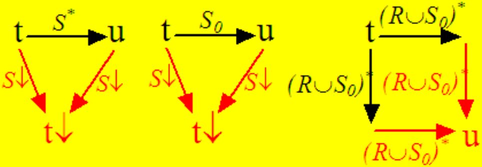

t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓
t
(R∪S0)*
(R∪S0)*
(R∪S0)*
(R∪S0)*
u
t
R
S*
S*
R
u

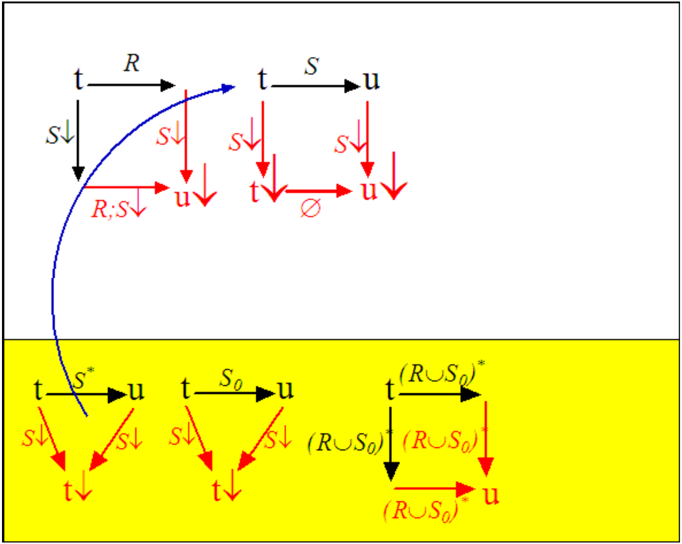

t
R
S*
S*
u
R
t
S*
u
S↓
S↓
t↓
t
S*
u
S↓
S↓
t↓
t
S₀
u
S↓
S↓
t↓
t
(R∪S₀)*
(R∪S₀)*
(R∪S₀)*
u
(R∪S₀)*
t
R
S*
S*
u
R

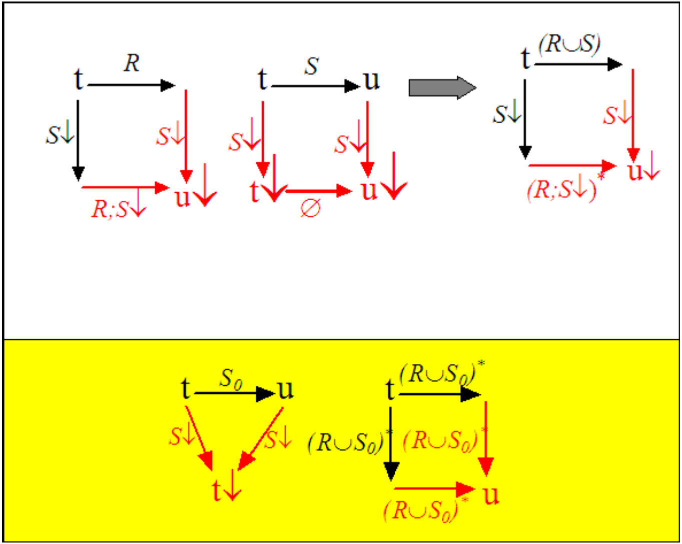

t
R
S*
S*
S↓
u
R
S↓
u↓
t'
*
S
u
S↓
S↓
u↓
t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓
t
(R∪S0)*
(R∪S0)*
(R∪S0)*
u
(R∪S0)*

