The dynamic Geometry of Interaction machine

a call-by-need graph rewriter

Koko Muroya & Dan R. Ghica
(University of Birmingham)

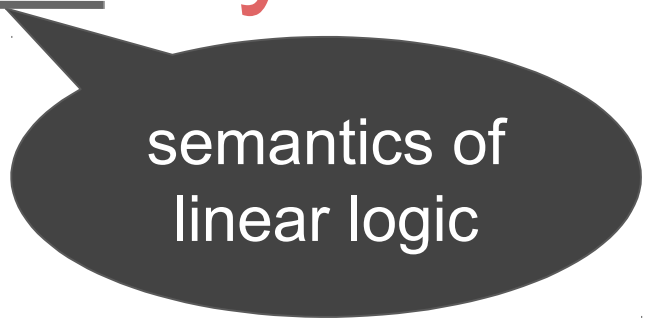
CSL (Stockholm), 21 Aug. 2017

Quantitative analysis by GoI-style token passing

of

space-time trade-off of program execution cost

Quantitative analysis by GoI-style token passing



of

space-time trade-off of program execution cost

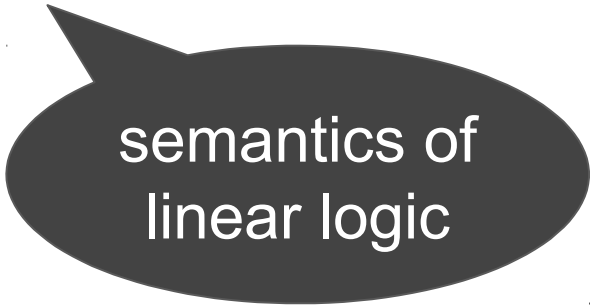
Quantitative analysis by GoI-style token passing of space-time trade-off of program execution cost

semantics of linear logic

abstract machines for lambda-calculus

Interaction abstract machine (IAM)

- [Danos & Regnier ’99]
- call-by-name evaluation
- designed after Geometry of Interaction [Girard ’89]

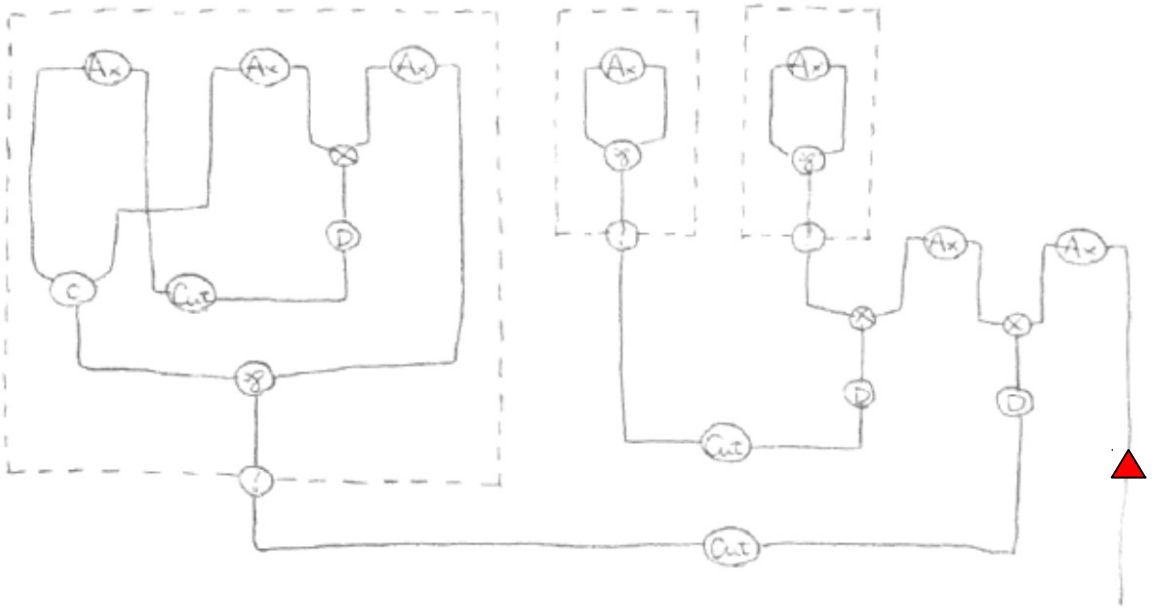


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

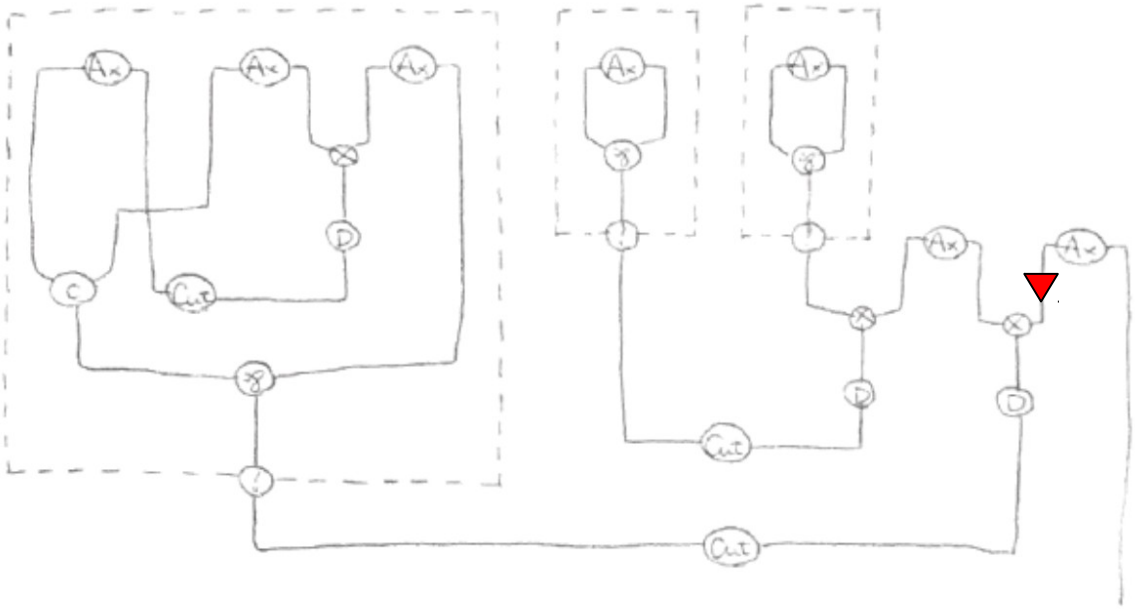


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

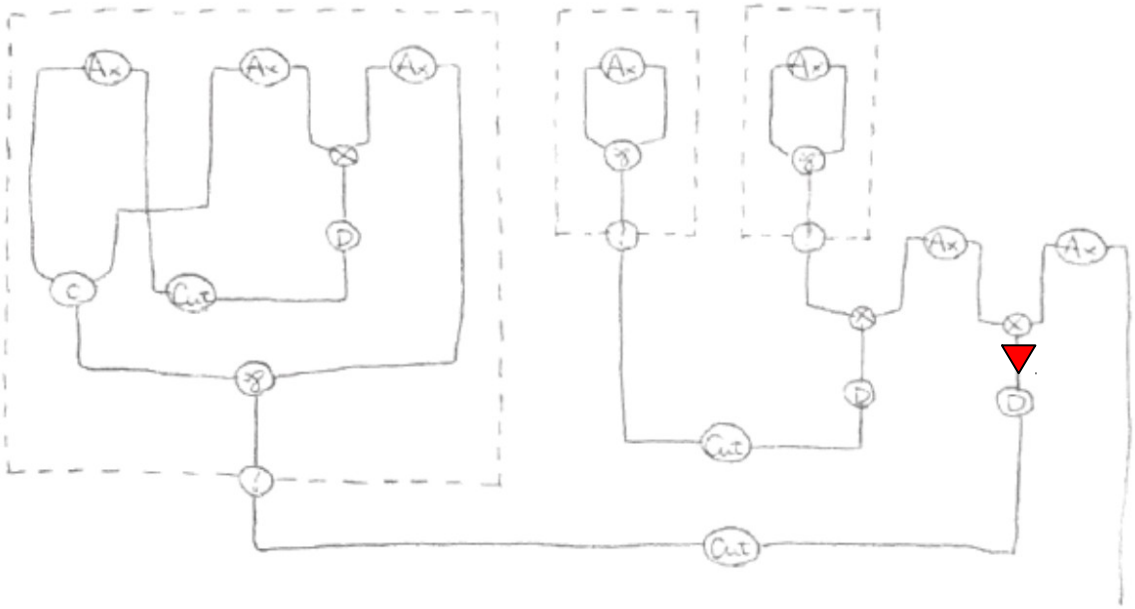


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

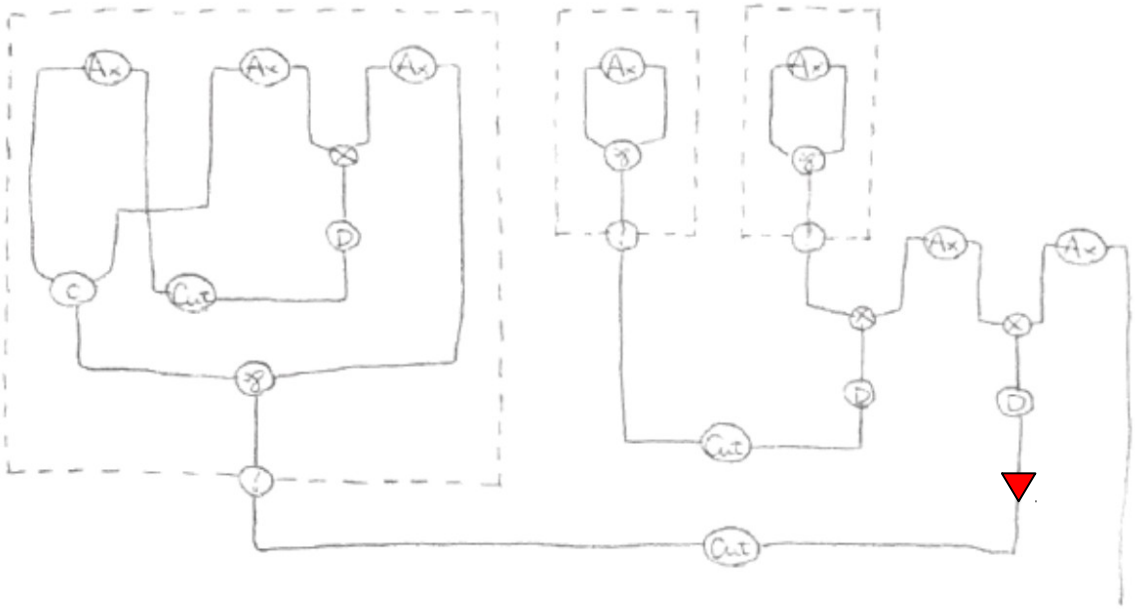


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



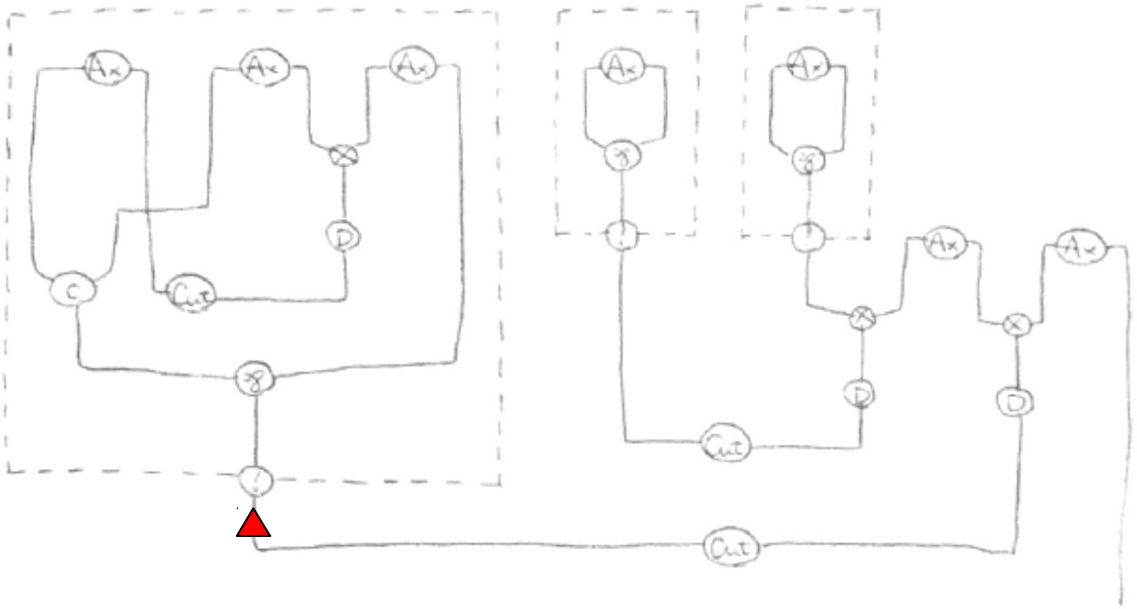
Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



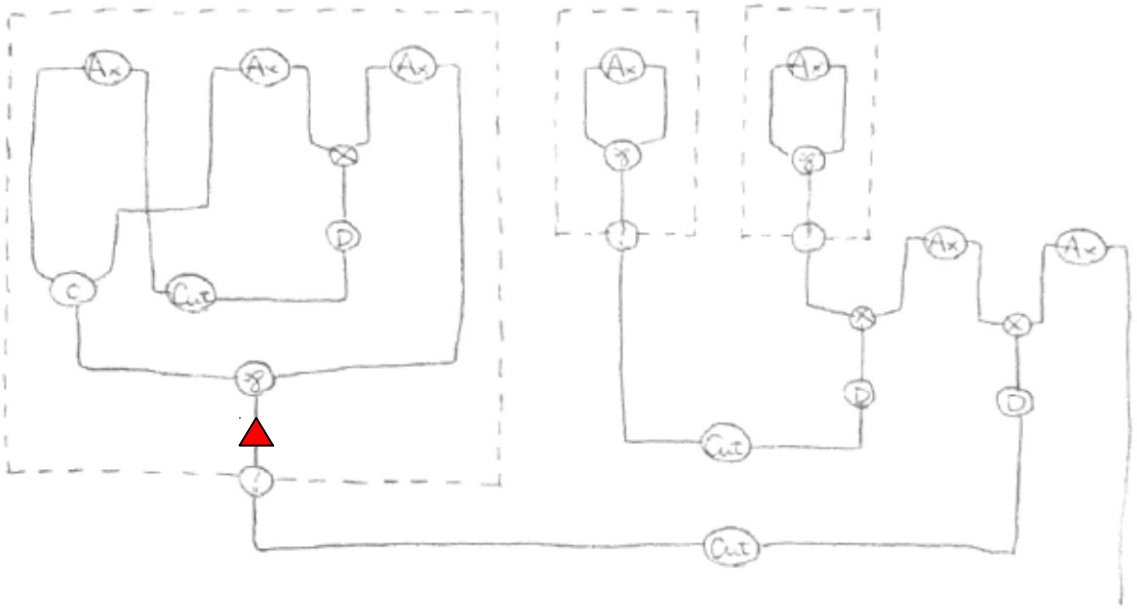
Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

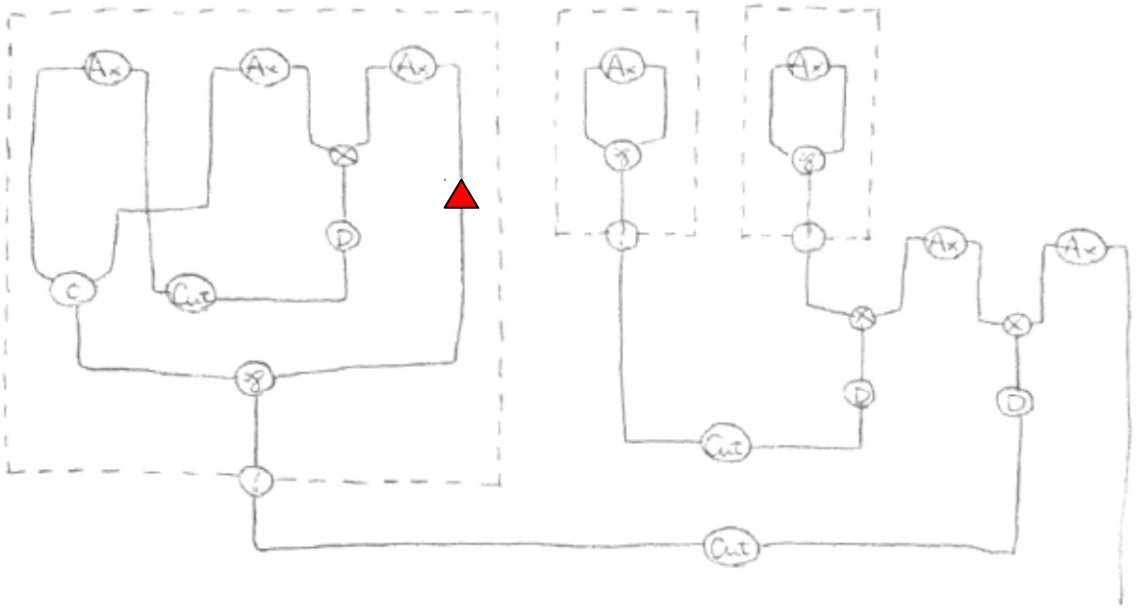
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

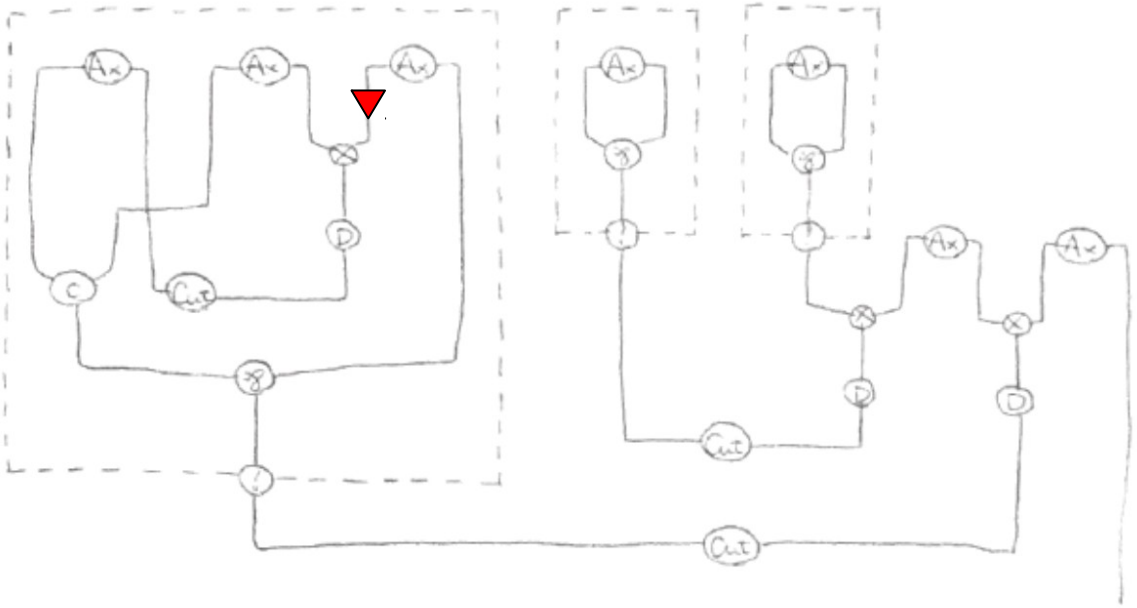
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

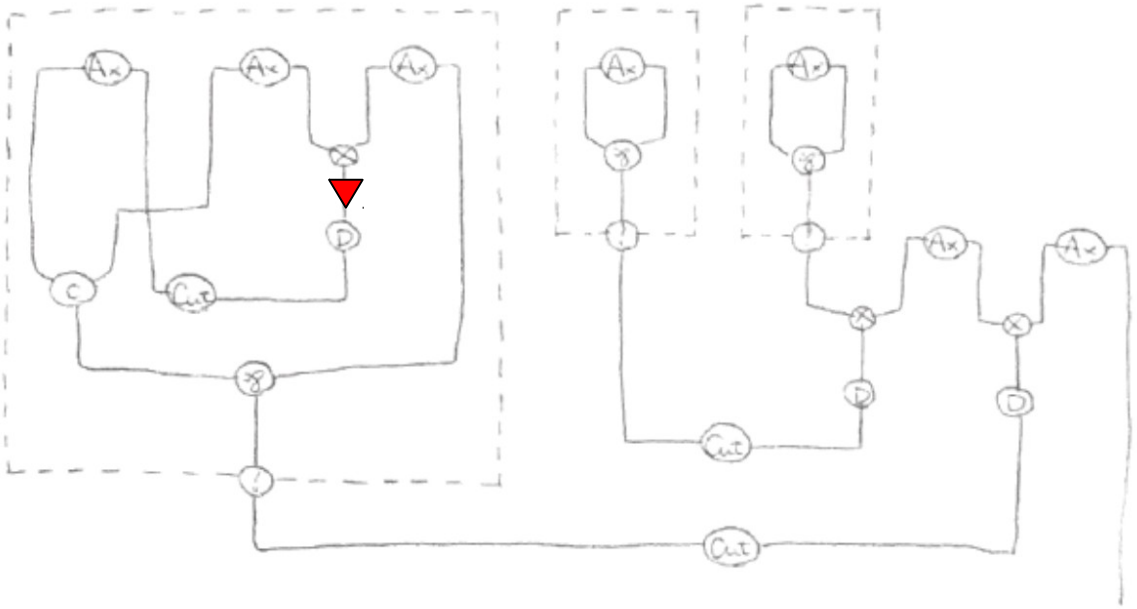
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

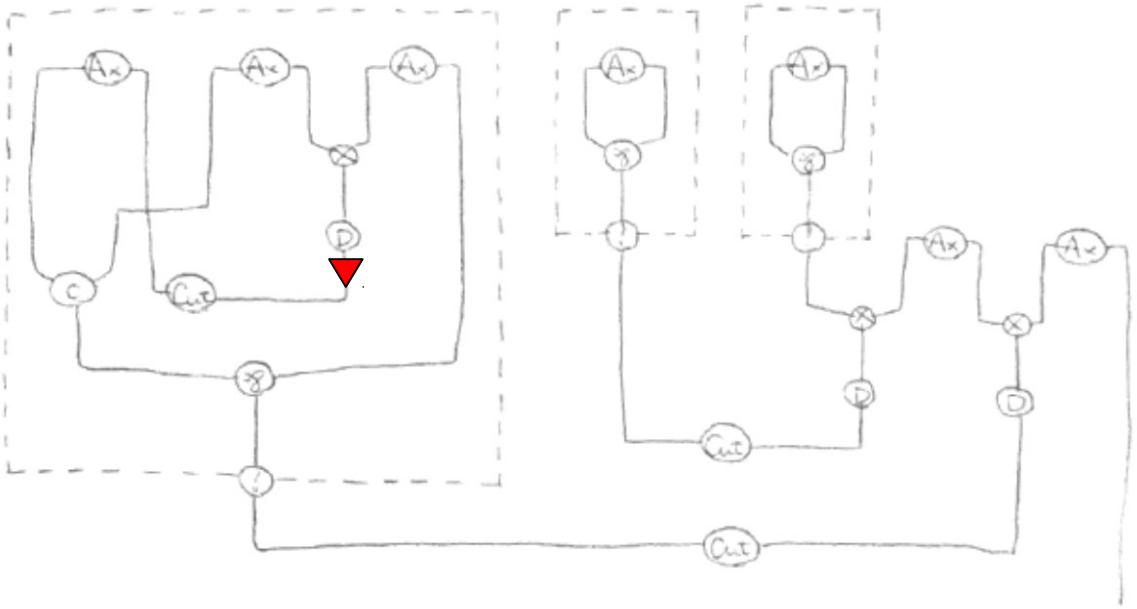
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