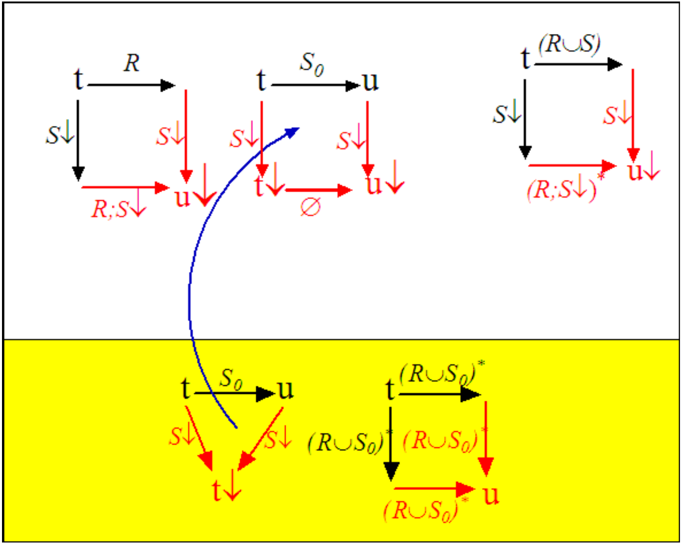

t
R
S
*
u
R
S↓
S↓
u↓
t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓
t
(R∪S0)*
(R∪S0)*
(R∪S0)*
u
(R∪S0)*

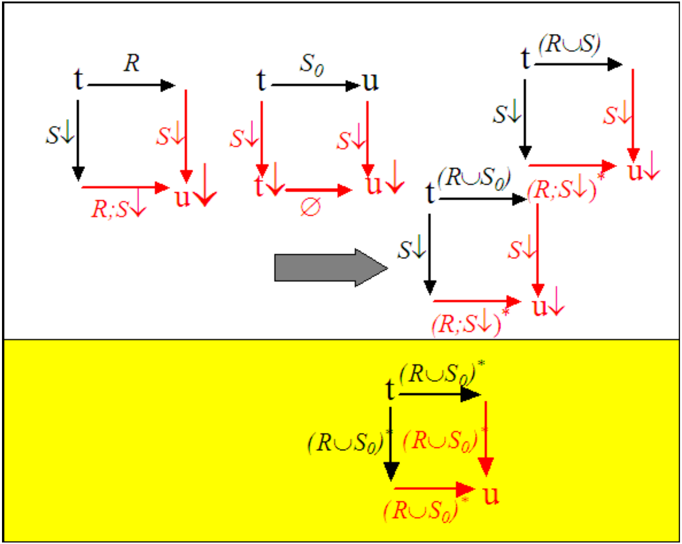

R
t
S↓
S↓
u↓
R;S↓

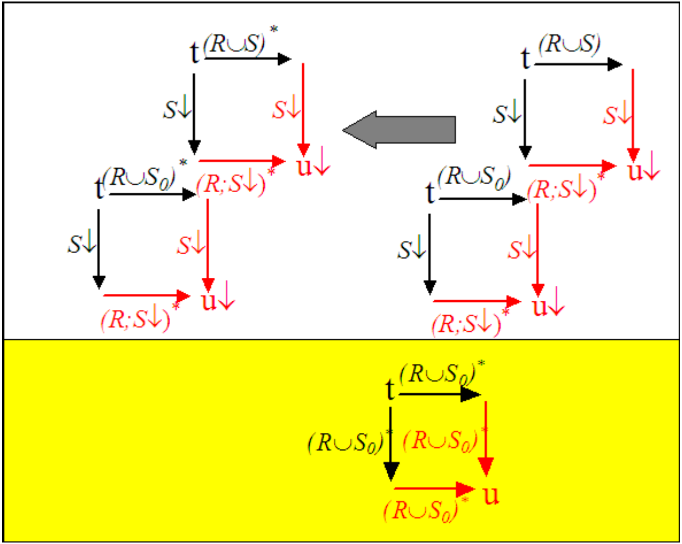

t
S*
u
S↓
S↓
t↓
t
S0
u
S↓
S↓
t↓
t
(R∪S0)*
(R∪S0)*
(R∪S0)*
(R∪S0)*
u

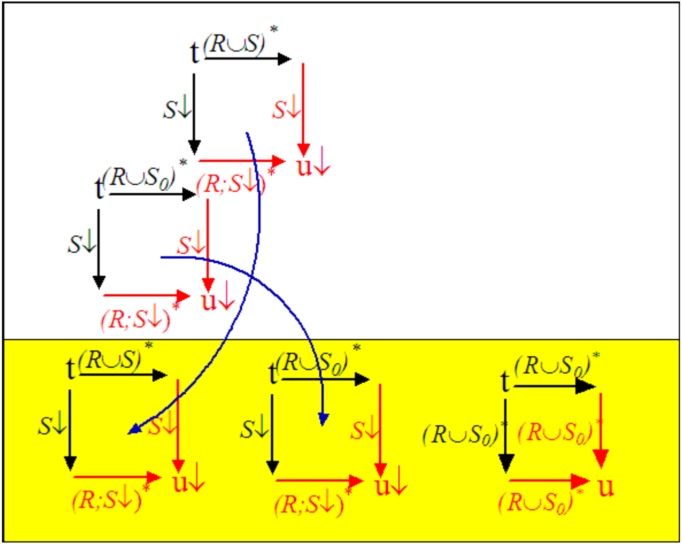

t
R
S↓
S↓
R;S↓
u↓
t
S
u
S↓
S↓
t↓
∅
u↓
t
S*
u
S↓
S↓
t↓
t
S_0
u
S↓
S↓
t↓
t
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
u

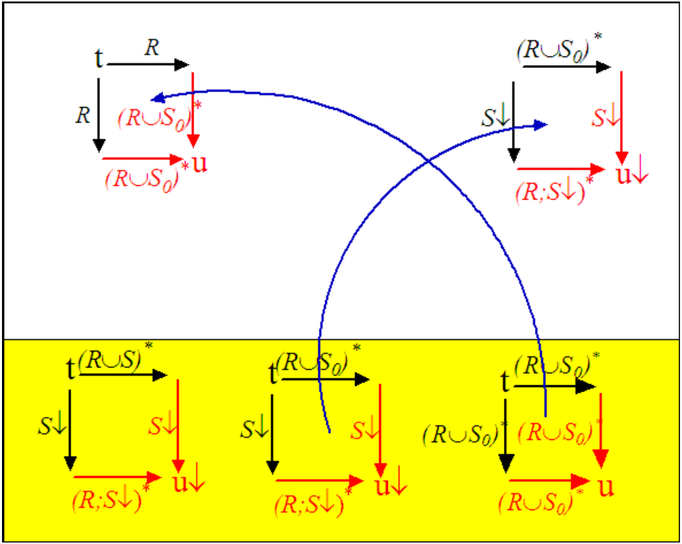

t
R
S↓
S↓
R;S↓
u↓
t
S
u
S↓
S↓
t↓
∅
u↓
t
(R∪S)
S↓
S↓
(R;S↓)*
u↓
t
S_0
u
S↓
S↓
t↓
t
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$
u