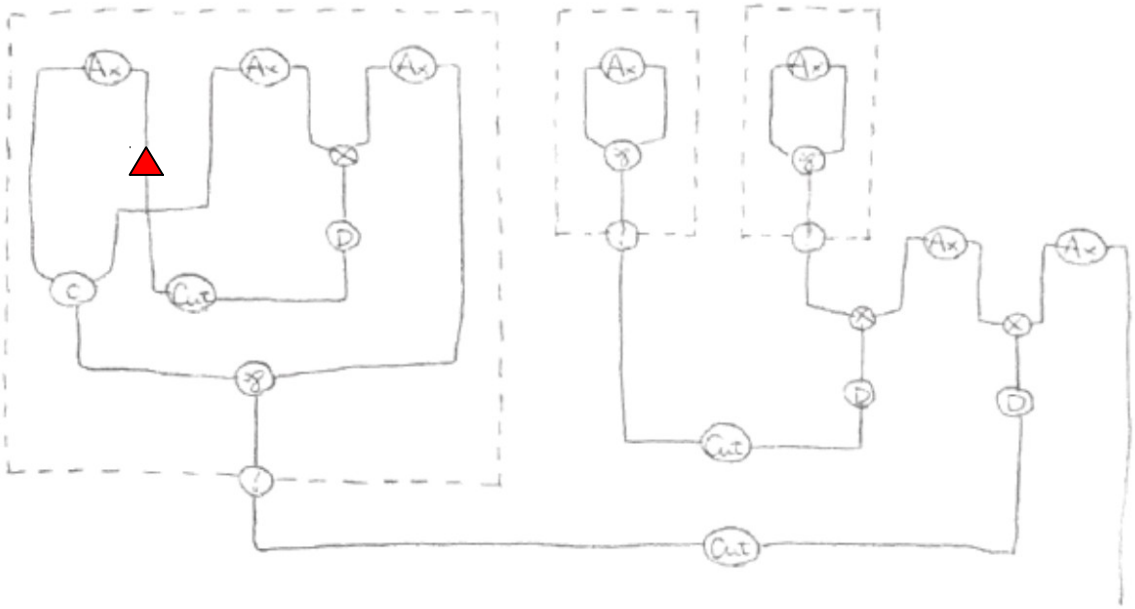
GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

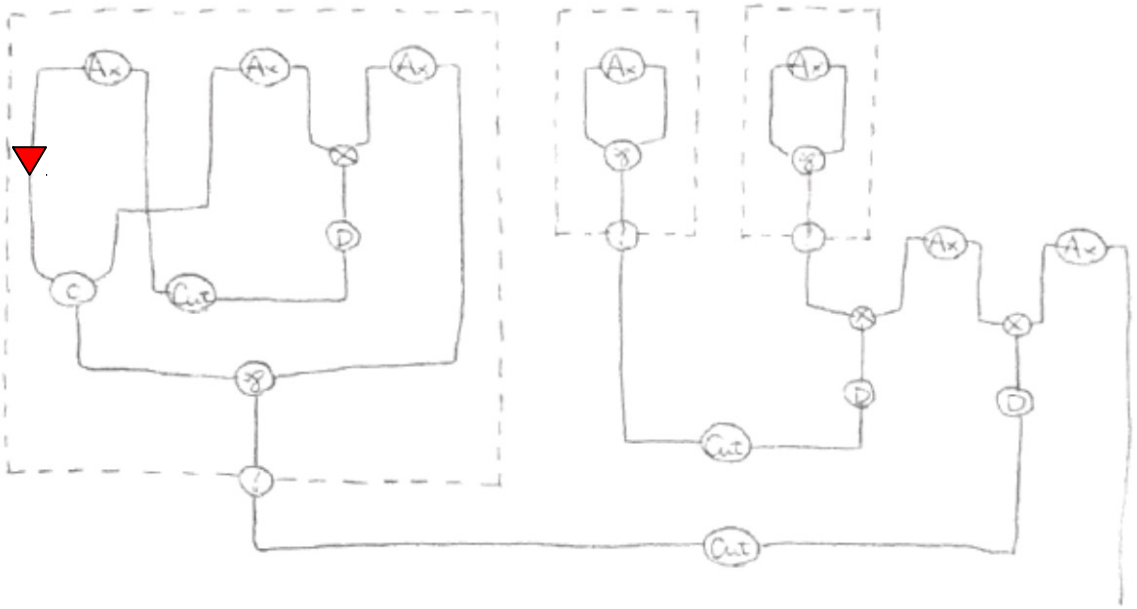
GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

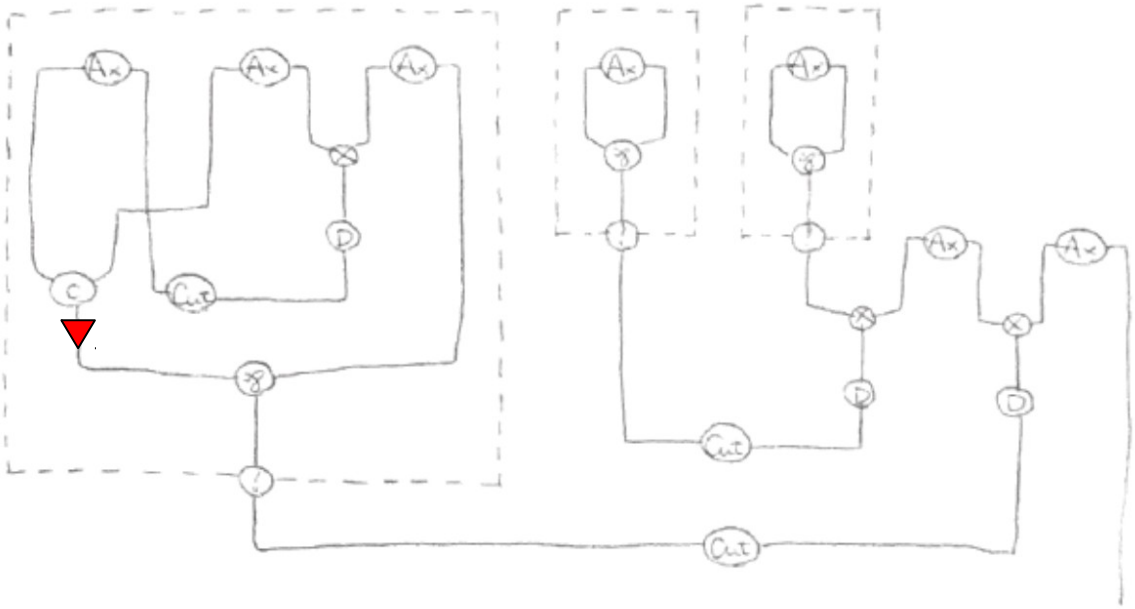
GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

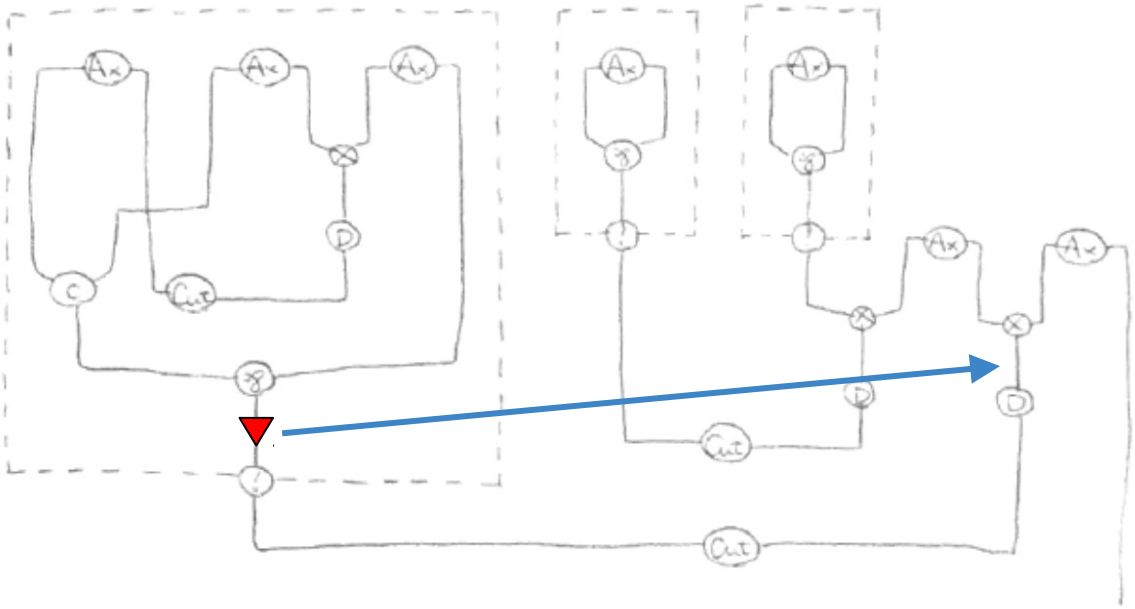
GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

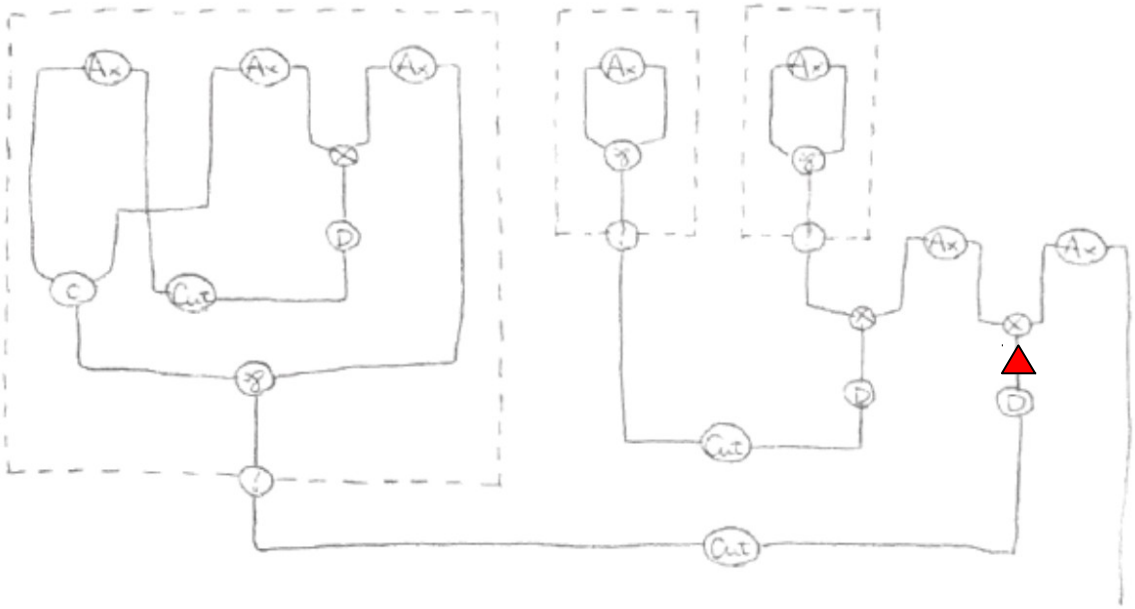
GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

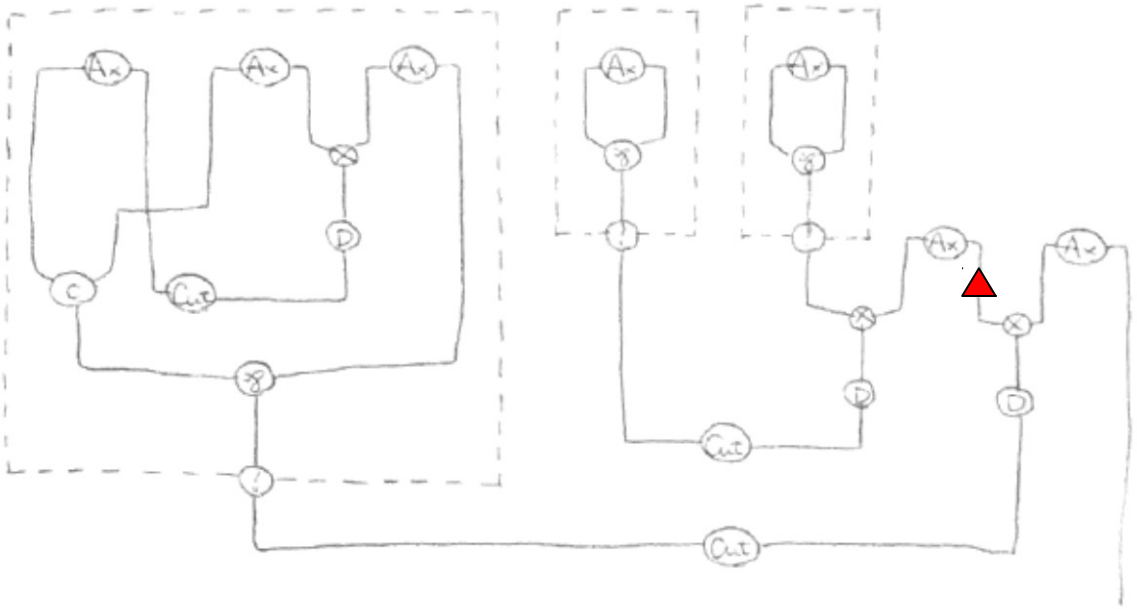
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

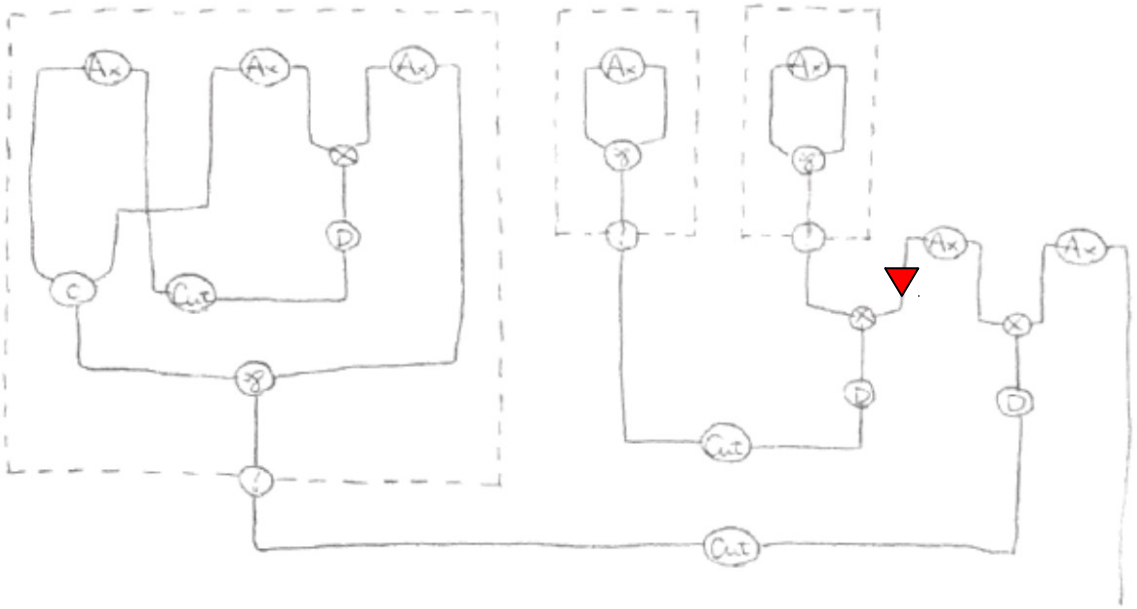
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

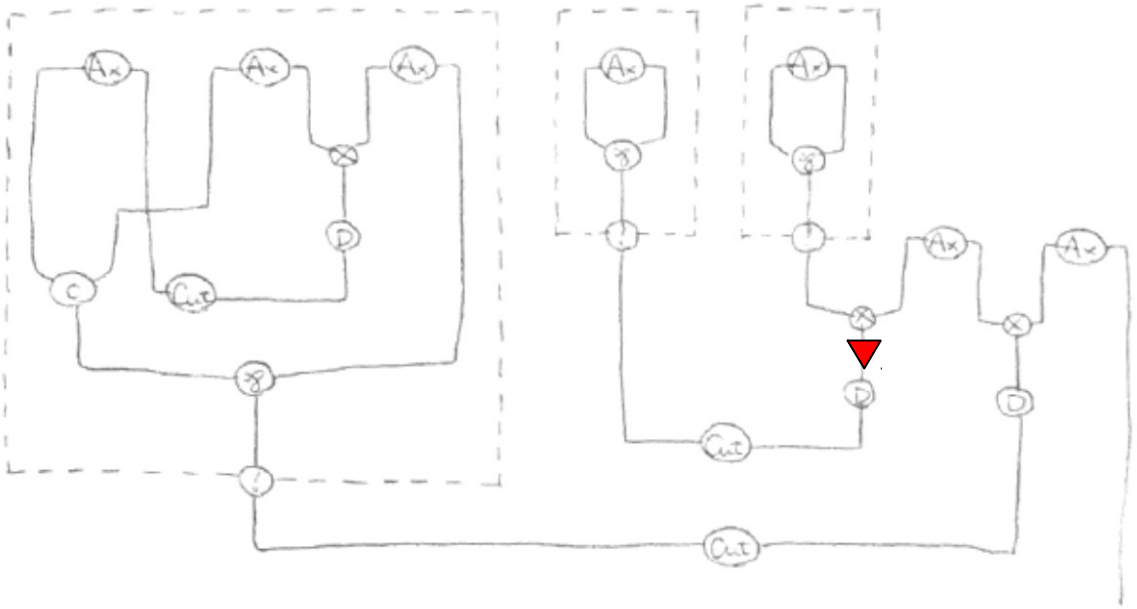
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

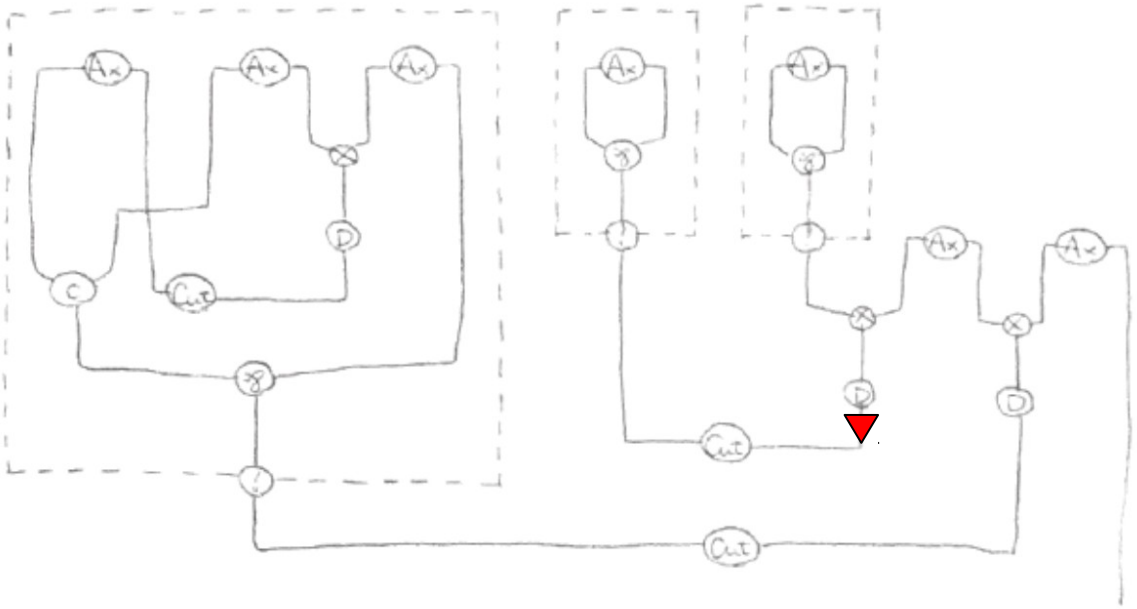
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

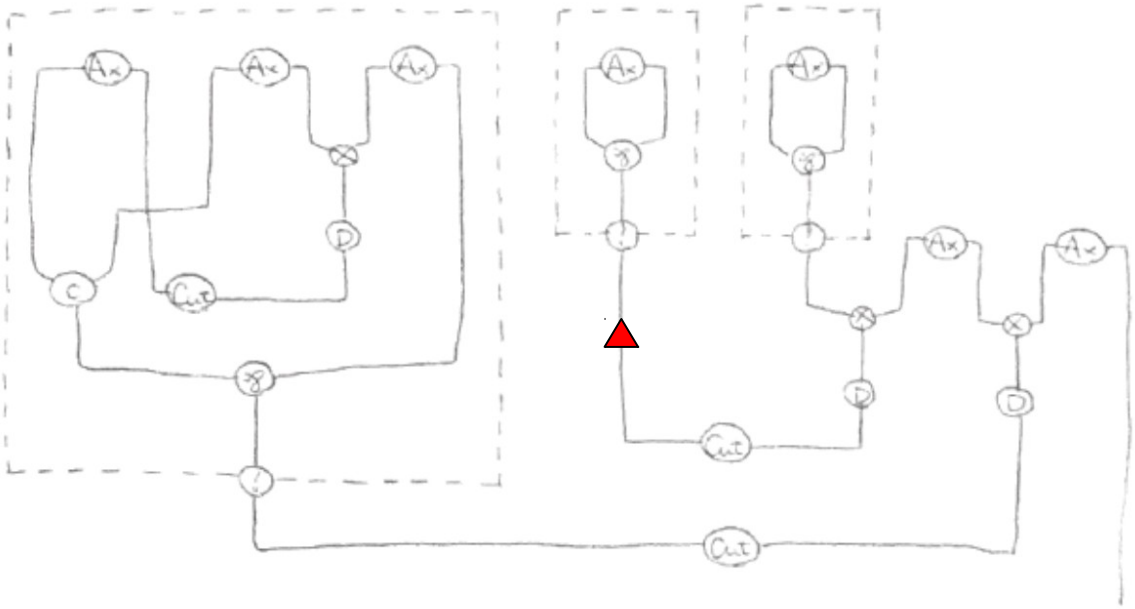
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