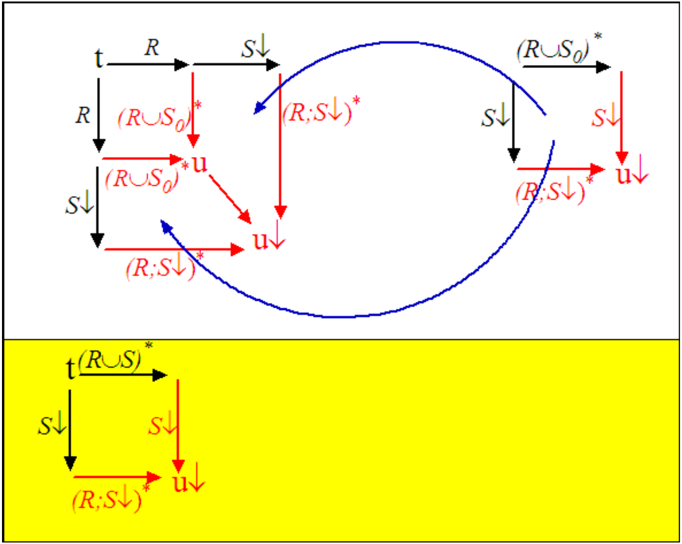


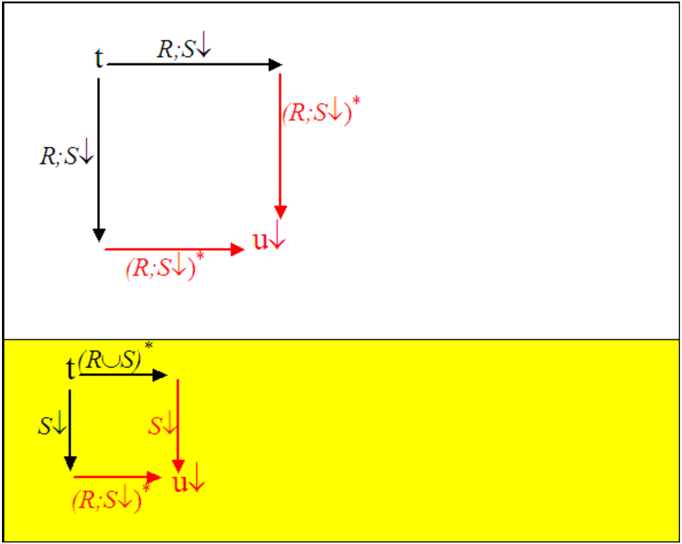

t
R
S↓
S↓
R;S↓
u↓
t
S_0
u
S↓
S↓
t↓
∅
u↓
t
(R∪S)
S↓
S↓
u↓
t
$(R∪S_0)$
$(R;S↓)^*$
S↓
S↓
$(R;S↓)^*$
u↓
t
$(R∪S_0)^*$
$(R∪S_0)^*$
$(R∪S_0)^*$
$(R∪S_0)^*$
u