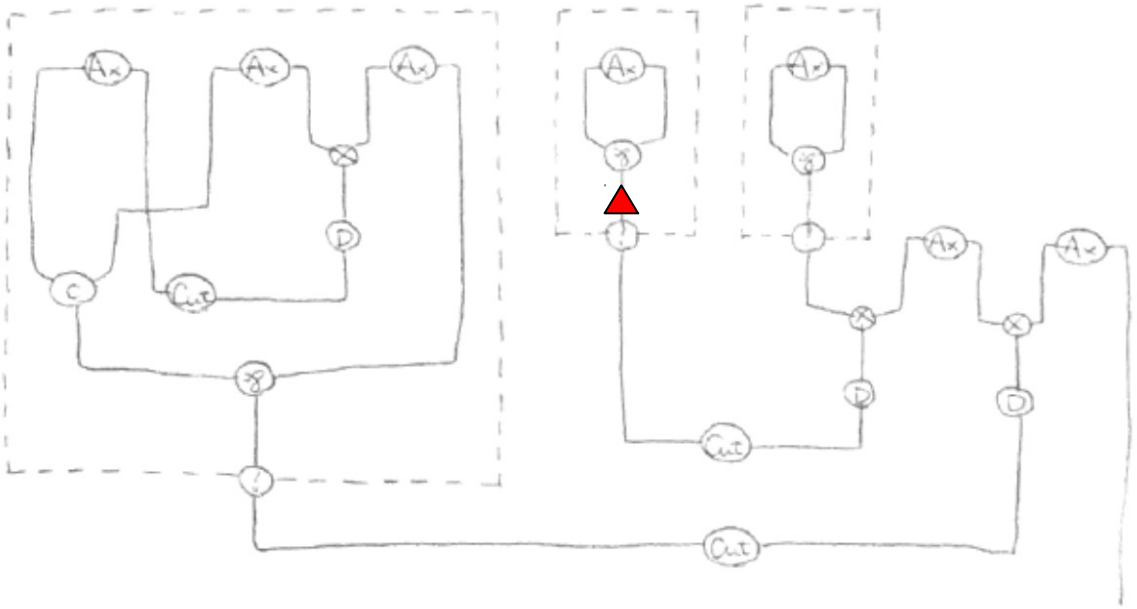
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

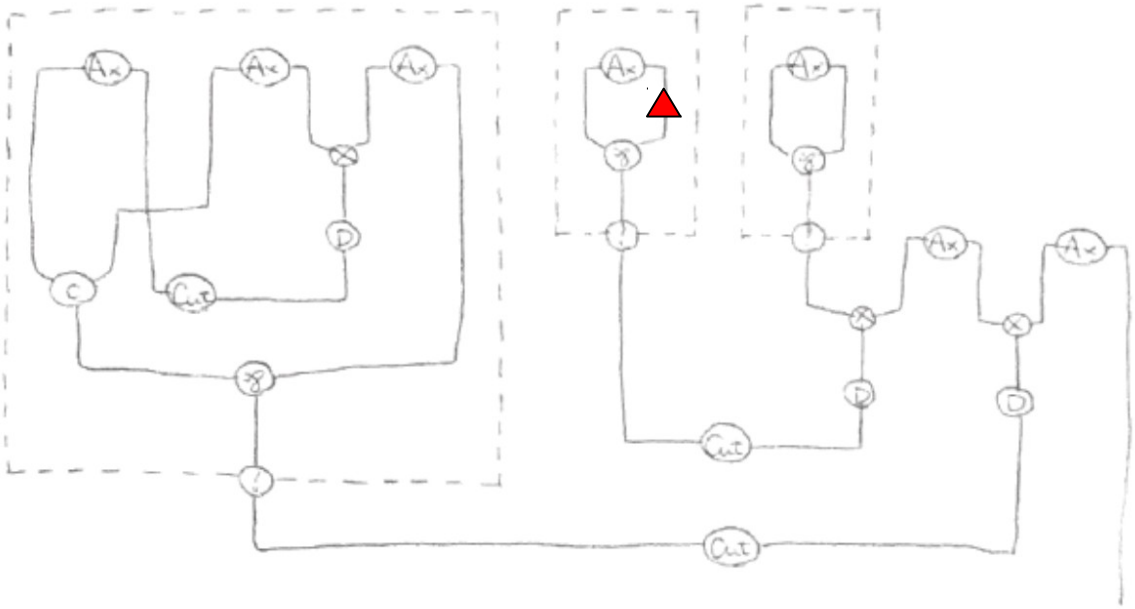
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

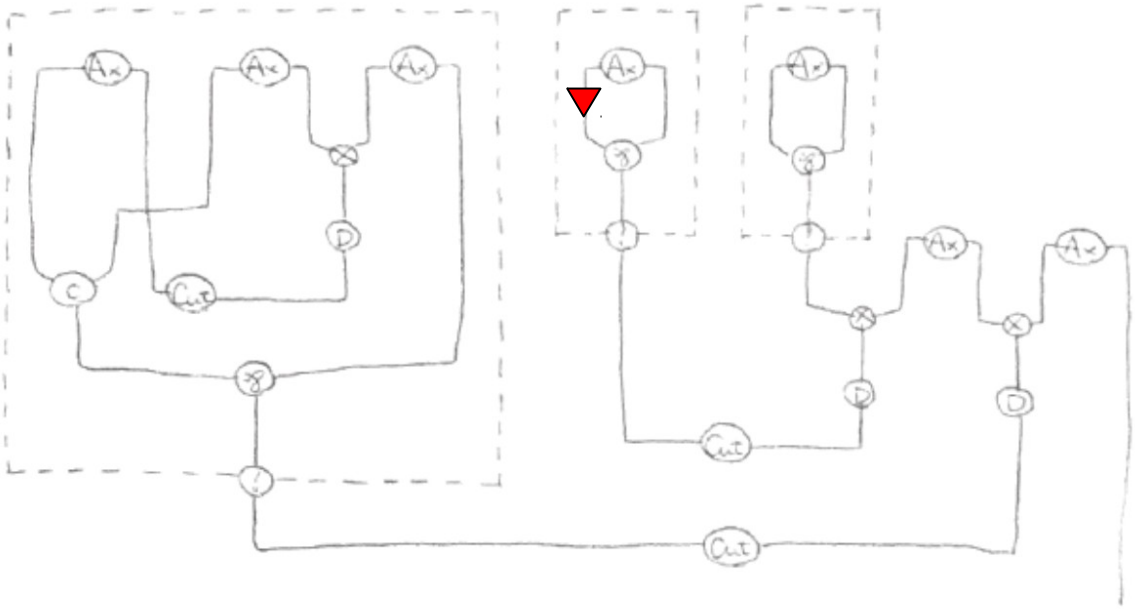
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

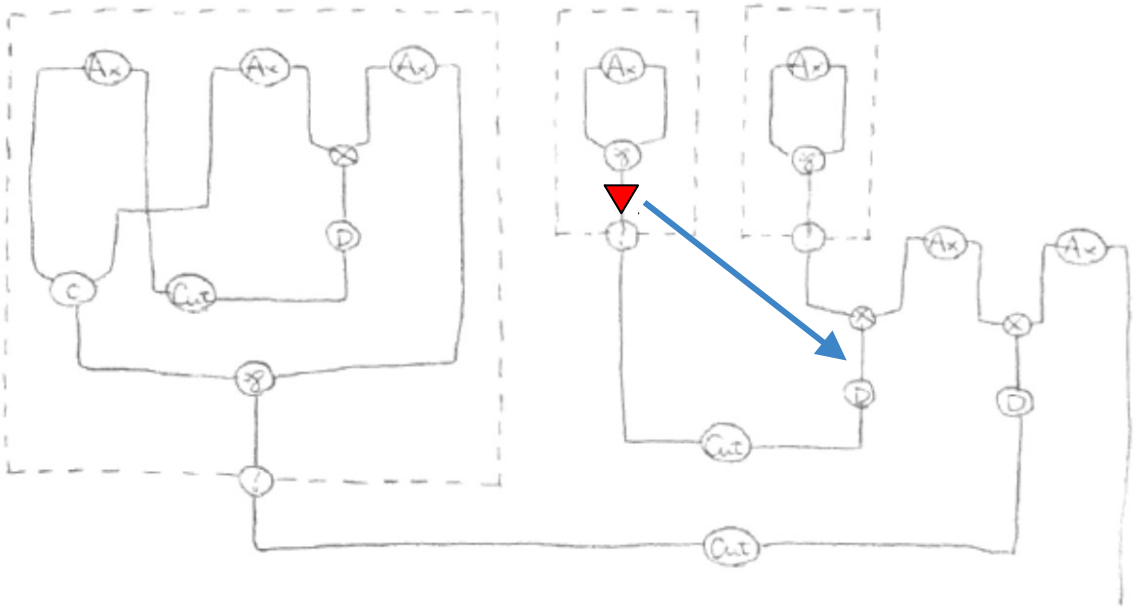
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

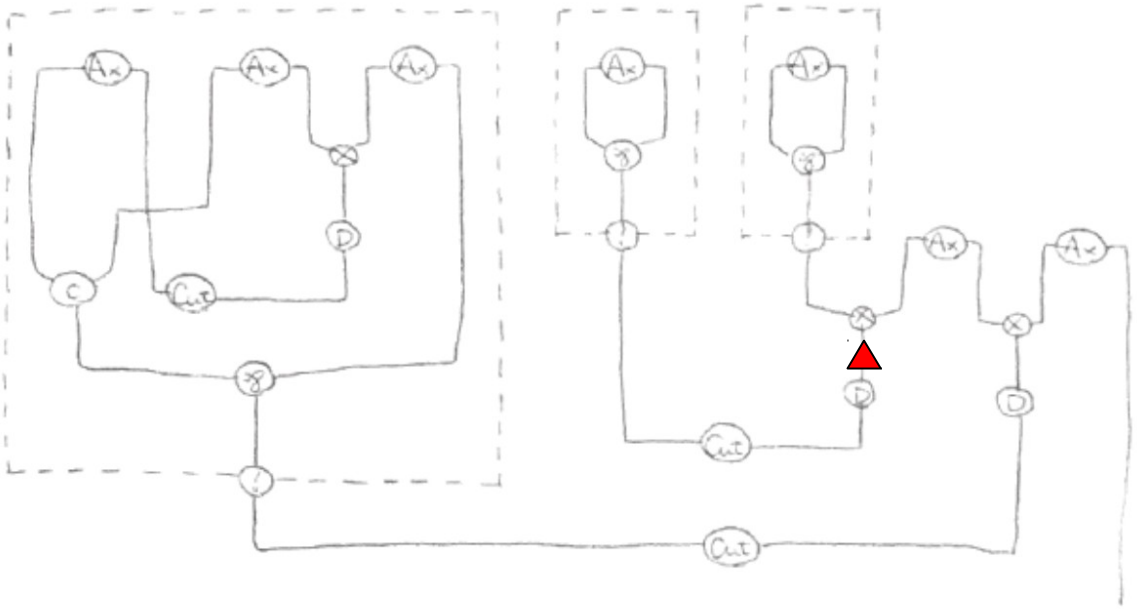
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

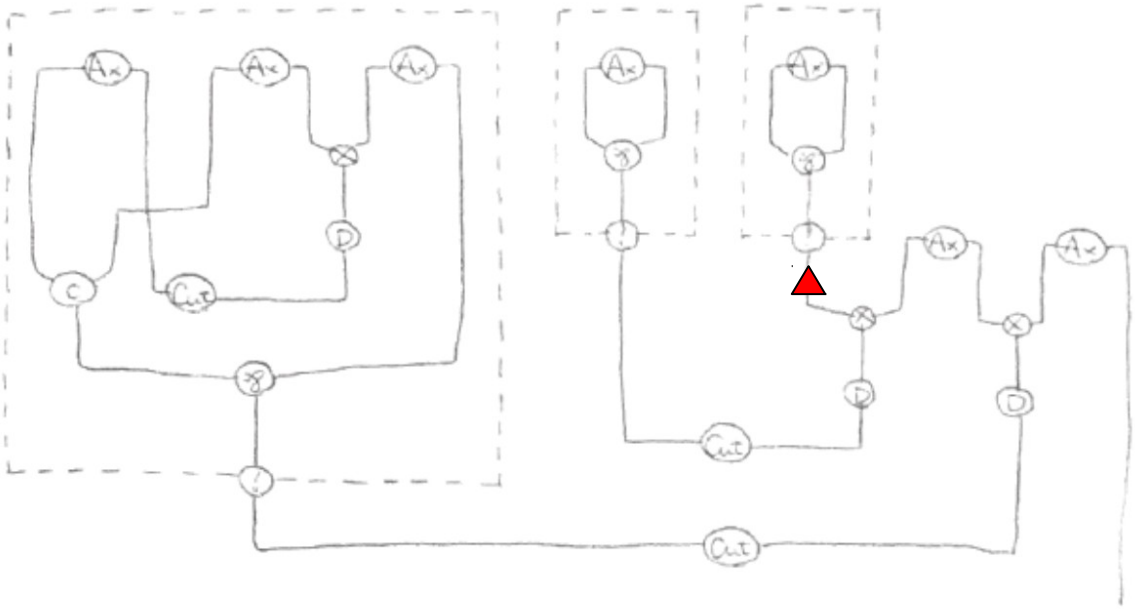
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

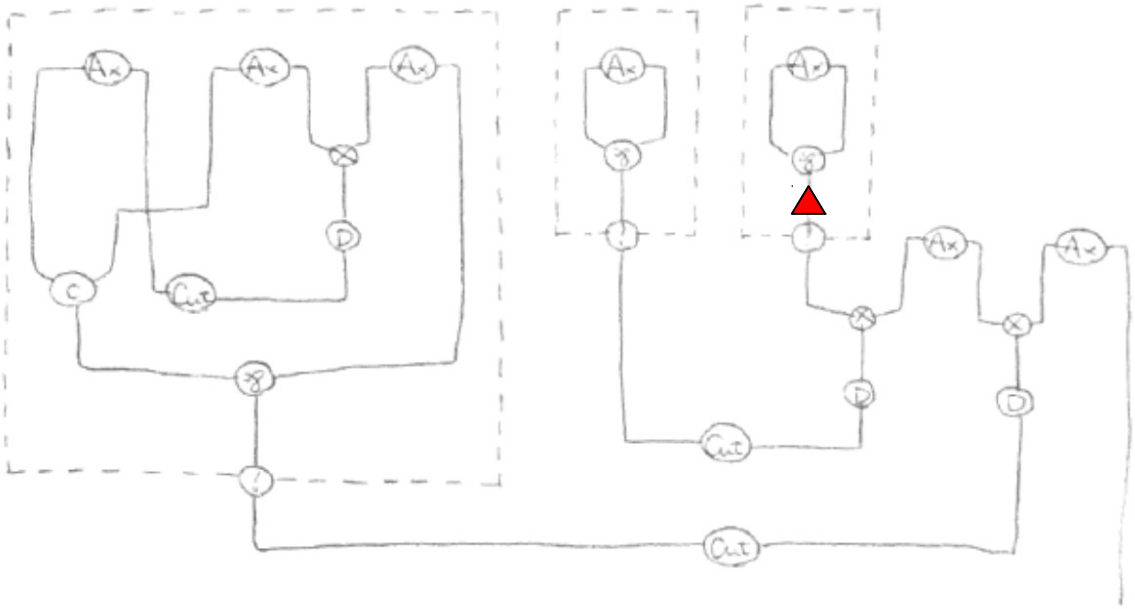
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

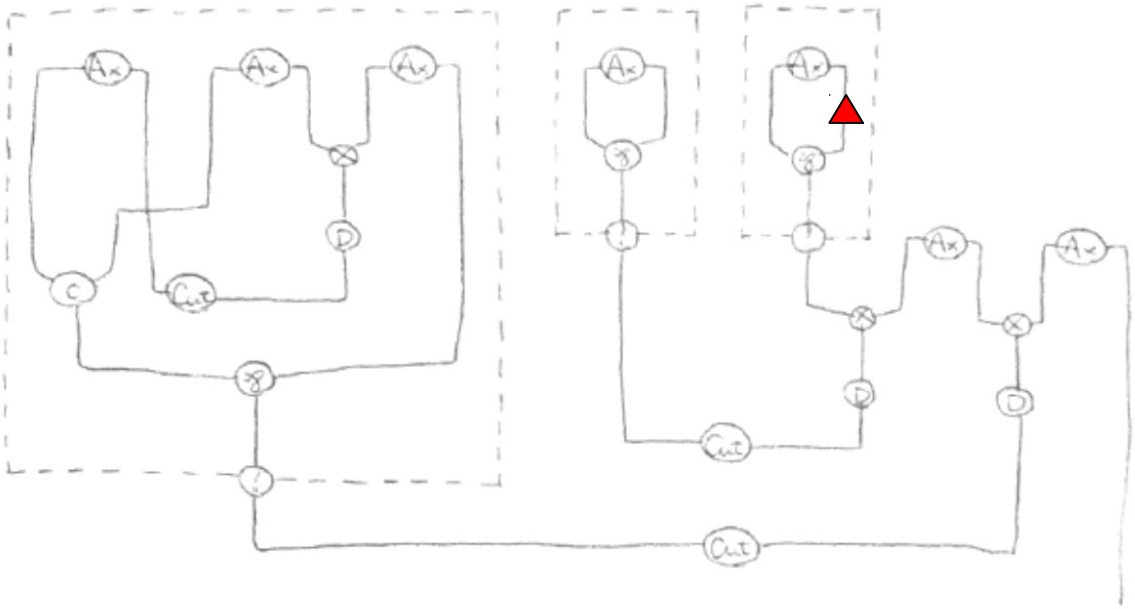
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

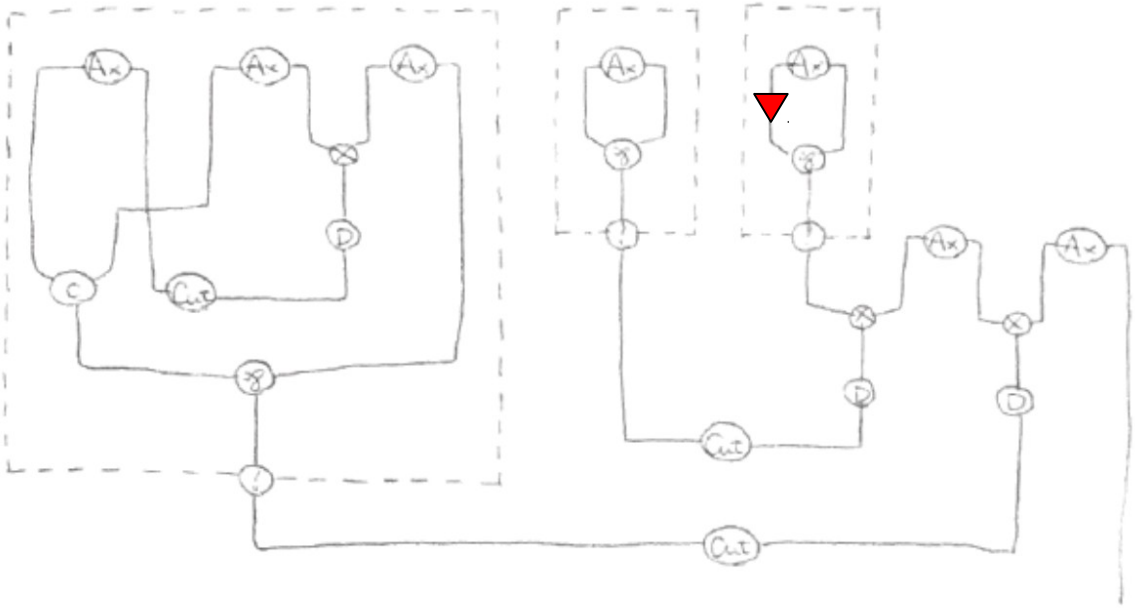


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

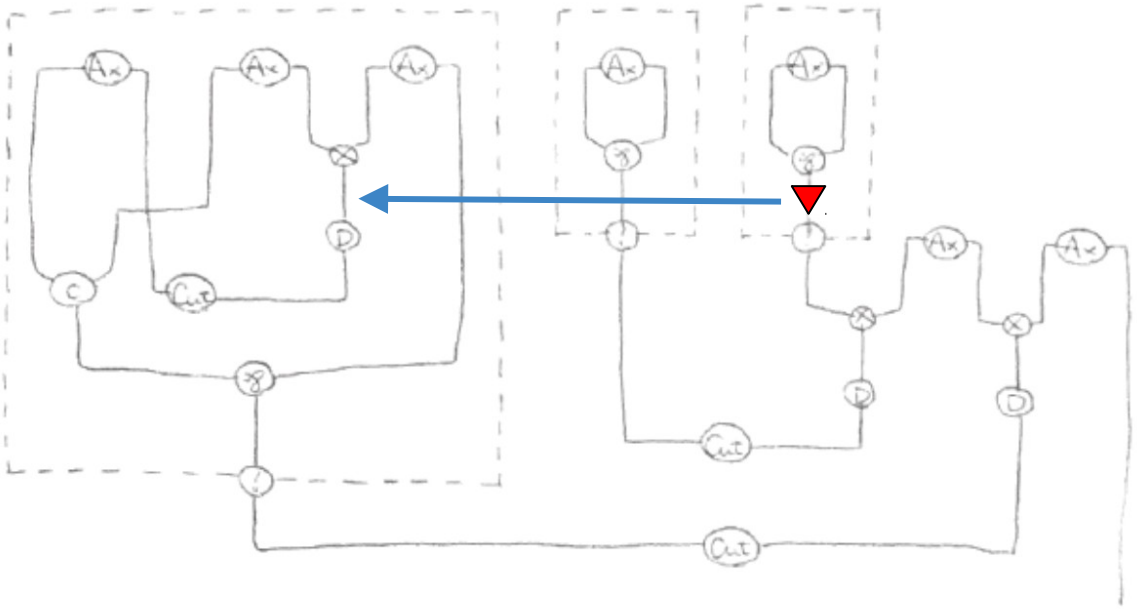


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

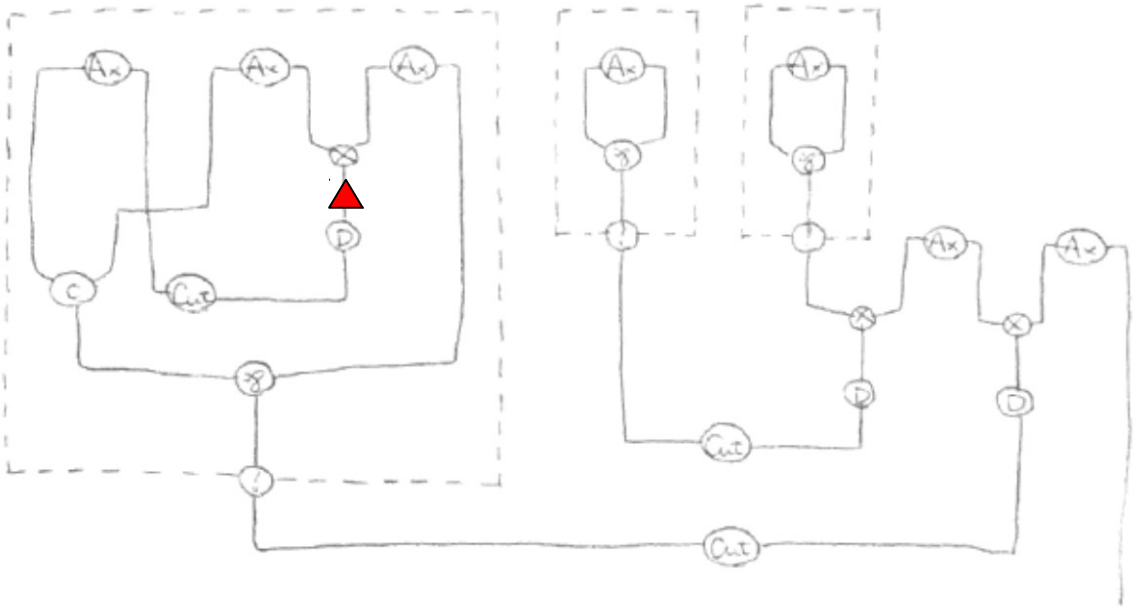
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

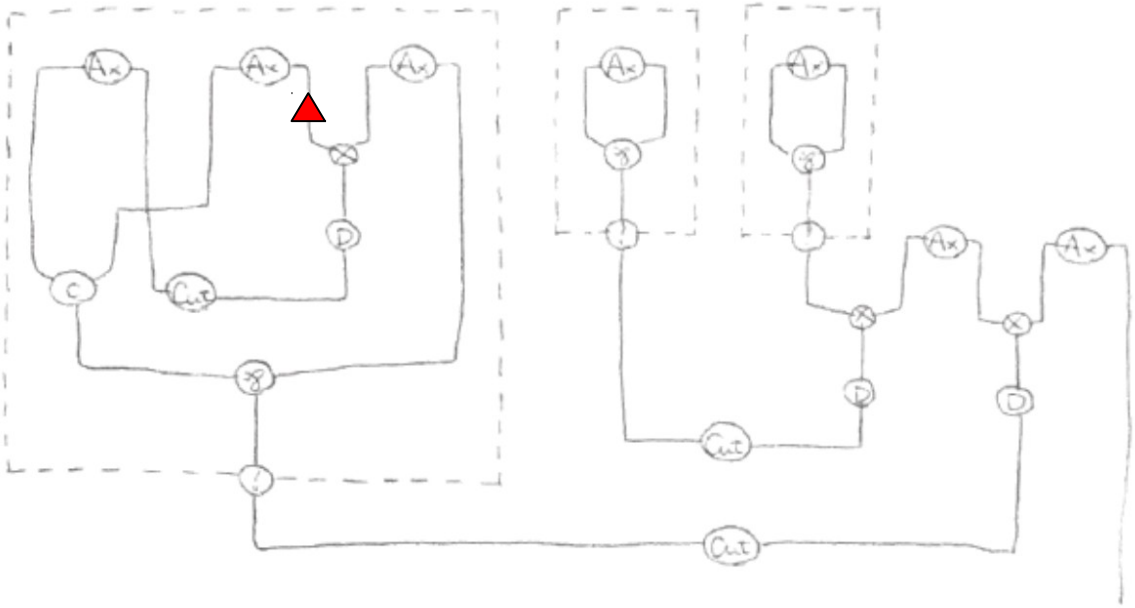


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

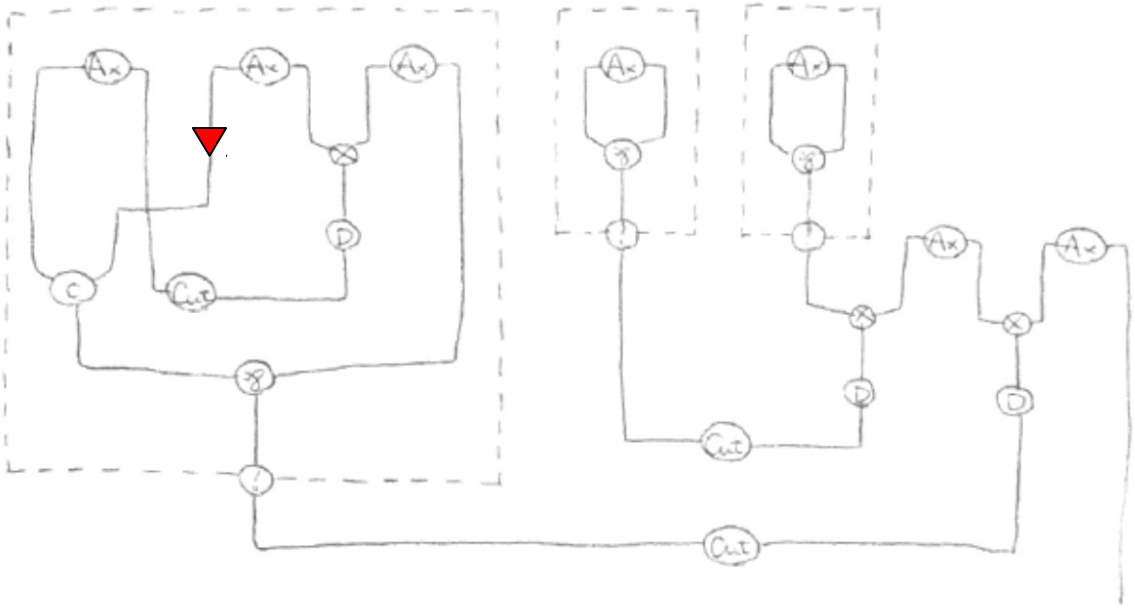


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

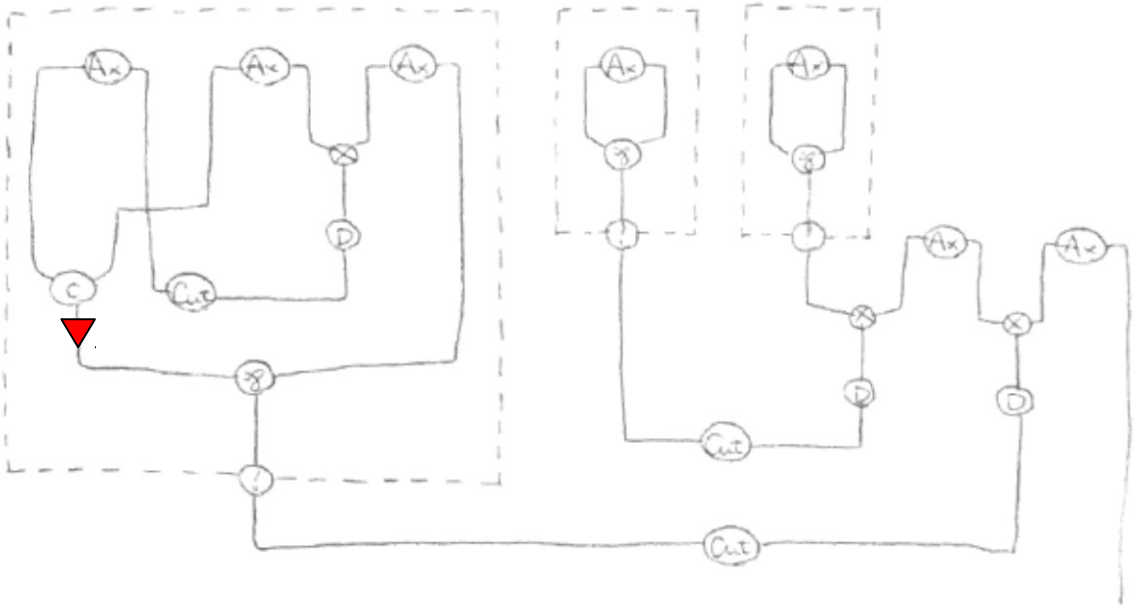


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

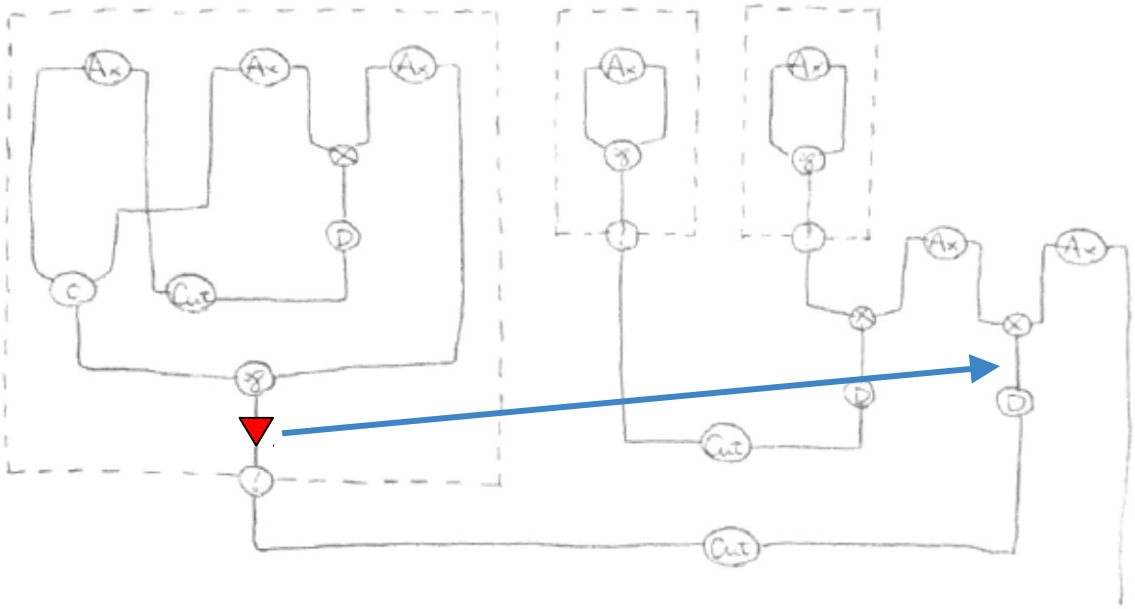


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

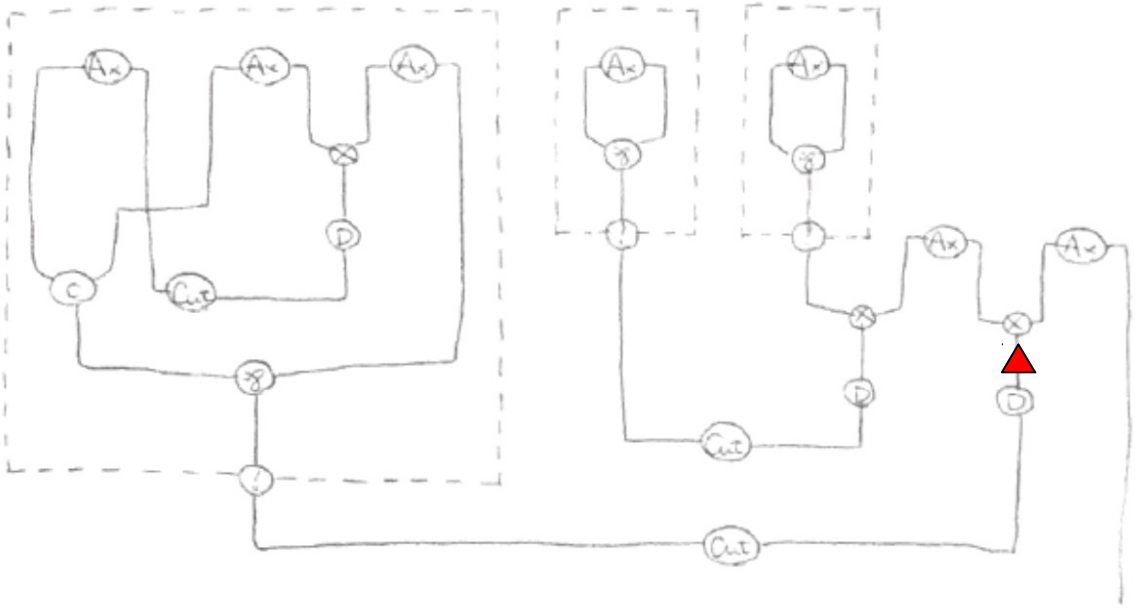


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

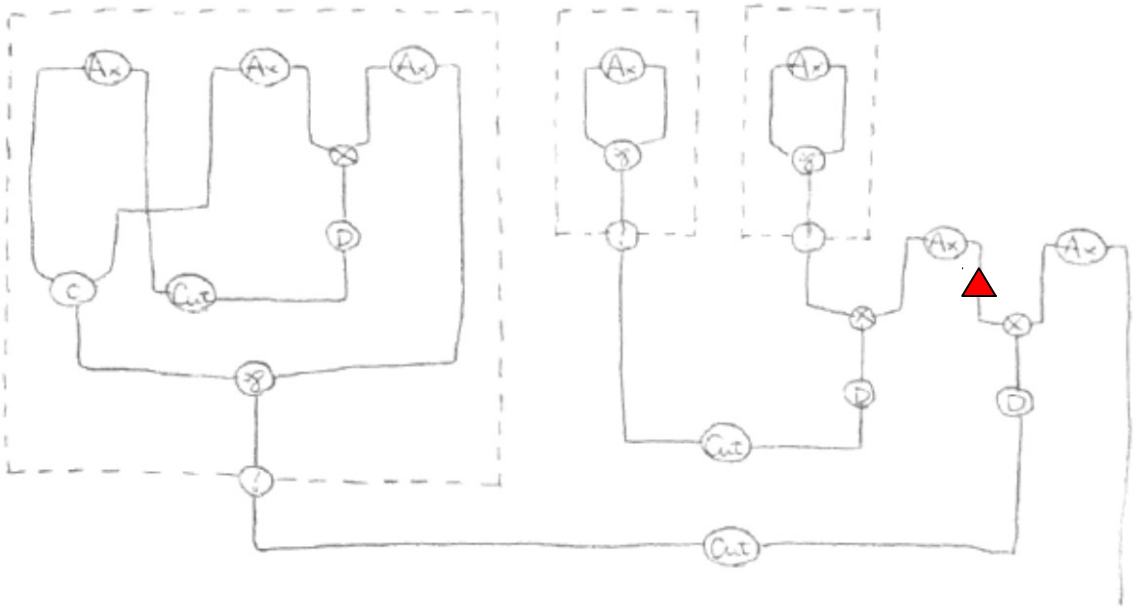


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

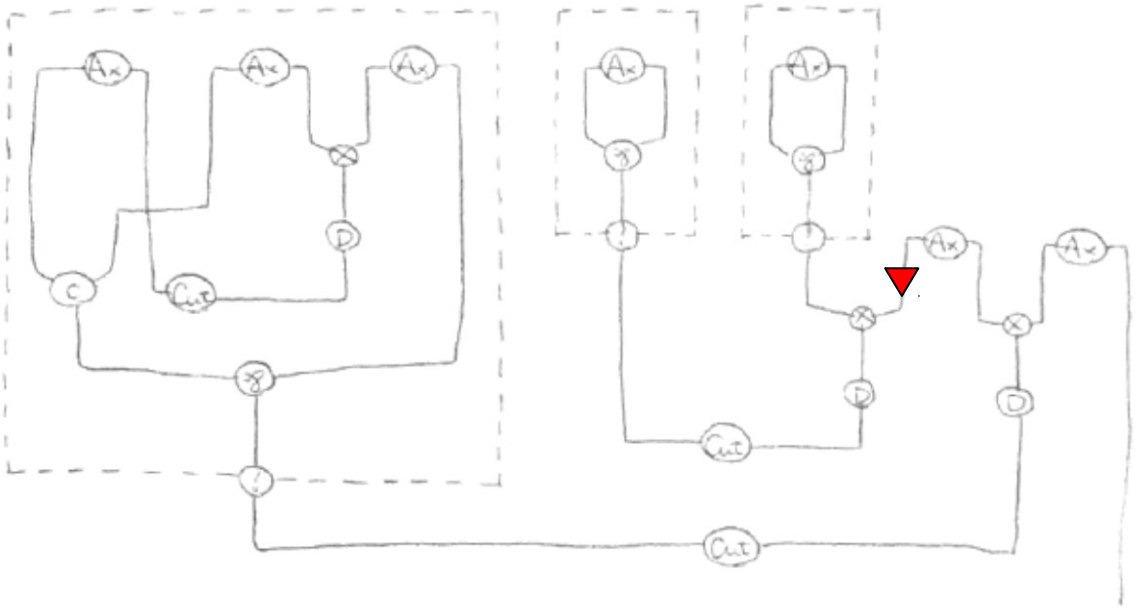
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