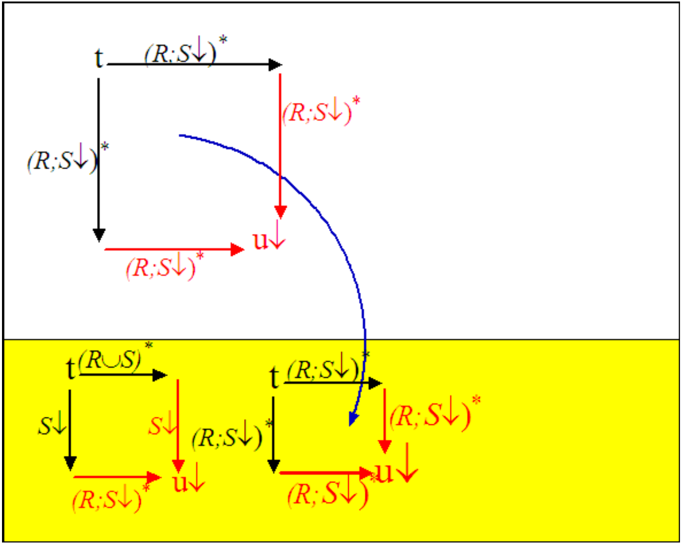


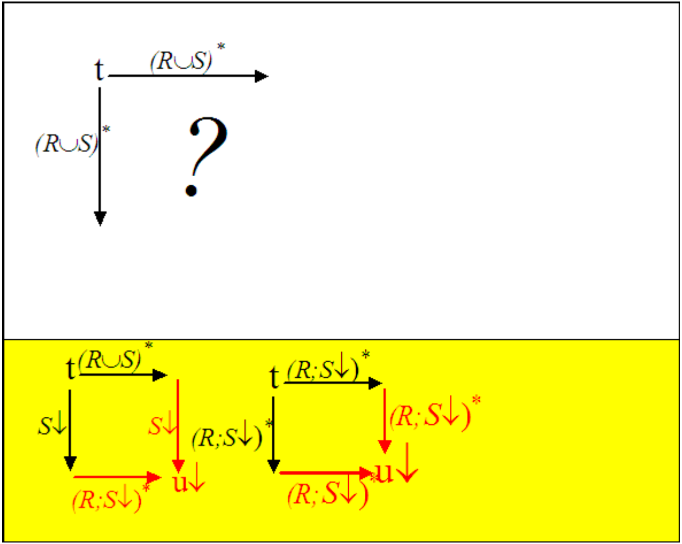

t
(R∪S)*
S↓
S↓
u↓
t
(R∪S₀)*
(R;S↓)*
S↓
S↓
u↓
(R;S↓)*
t
(R∪S)*
S↓
S↓
u↓
(R;S↓)*
t
(R∪S₀)*
S↓
S↓
u↓
(R;S↓)*
t
(R∪S₀)*
(R∪S₀)*
(R∪S₀)*
u
(R∪S₀)*

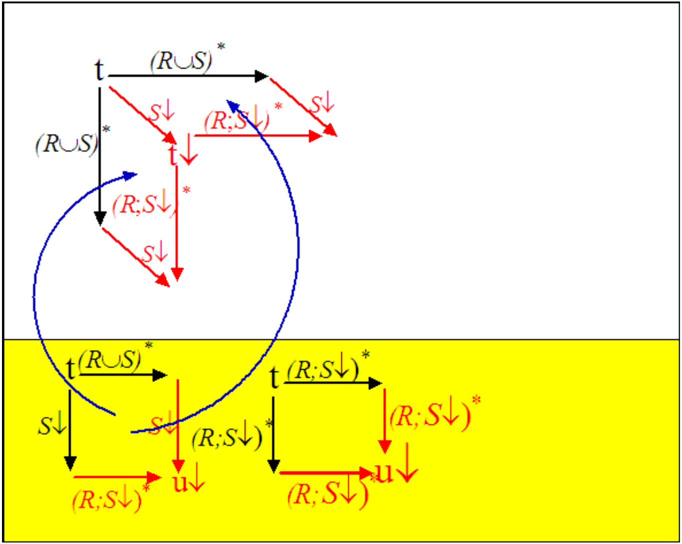

t
R
R
$(R\cup S_0)^*$
u
$(R\cup S_0)^*$
$S\downarrow$
$S\downarrow$
$u\downarrow$
$(R;S\downarrow)^*$
$(R\cup S)^*$
$(R\cup S_0)^*$
$(R\cup S_0)^*$

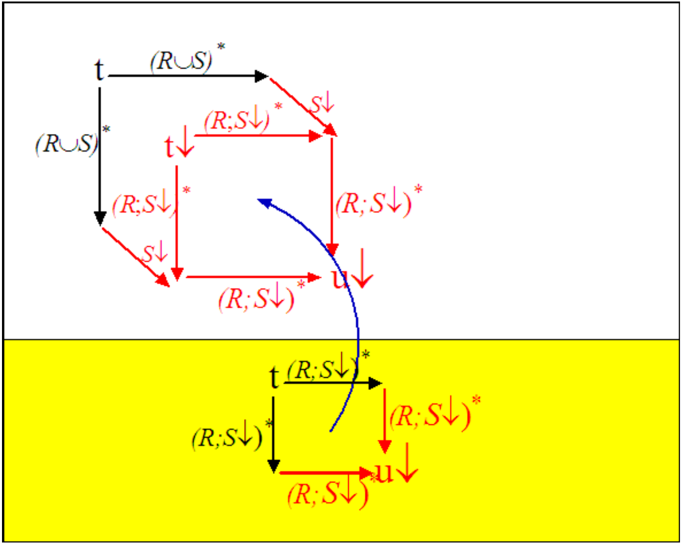

t
R
S↓
R
(R∪S0)*
(R;S↓)*
(R∪S0)*
u
S↓
u↓
(R;S↓)*
(R∪S0)*
S↓
S↓
(R;S↓)*
u↓
t (R∪S)*
S↓
S↓
u↓
(R;S↓)*