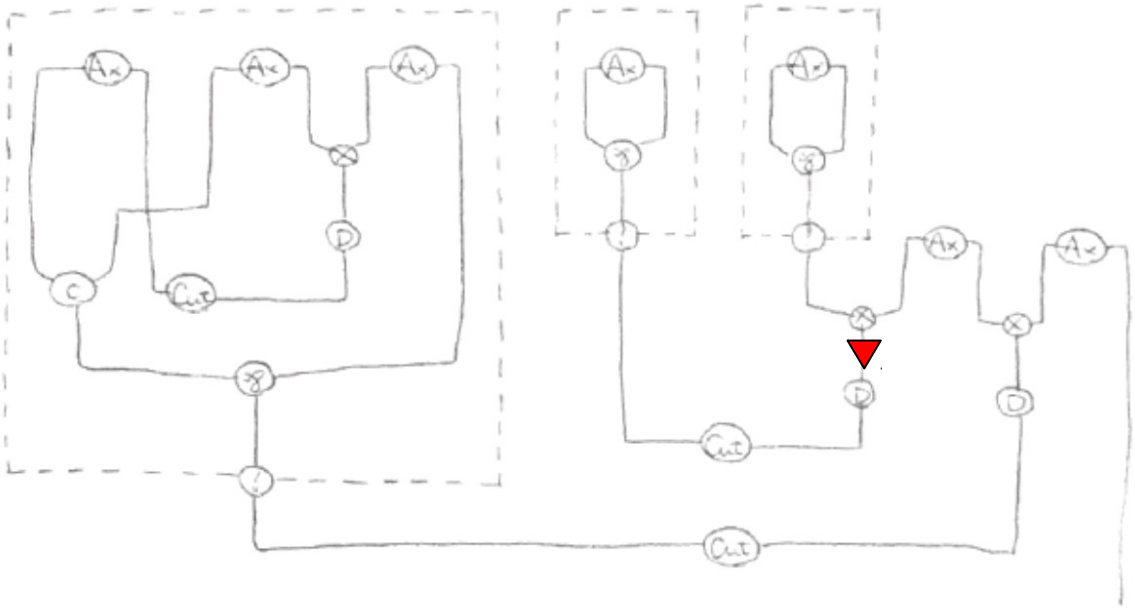


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

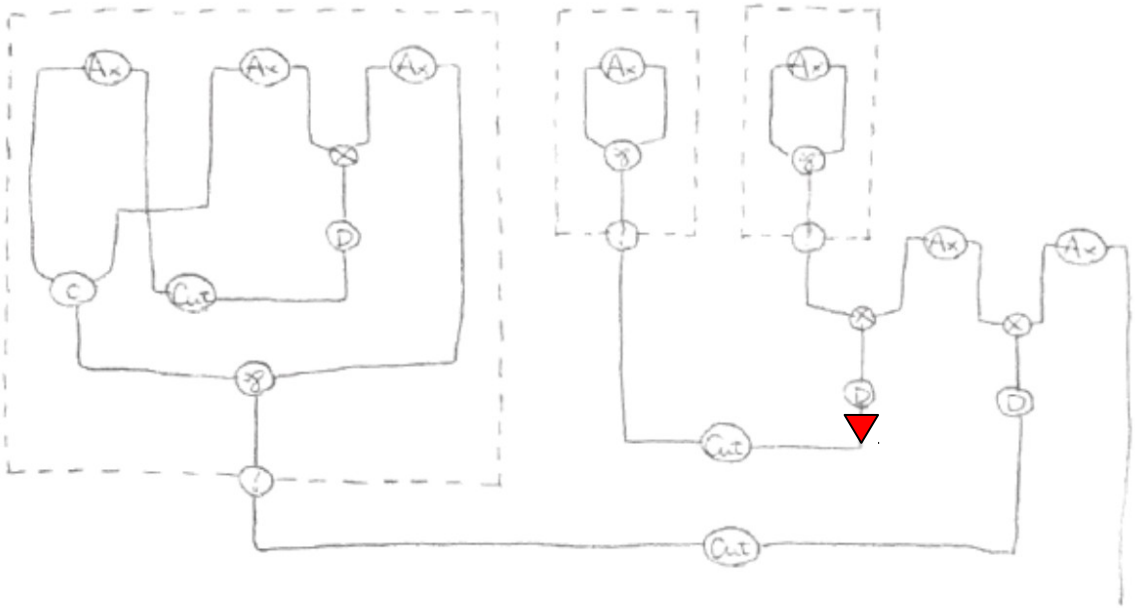


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

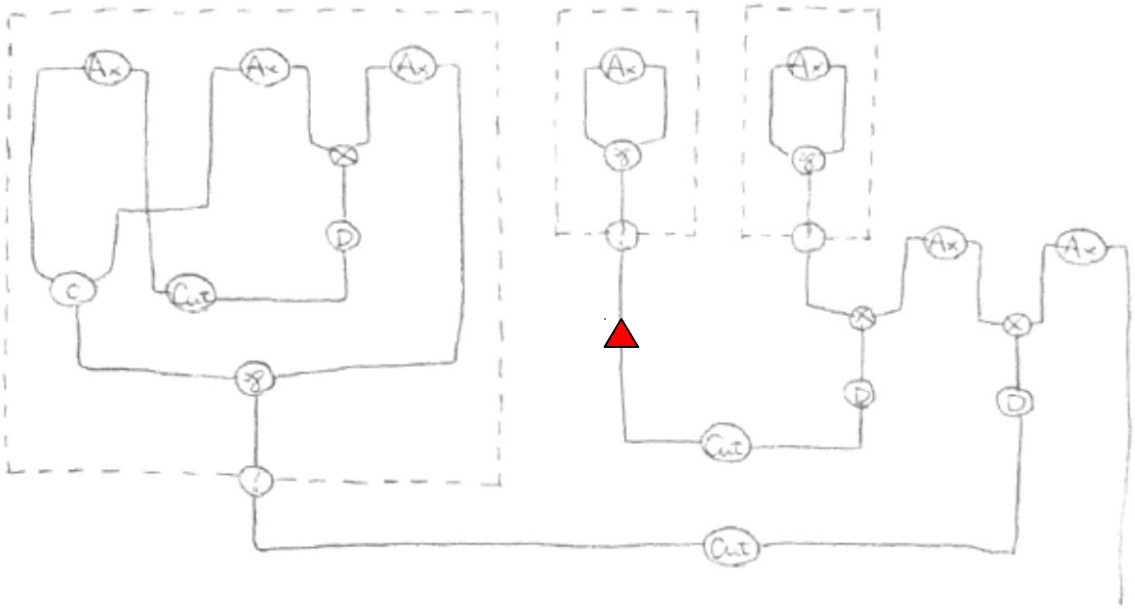


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

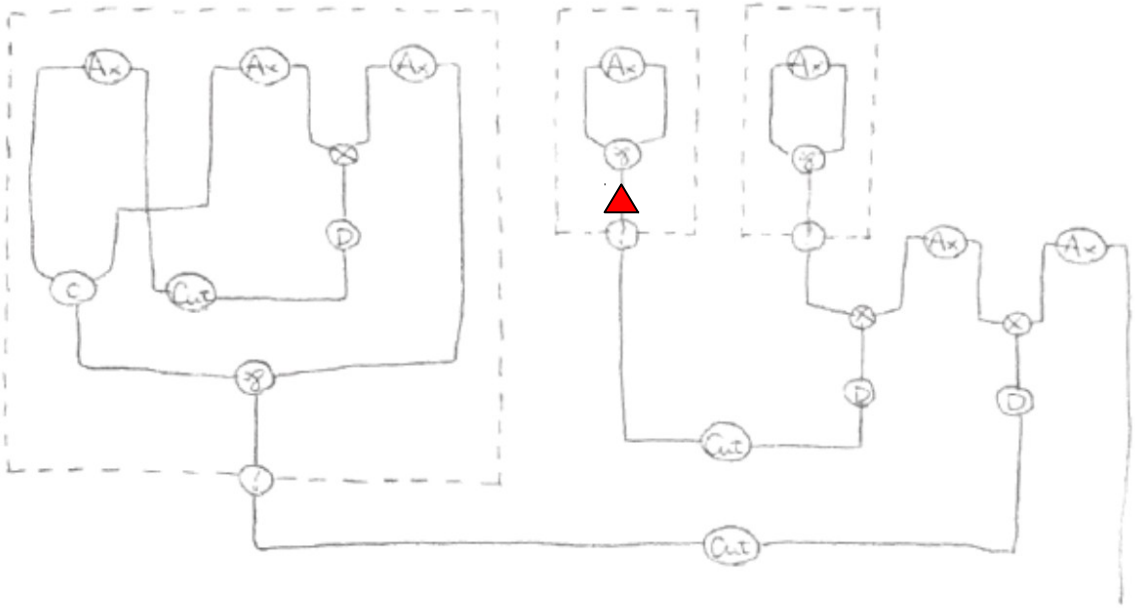


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

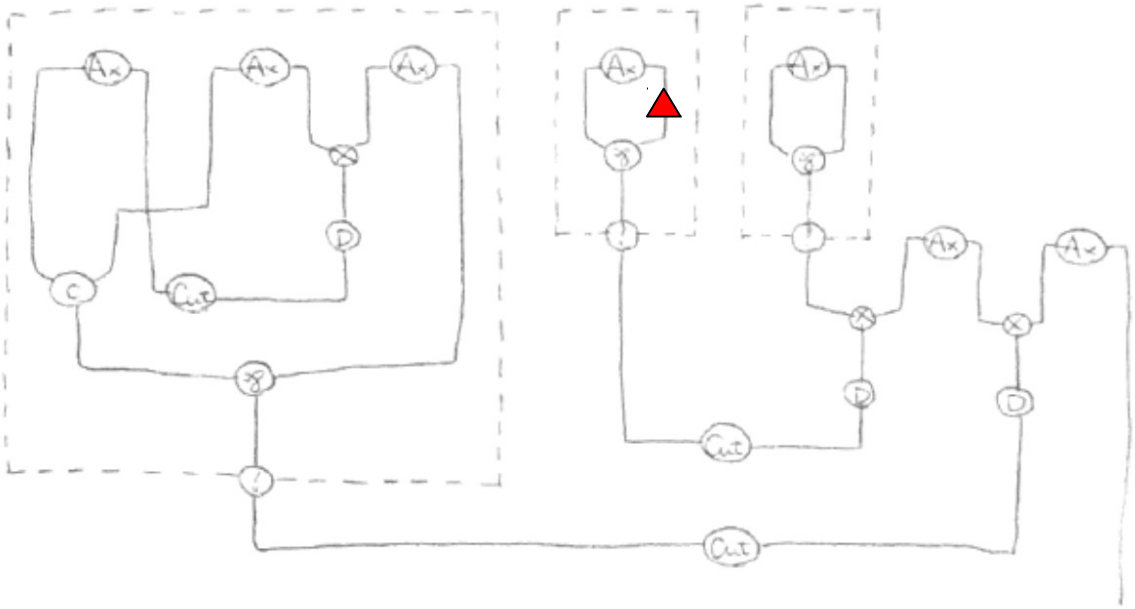


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

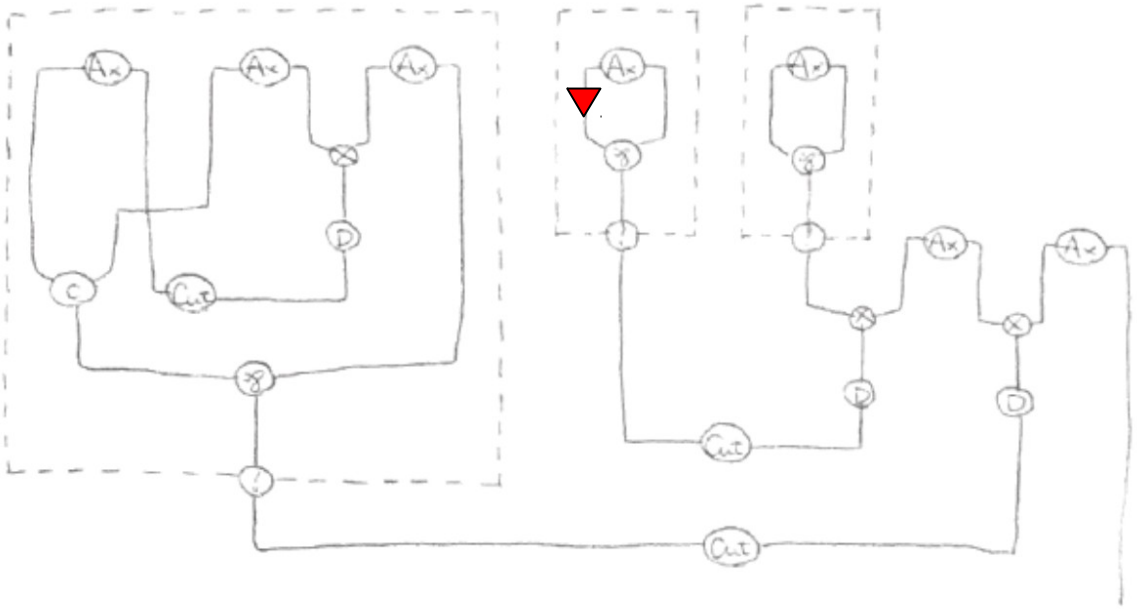


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

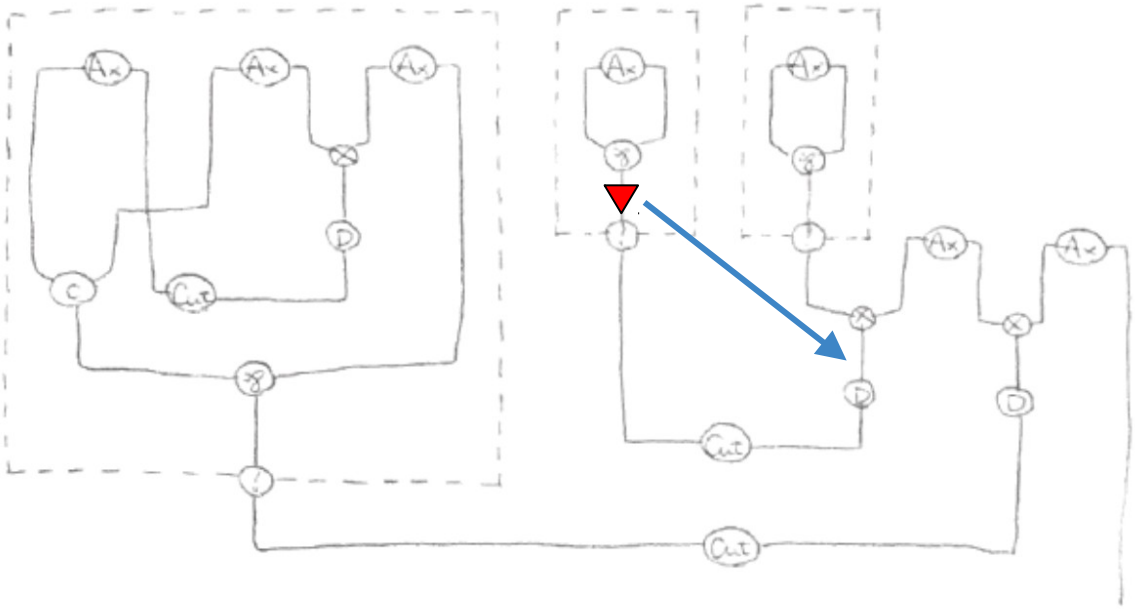


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

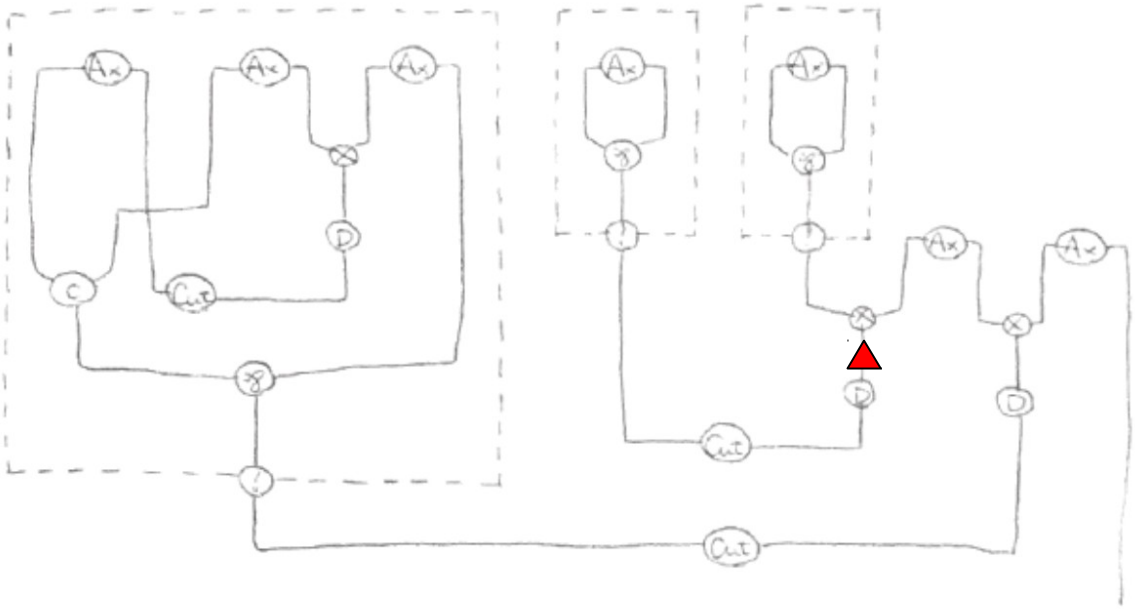


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

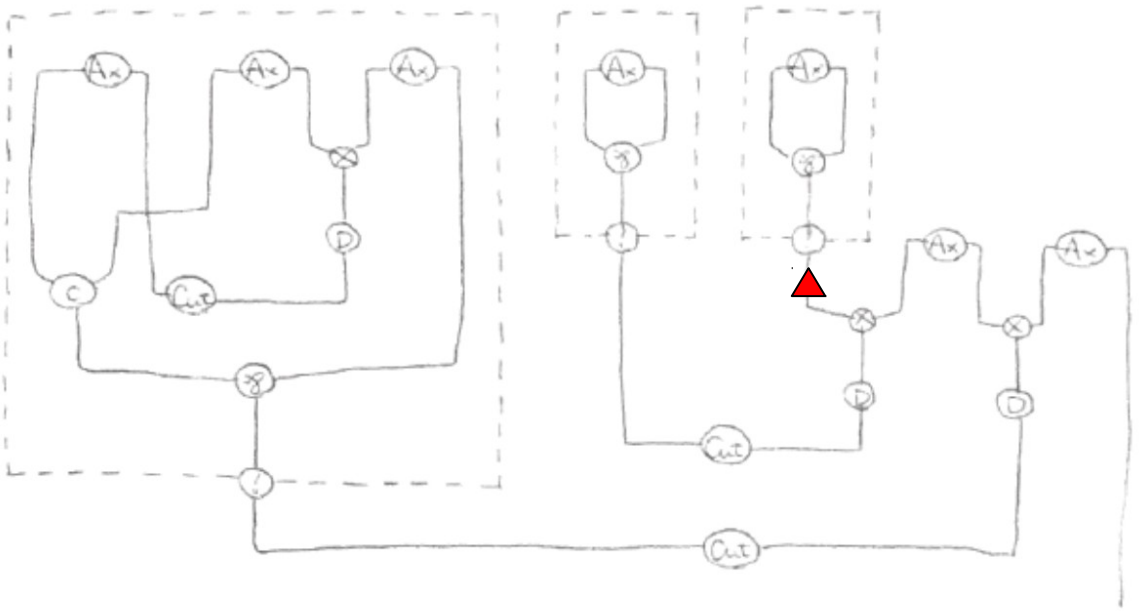


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

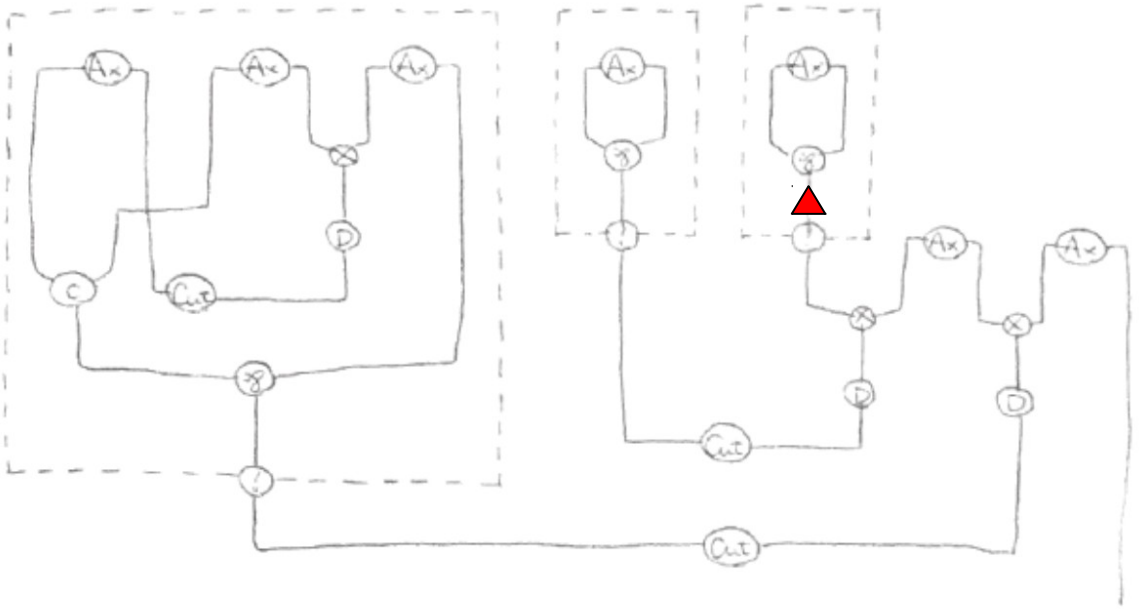


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

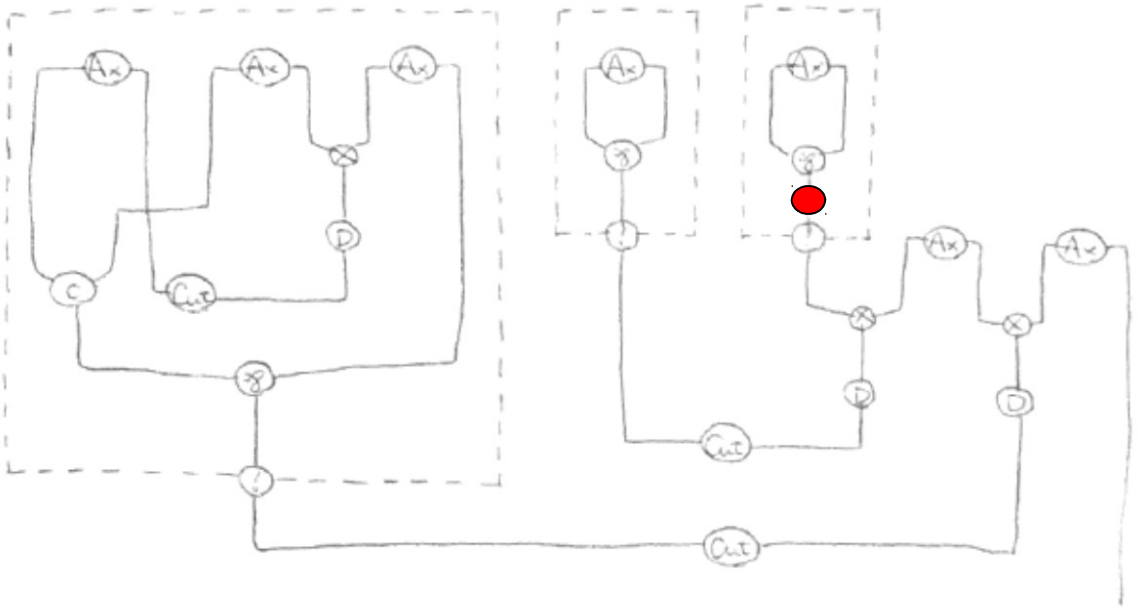


Interaction abstract machine (IAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

GoI token passing fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



Storeless abstract machine (SAM)

- [Danvy & Zerny '13]
- call-by-need evaluation
- syntactical environment

$$
\begin{array}{rll}
\text{Terms} & t ::= x \mid \lambda x.t \mid t\,t \mid t[x \leftarrow t] & \text{Pure terms} \quad \bar{t} ::= x \mid \lambda x.\bar{t} \mid \bar{t}\,\bar{t} \\
\text{Values} & v ::= \lambda x.t & \text{Pure values} \quad \bar{v} ::= \lambda x.\bar{t} \\
\text{Evaluation contexts} & E ::= \langle\cdot\rangle \mid E\,\bar{t} \mid E[x \leftarrow \bar{t}] \mid E\langle x\rangle[x \leftarrow E] & \\
\text{Substitution contexts} & A ::= \langle\cdot\rangle \mid A[x \leftarrow \bar{t}] &
\end{array}
$$

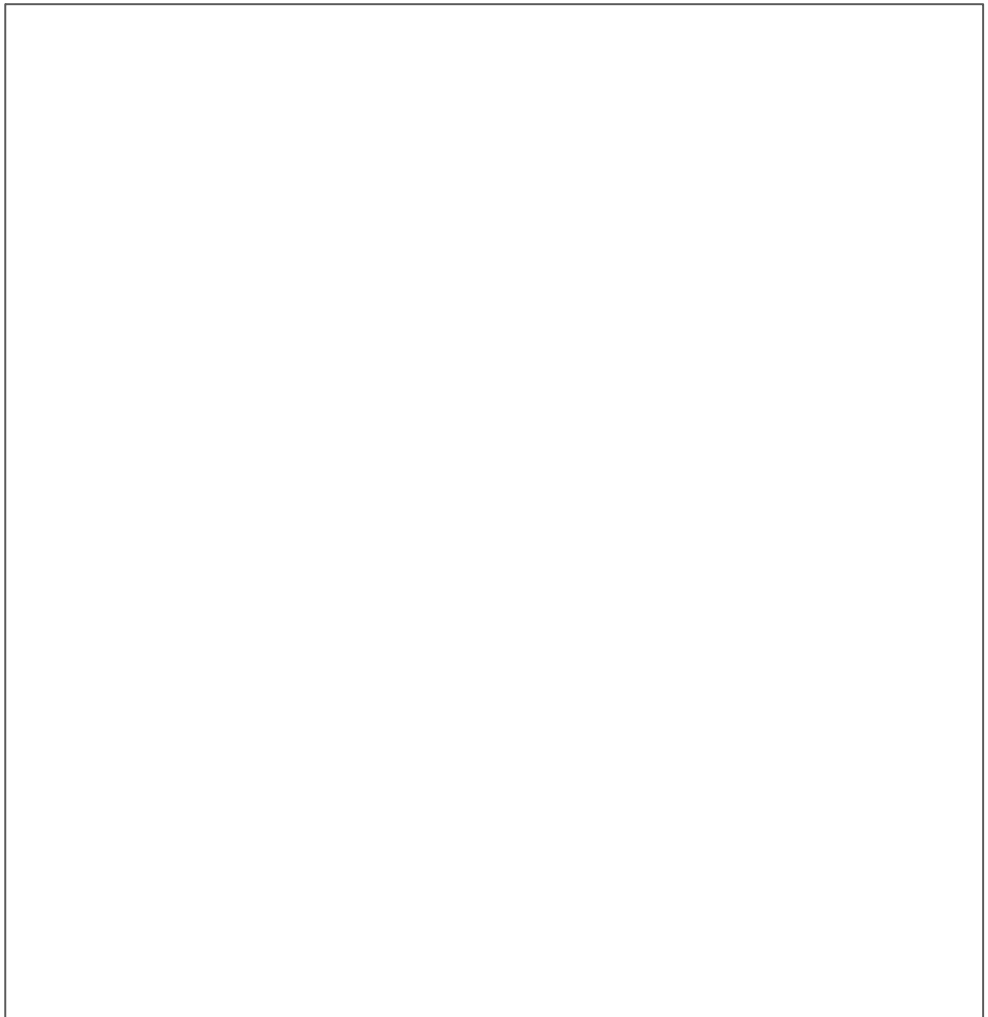
$$
\begin{array}{rll}
(\bar{t}\,\bar{u},\, E)_{term} \to_{\mathsf{o}} (\bar{t},\, E\langle\langle\cdot\rangle\,\bar{u}\rangle)_{term} & & (8) \\
(x,\, E_1\langle E_2[x \leftarrow \bar{t}]\rangle)_{term} \to_{\mathsf{o}} (\bar{t},\, E_1\langle E_2\langle x\rangle[x \leftarrow \langle\cdot\rangle]\rangle)_{term} & & (9) \\
(\bar{v},\, E)_{term} \to_{\mathsf{o}} (\bar{v},\, E)_{ctxt} & & (10) \\
(\lambda x.\bar{t},\, E\langle A\,\bar{u}\rangle)_{ctxt} \to_{\mathsf{b}} (\bar{t},\, E\langle A\langle\langle\cdot\rangle[x \leftarrow \bar{u}]\rangle\rangle)_{term} & & (11) \\
(\bar{v},\, E_1\langle E_2\langle x\rangle[x \leftarrow A]\rangle)_{ctxt} \to_{\mathsf{s}} (\bar{v}^{\approx},\, E_1\langle A\langle E_2[x \leftarrow \bar{v}]\rangle\rangle)_{ctxt} & (\text{if } x \in \mathrm{FV}_{\emptyset}(E_2)) & (12) \\
(\bar{v},\, E_1\langle E_2\langle x\rangle[x \leftarrow A]\rangle)_{ctxt} \to_{\mathsf{s}} (\bar{v},\, E_1\langle A\langle E_2\rangle\rangle)_{ctxt} & (\text{if } x \notin \mathrm{FV}_{\emptyset}(E_2)) & (13)
\end{array}
$$

Figure 5 Call-by-need Storeless Abstract Machine (SAM)

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



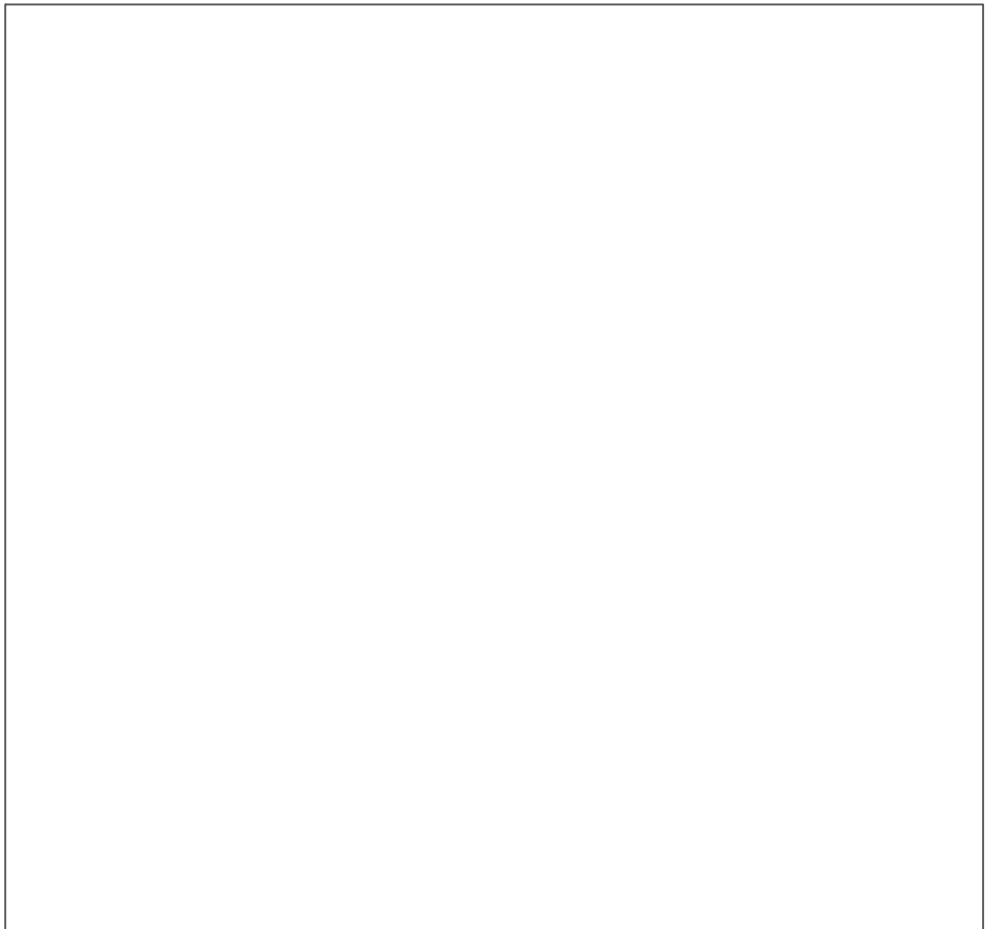
redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$



redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

redex searching
term rewriting

Storeless abstract machine (SAM)

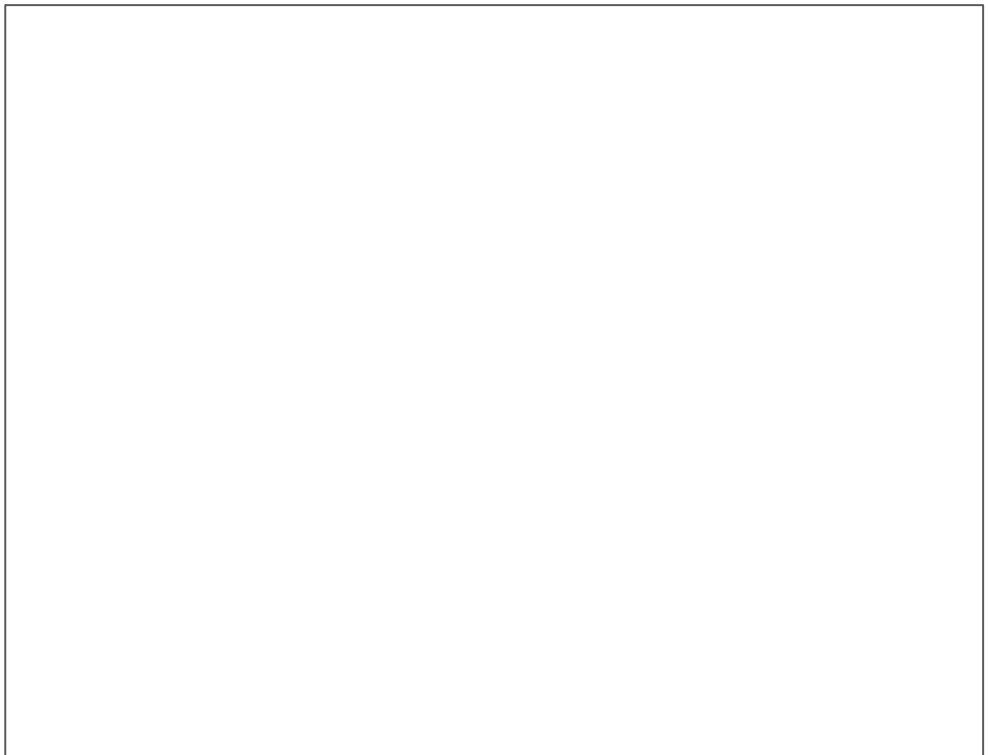
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$



redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

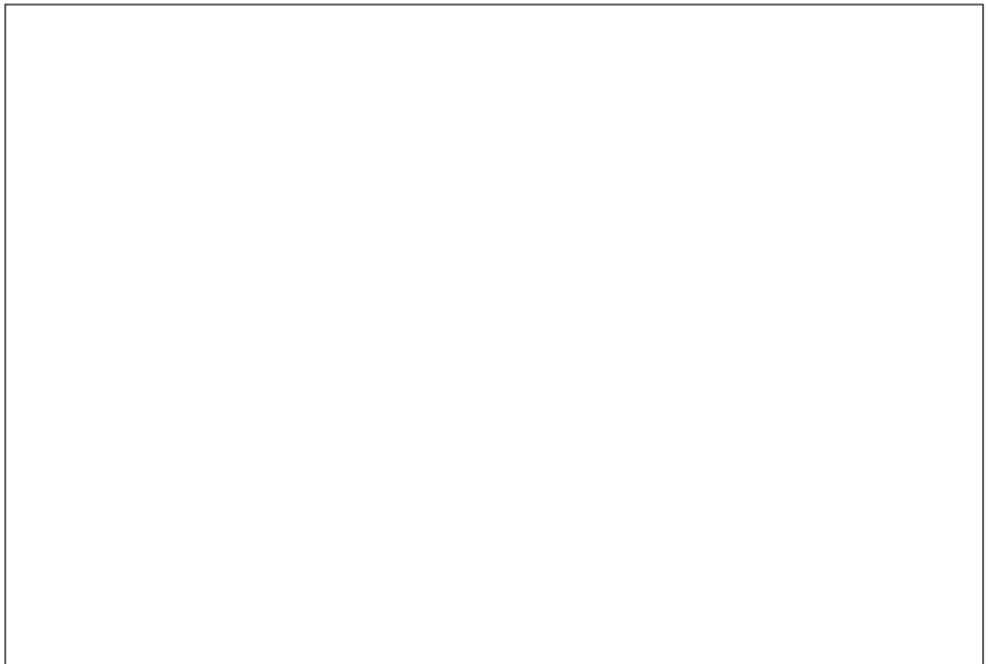
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$



redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

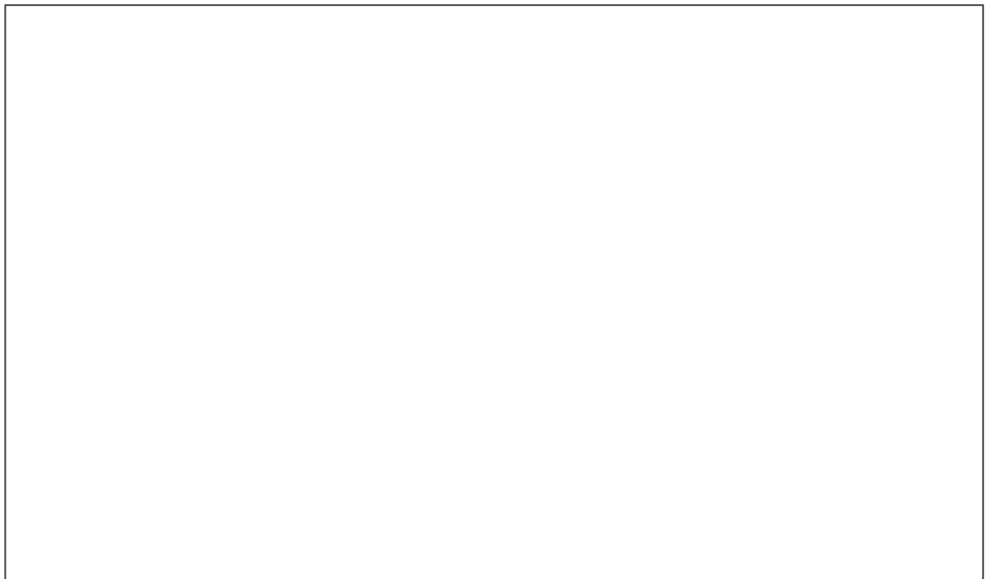
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$

$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$



redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$$

redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$$
$$((\lambda z.z)\,x)[x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$

redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$$
$$((\lambda z.z)\,x)[x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$

redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$$
$$((\lambda z.z)\,x)[x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$

redex searching
term rewriting

Storeless abstract machine (SAM)

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$$
$$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$$
$$((\lambda z.z)\,x)[x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$
$$(\lambda z'.z')[z \leftarrow \lambda z'.z'][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$$

redex searching
term rewriting

IAM vs. SAM

IAM (call-by-name)

GoI token passing
fixed graph

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$

SAM (call-by-need)

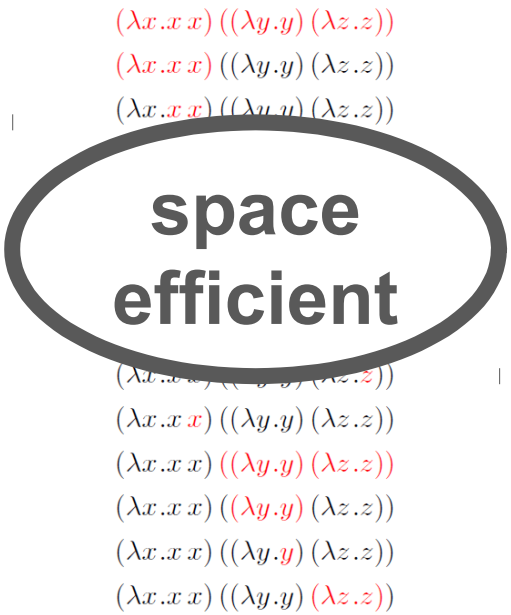
redex searching
term rewriting

$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z))$
$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$
$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$
$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$
$(x\,x)[x \leftarrow (\lambda y.y)\,(\lambda z.z)]$
$(x\,x)[x \leftarrow y[y \leftarrow \lambda z.z]]$
$((\lambda z.z)\,x)[x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$
$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$
$z[z \leftarrow x][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$
$(\lambda z'.z')[z \leftarrow \lambda z'.z'][x \leftarrow \lambda z.z][y \leftarrow \lambda z.z]$

IAM vs. SAM

IAM (call-by-name)

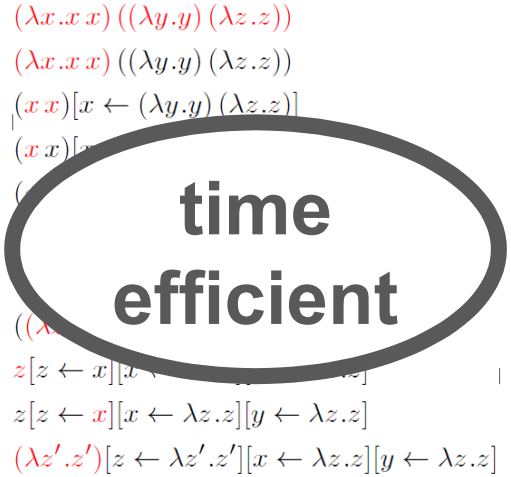
GoI token passing
fixed graph



space
efficient

SAM (call-by-need)

redex searching
term rewriting



time
efficient

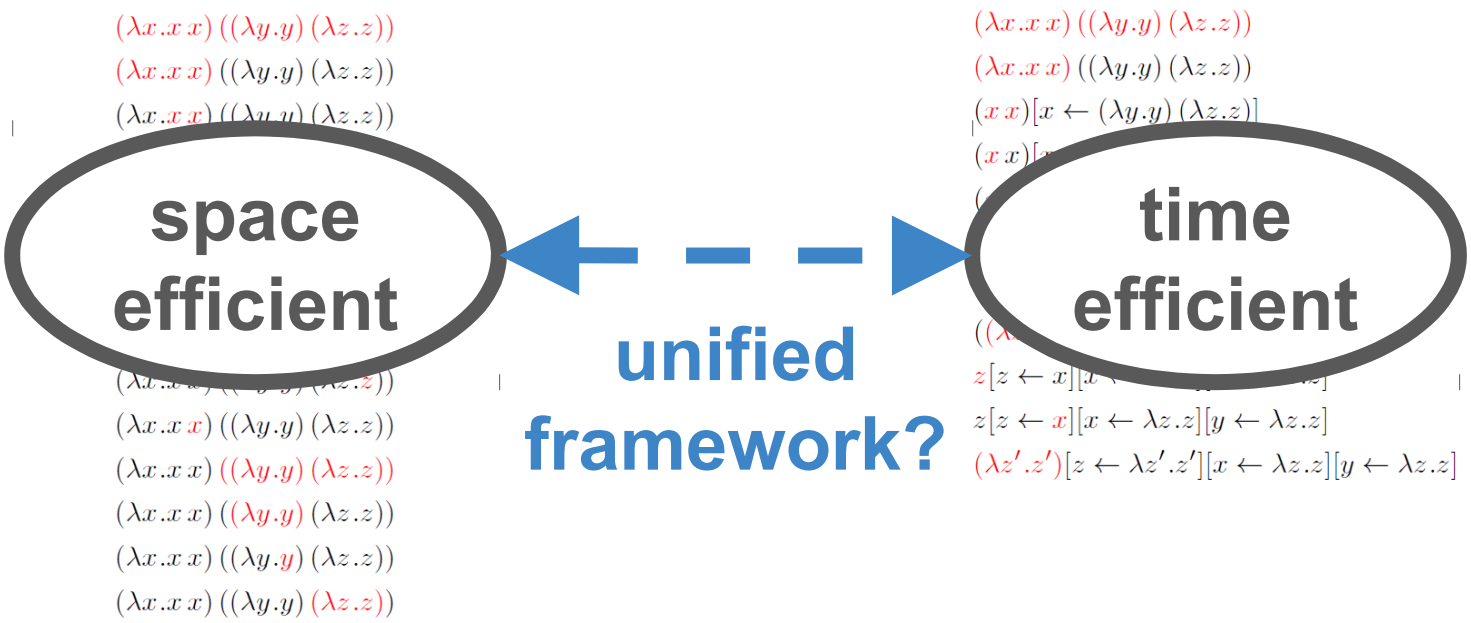
IAM vs. SAM

IAM (call-by-name)

GoI token passing
fixed graph

SAM (call-by-need)

redex searching
term rewriting



Quantitative analysis by GoI-style token passing

of

semantics of linear logic

space-time trade-off of program execution cost

abstract machines for lambda-calculus

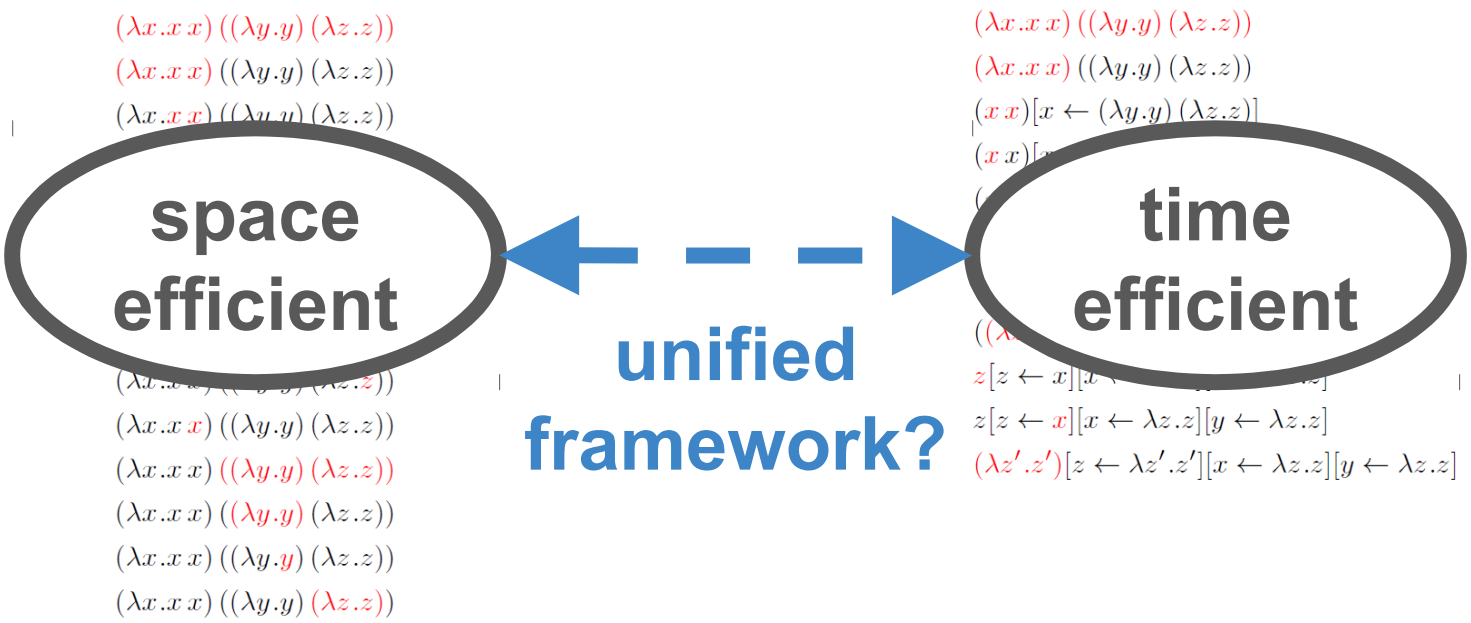
IAM vs. SAM

IAM (call-by-name)

GoI token passing
fixed graph

SAM (call-by-need)

redex searching
term rewriting



Dynamic GoI machine (DGoIM)

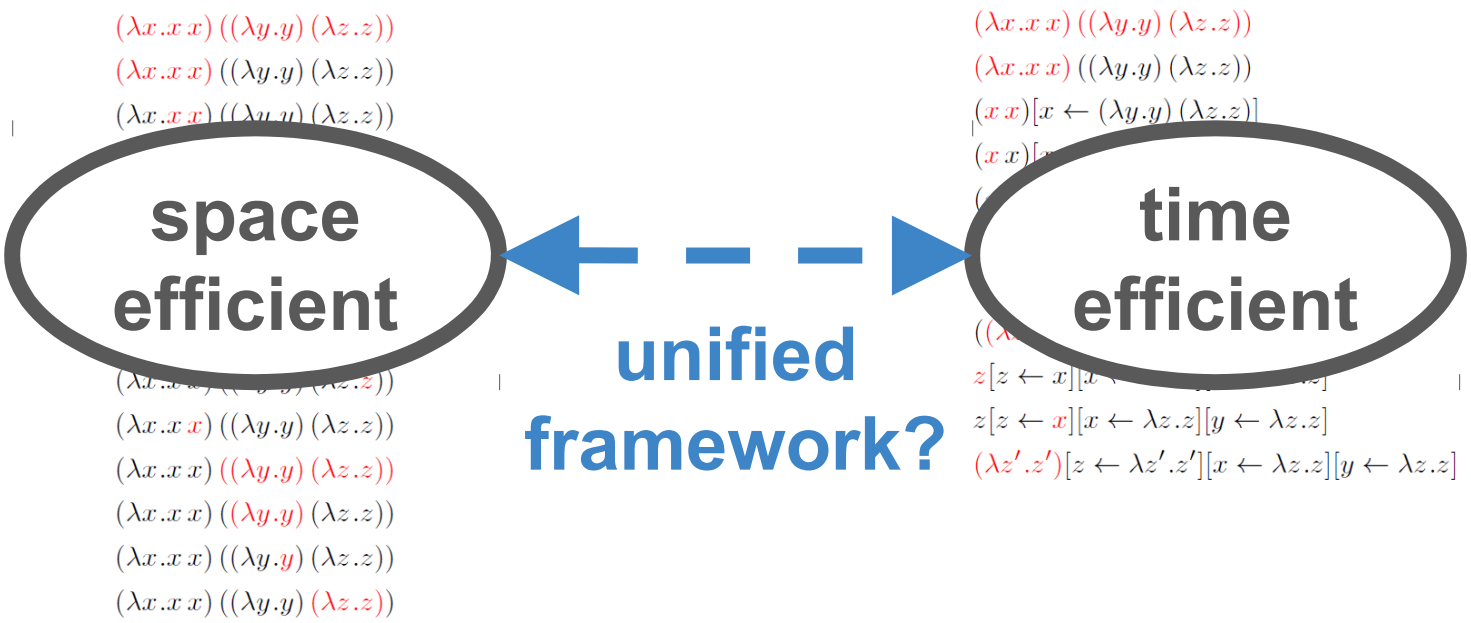
GoI token passing graph rewriting

IAM (call-by-name)

GoI token passing fixed graph

SAM (call-by-need)

redex searching term rewriting



Dynamic GoI machine (DGoIM)