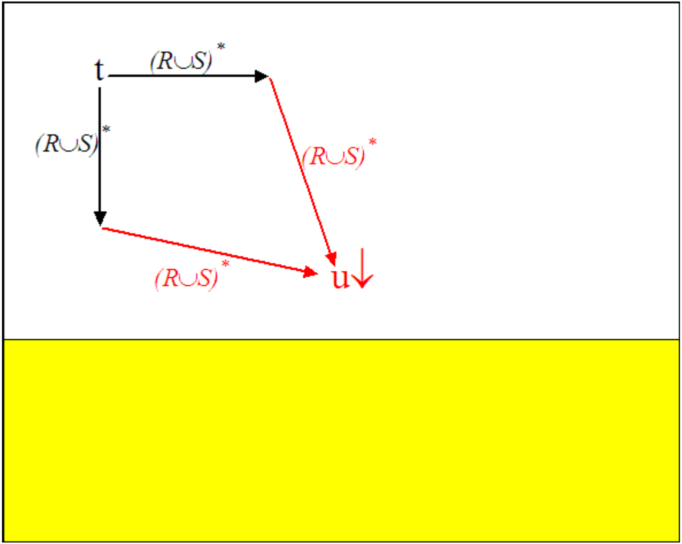

t
R;S↓
(R;S↓)*
R;S↓
u↓
(R;S↓)*
t
(R∪S)*
S↓
S↓
u↓
(R;S↓)*


t
(R;S↓)*
(R;S↓)*
(R;S↓)*
u↓
(R;S↓)*
t
(R∪S)*
S↓
S↓
u↓
(R;S↓)*
t
(R;S↓)*
(R;S↓)*
(R;S↓)*
u↓
(R;S↓)*


t
(R∪S)*
(R∪S)*
?
t
(R∪S)*
S↓
S↓
(R;S↓)*
u↓
(R;S↓)*
t
(R;S↓)*
(R;S↓)*
(R;S↓)*
u↓
(R;S↓)*


t
(R∪S)*
S↓
(R;S↓)*
S↓
(R∪S)*
t↓
(R;S↓)*
S↓
t
(R∪S)*
S↓
S↓
(R;S↓)*
u↓
(R;S↓)*
t
(R;S↓)*
(R;S↓)*
(R;S↓)*
u↓
(R;S↓)*


t
$(R\cup S)^*$
$S\downarrow$
$(R;S\downarrow)^*$
$t\downarrow$
$(R\cup S)^*$
$(R;S\downarrow)^*$
$(R;S\downarrow)^*$
$S\downarrow$
$u\downarrow$
$(R;S\downarrow)^*$
t
$(R;S\downarrow)^*$
$(R;S\downarrow)^*$
$(R;S\downarrow)^*$
$u\downarrow$
$(R;S\downarrow)^*$


t
$(R \cup S)^*$
$(R \cup S)^*$
$(R \cup S)^*$
$(R \cup S)^*$
u↓

The field of quantum computing is a. . .

[Solovay 95, Adleman et al. 97, Kitaev 97, Boykin et al.. . .]
A ring: the additive and multiplicative closure of diadic numbers (binary floats) together with $i^2 = -1$, $\frac{1}{\sqrt{2}} * \frac{1}{\sqrt{2}} = 1/2$.
I.e. a 4-dimensional module.

$$\mathbf{1} * \mathbf{v} \longrightarrow \mathbf{v}$$
$$\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} * \frac{\mathbf{1}}{\sqrt{\mathbf{2}}} \longrightarrow (1/2).\mathbf{1}$$
$$\frac{\mathbf{1}}{\sqrt{\mathbf{2}}} * \mathbf{i} \longrightarrow \frac{\mathbf{i}}{\sqrt{\mathbf{2}}}$$
$$\mathbf{i} * \mathbf{i} \longrightarrow -1.\mathbf{1}$$
$$\vdots$$