GoI token passing graph rewriting

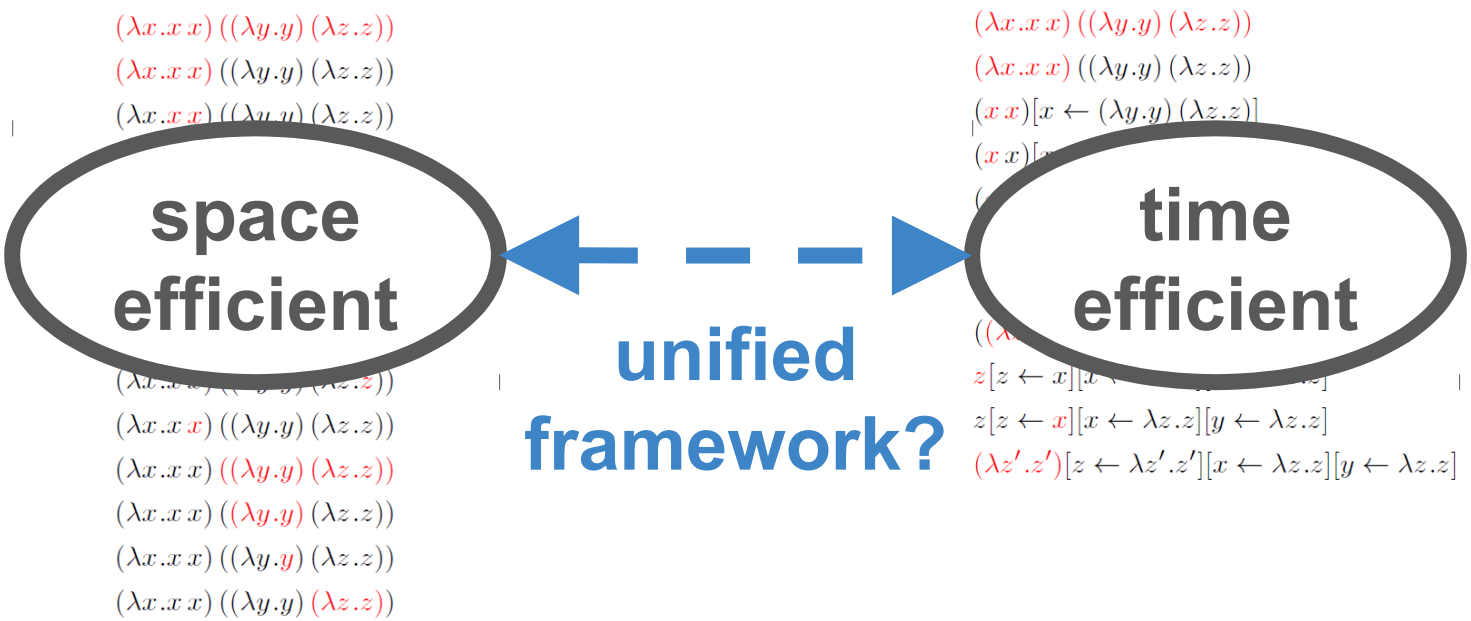
[Sinot '05 & '06]

IAM (call-by-name)

GoI token passing fixed graph

SAM (call-by-need)

redex searching term rewriting



Dynamic GoI machine (DGoIM)

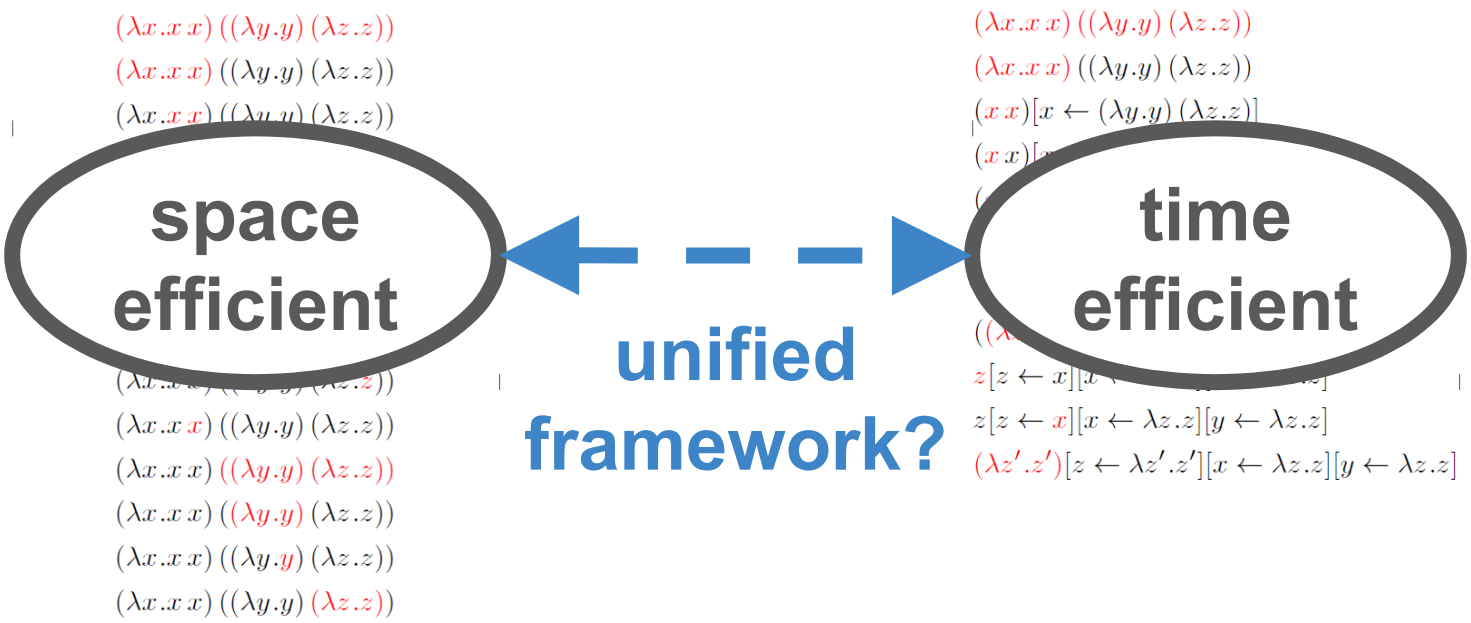
GoI token passing graph rewriting

IAM (call-by-name)

GoI token passing fixed graph

SAM (call-by-need)

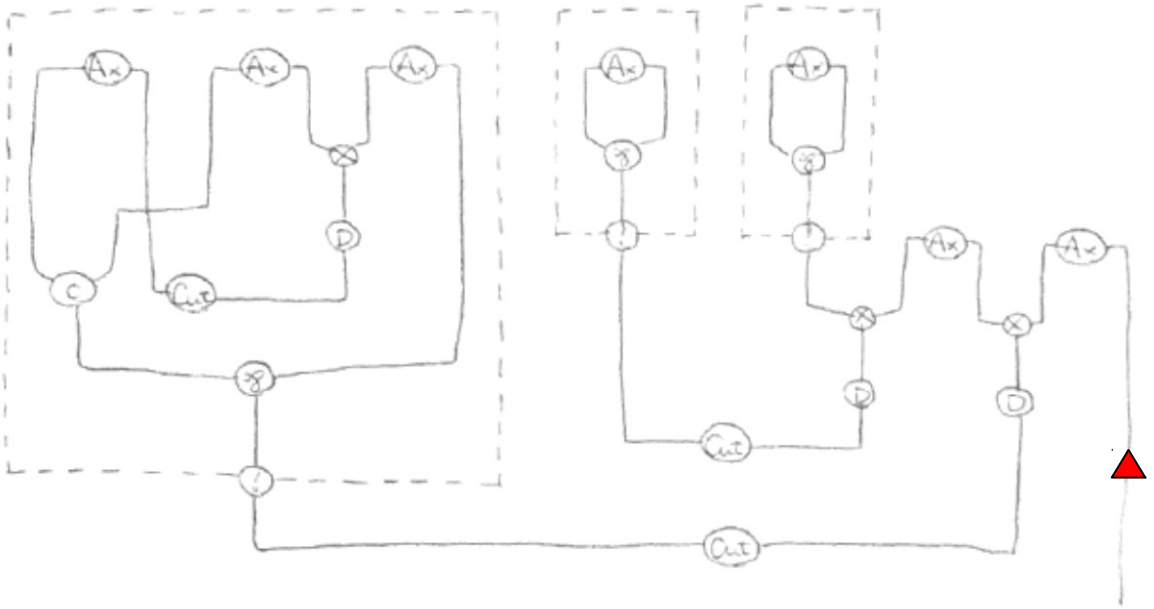
redex searching term rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

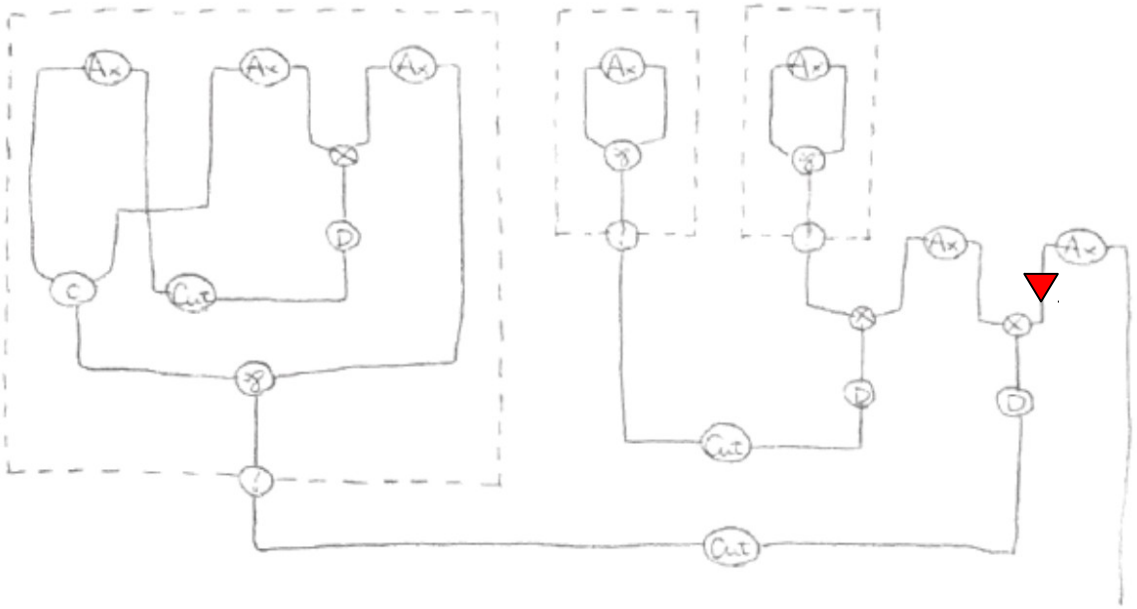
GoI token passing graph rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

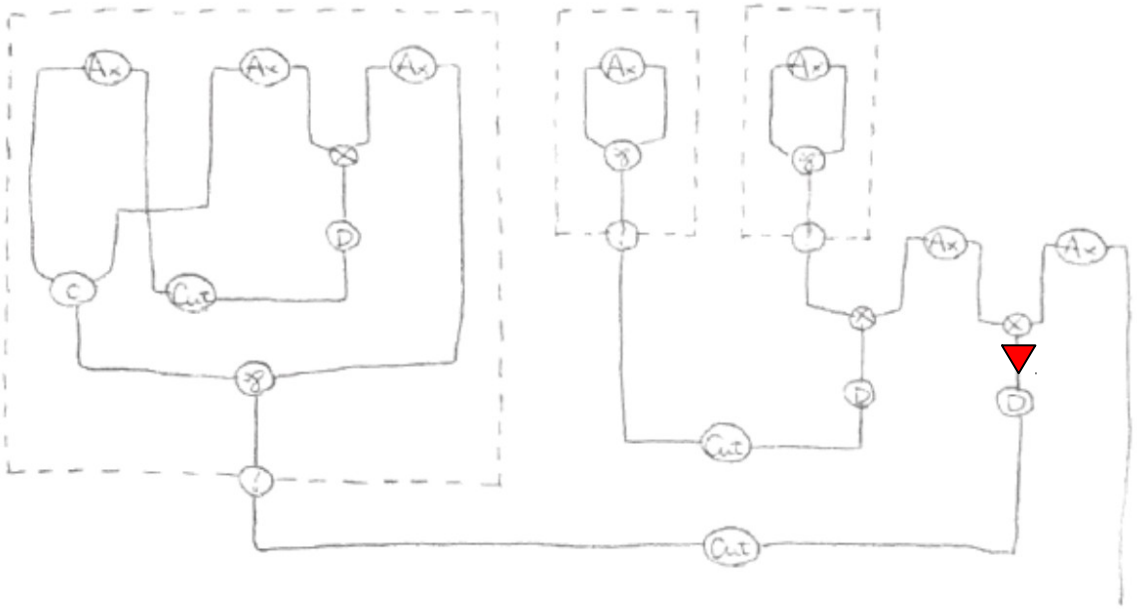
GoI token passing graph rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

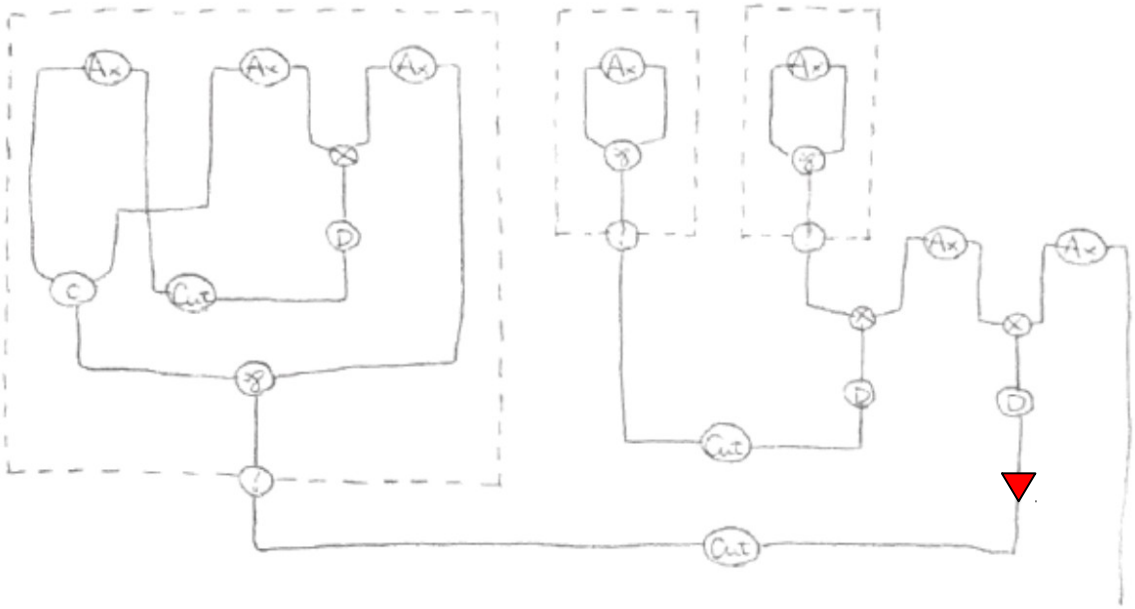
GoI token passing graph rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \Downarrow \lambda z.z$$

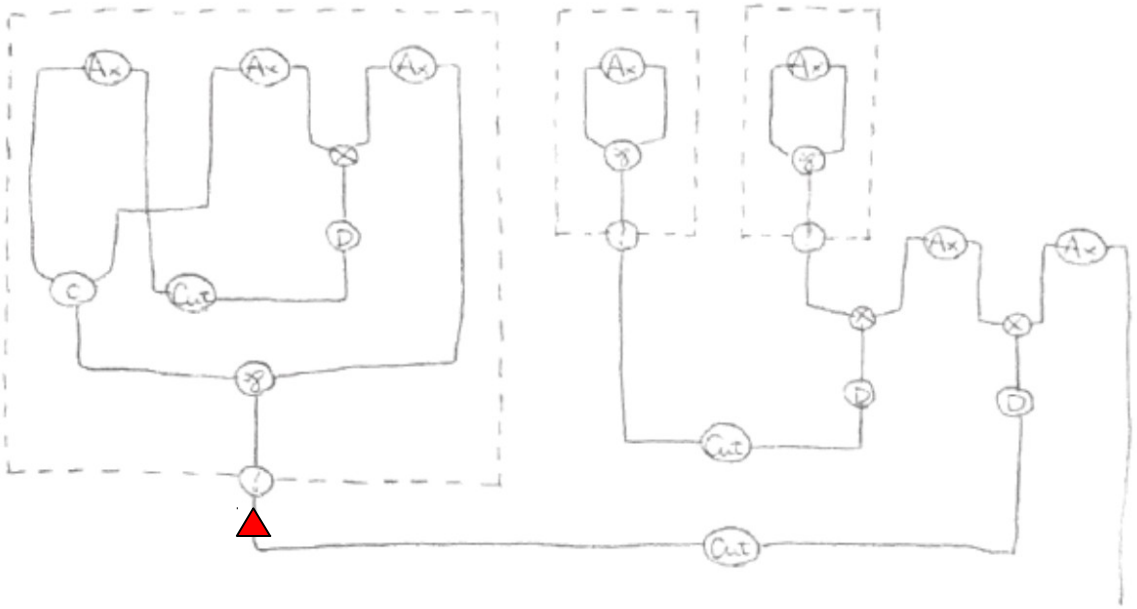
GoI token passing graph rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

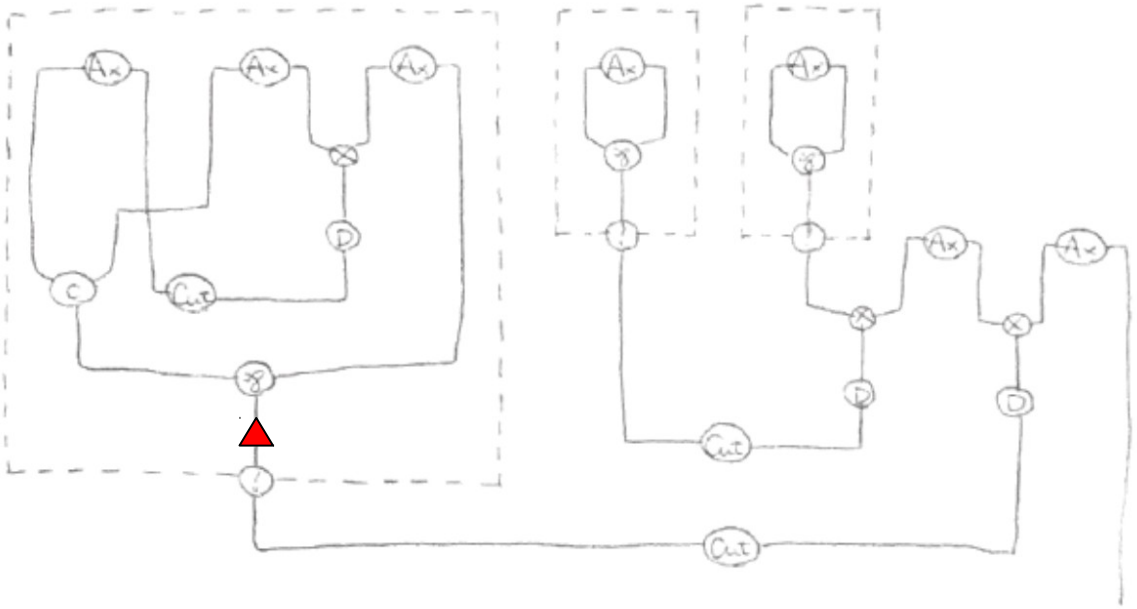
GoI token passing graph rewriting



DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

GoI token passing graph rewriting

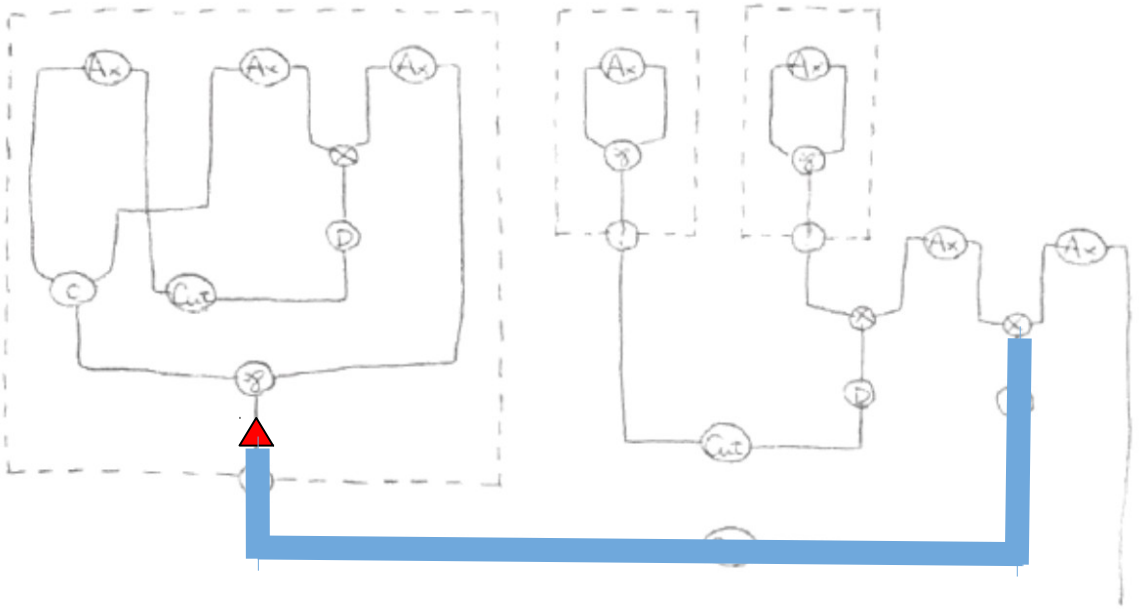


DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

GoI token passing graph rewriting

redex detected

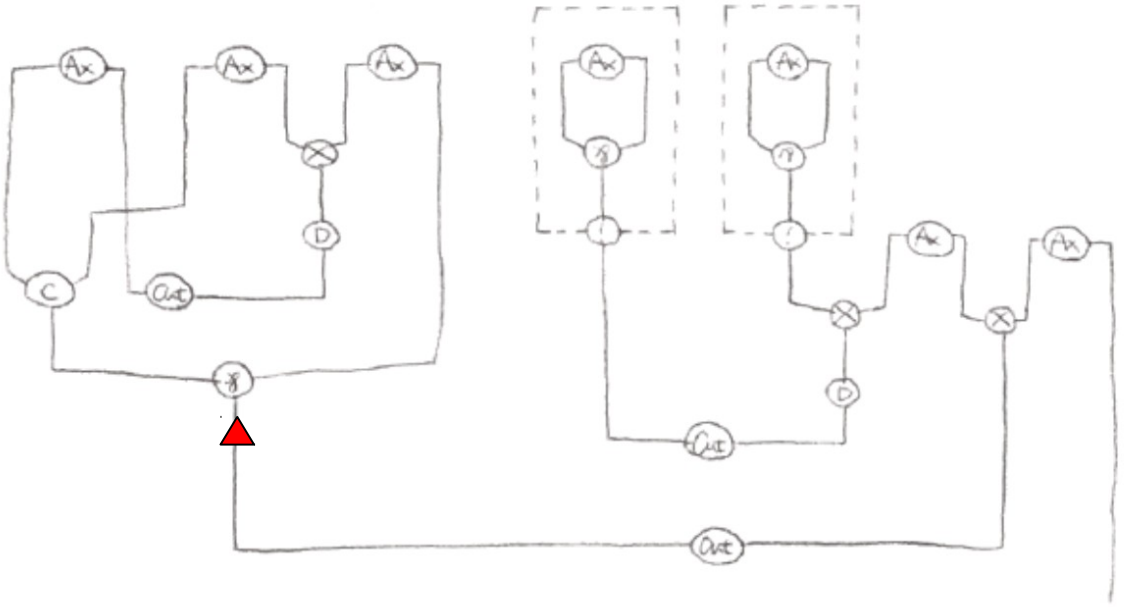


DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

GoI token passing graph rewriting

(1) trigger rewriting

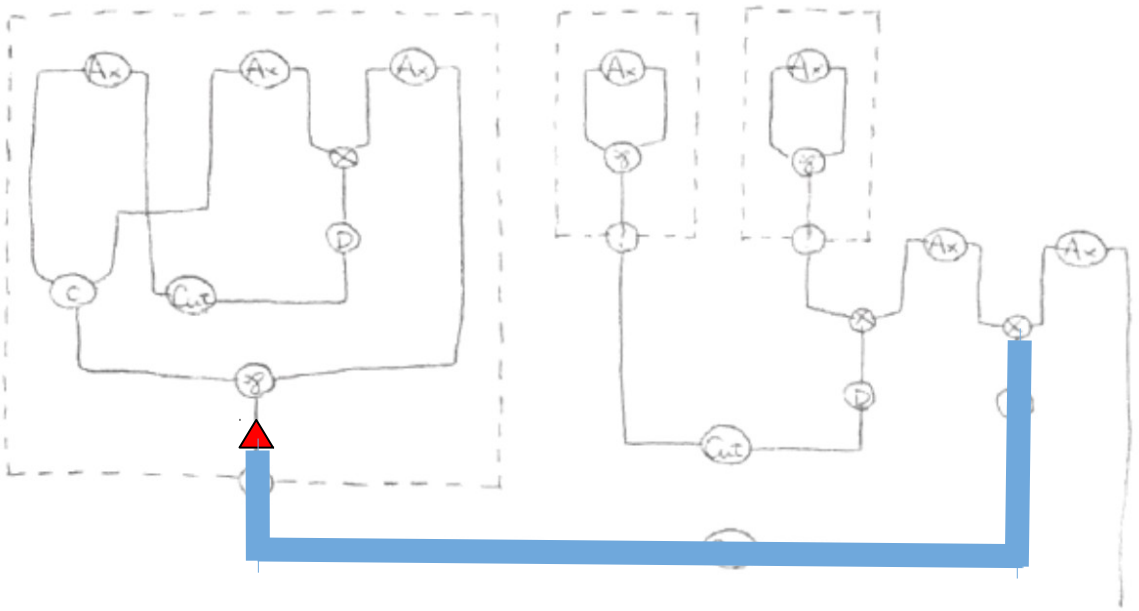


DGoIM = IAM + proof-net cut elimination

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

GoI token passing graph rewriting

redex detected

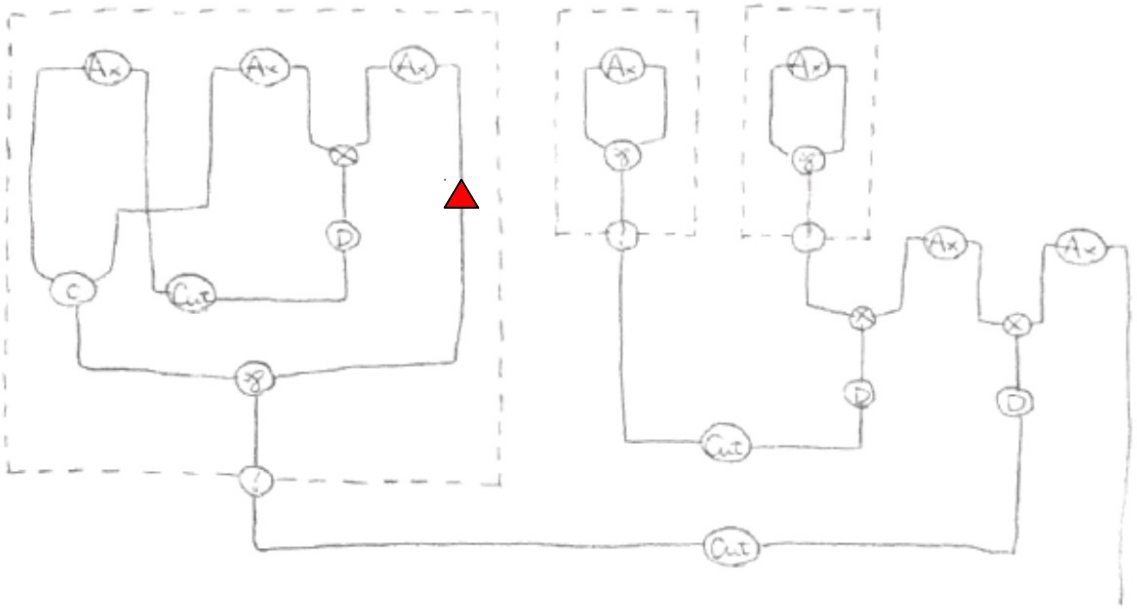


DGoIM = IAM + proof-net cut elimination

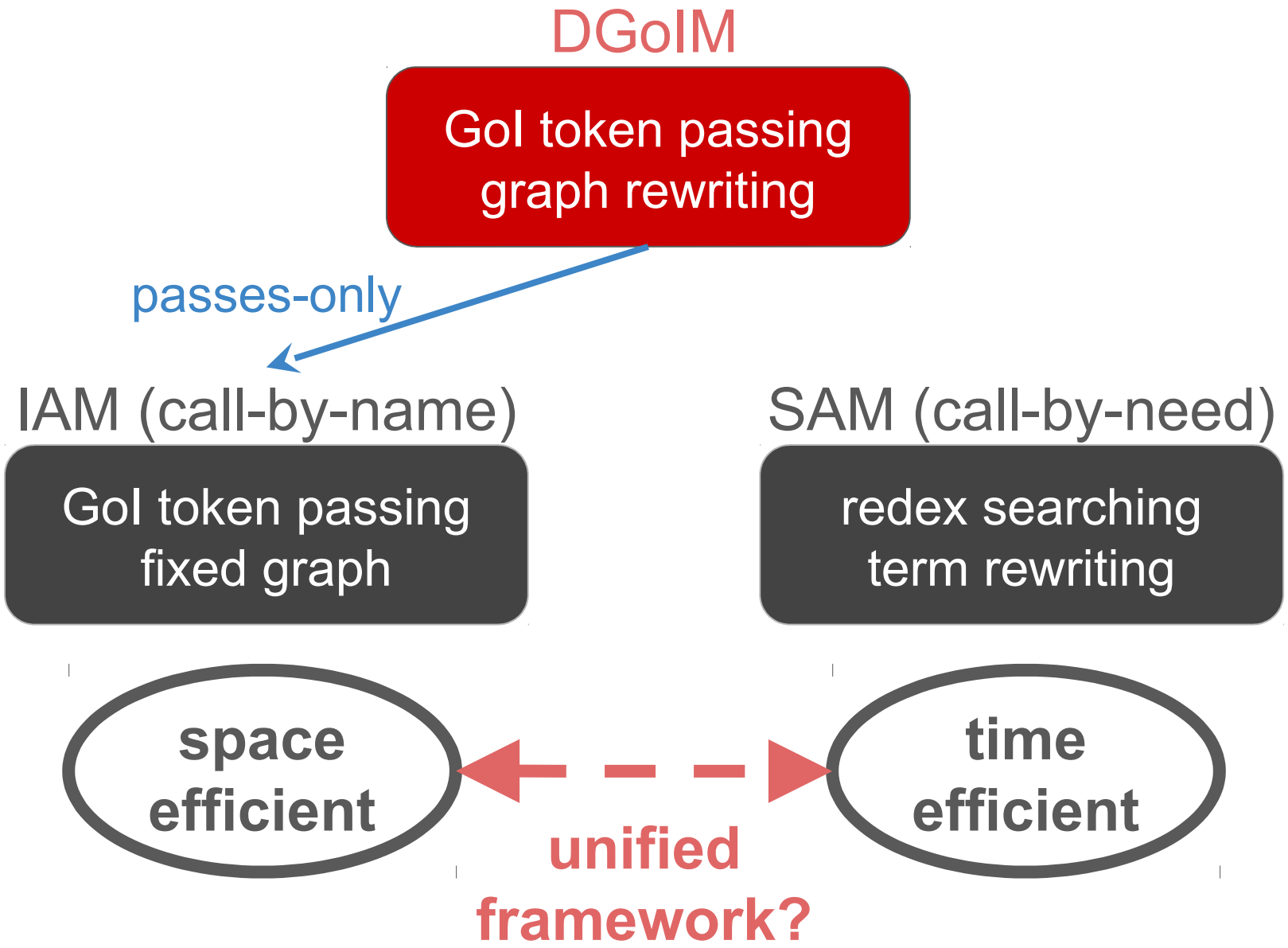
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

GoI token passing graph rewriting

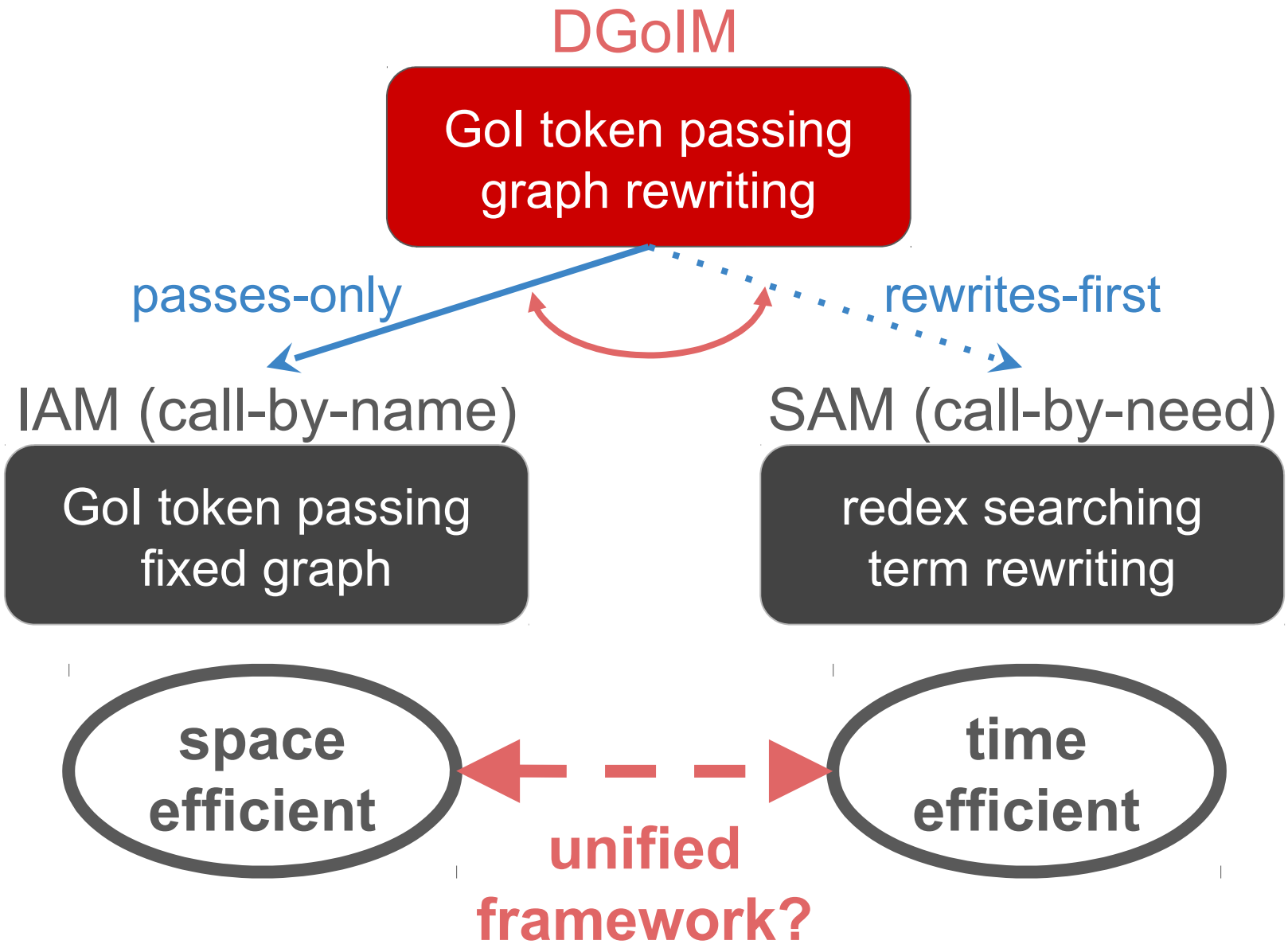
(2) keep passing



DGoIM: *flexible interleaving*



DGoIM: *flexible interleaving*

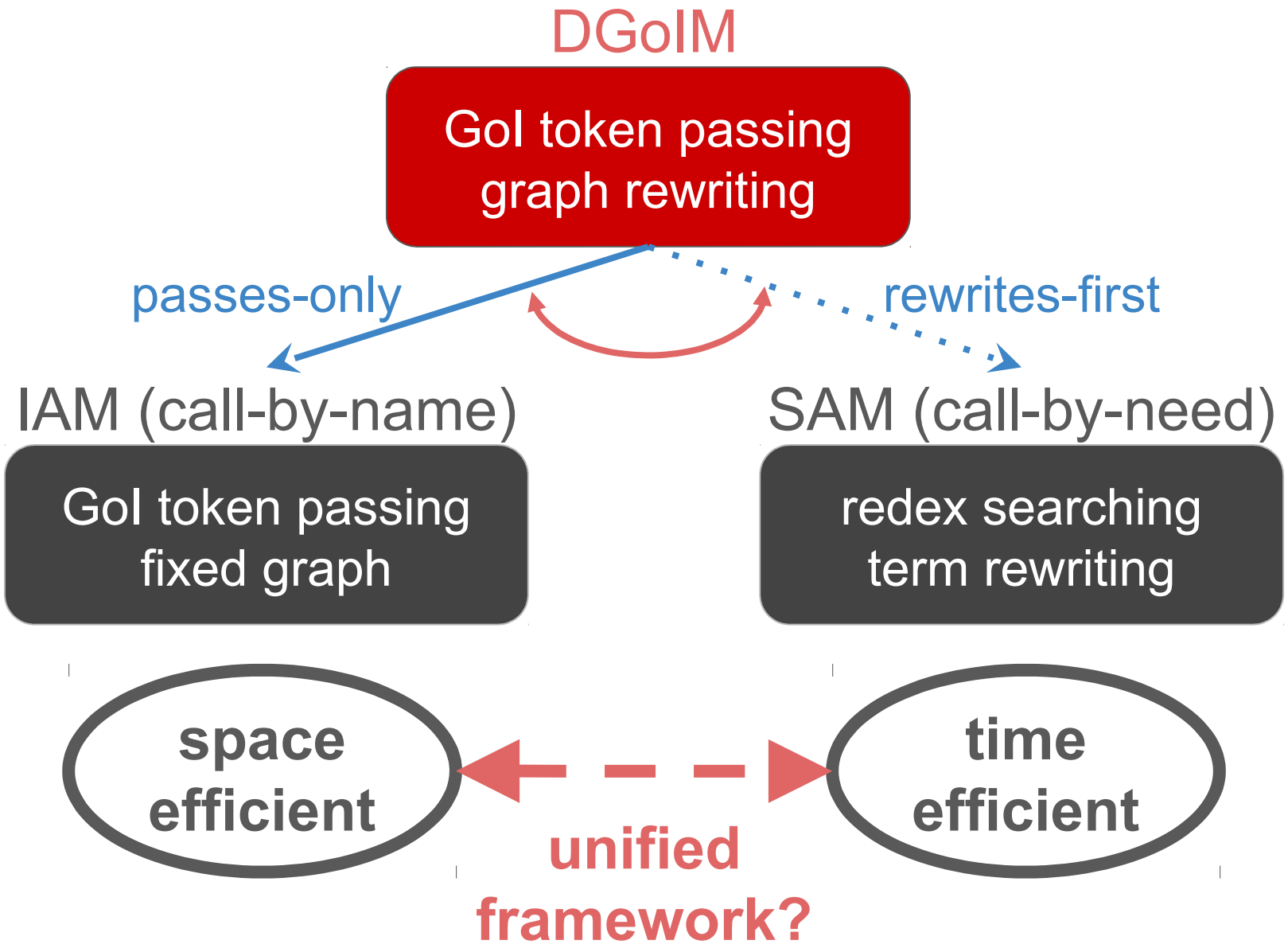


Quantitative analysis by GoI-style token passing of space-time trade-off of program execution cost

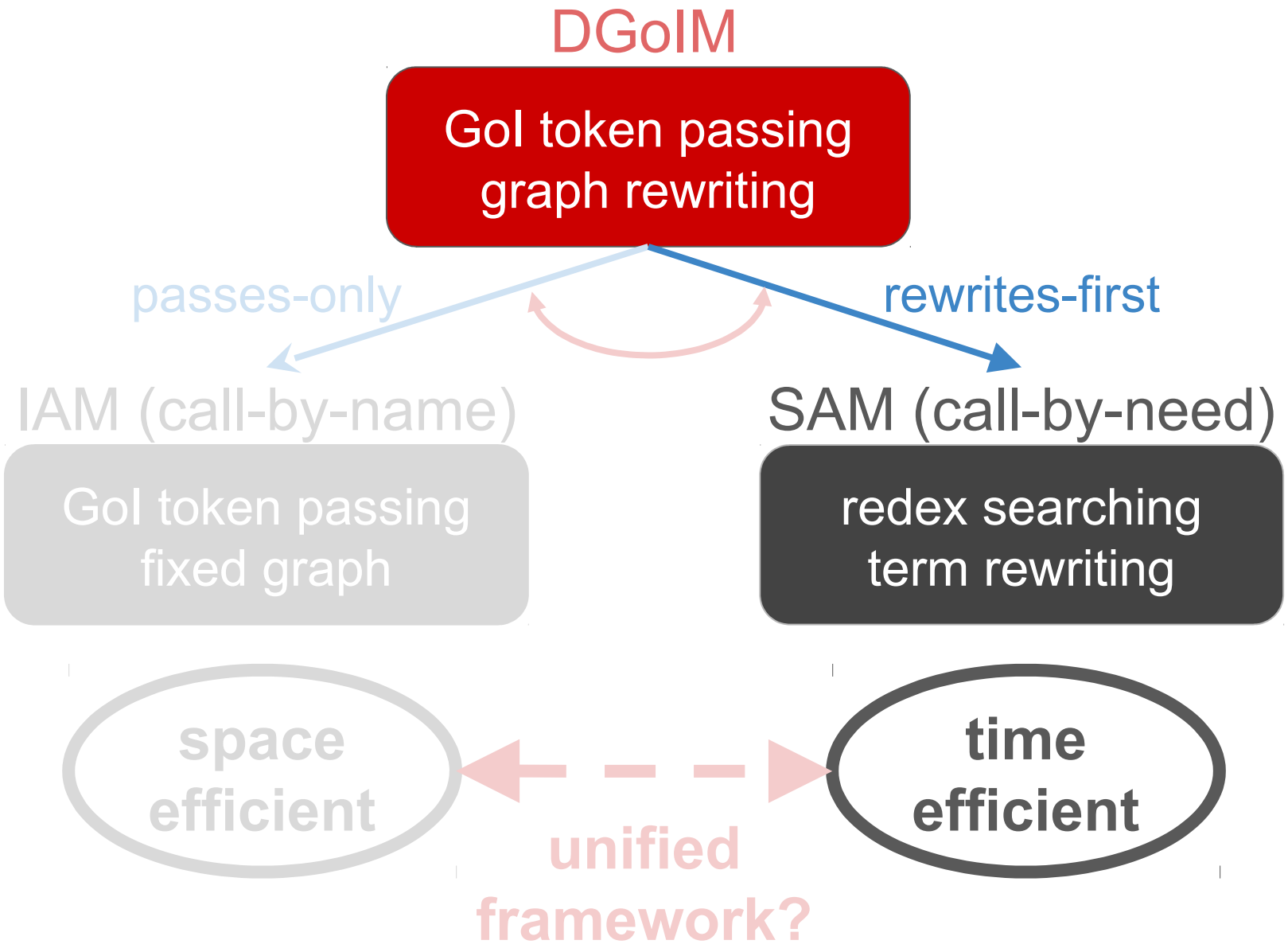
semantics of linear logic

abstract machines for lambda-calculus

DGoIM: *flexible interleaving*



DGoIM: *rewrites-first interleaving*



Rewrites-first DGoIM: states

▶ **Definition 2.** Let $\mathcal{L}$ be a fixed countable (infinite) set of *names*. The state of the transition system $s = (\mathbb{G}, p, h, m)$ consists of the following elements:

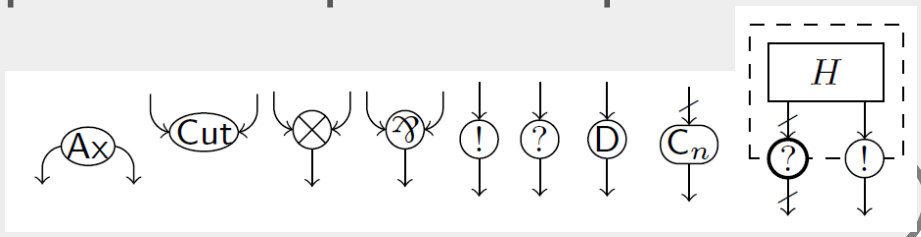
- a *named well-boxed graph* $\mathbb{G} = (G, \ell_G)$, that is a well-boxed graph G with a *naming* l_G that assigns a unique name $\alpha \in \mathcal{L}$ to each node of G
- a pair $p = (e, d)$ called *position*, of an edge e of G and a *direction* $d \in \{\uparrow, \downarrow\}$
- a *history stack* h defined by the grammar below, $\alpha \in \mathcal{L}, n \in \mathbb{N}$:

$$h ::= \square \mid \mathsf{Ax}_\alpha : h \mid \mathsf{Cut}_\alpha : h \mid \otimes_\alpha : h \mid \parr_\alpha : h \mid !_\alpha : h \mid \mathsf{D}_\alpha : h \mid \mathsf{C}^n_\alpha : h.$$

- a *multiplicative stack* m defined by the BNF grammar $m ::= \square \mid \mathsf{l} : m \mid \mathsf{r} : m$.

Rewrites-first DGoIM: states

multiplicative-exponential proof net



▶ **Definition 2.** Let $\mathcal{L}$ be a fixed countable (infinite) set of *na* e state of the transition system $s = (\mathbb{G}, p, h, m)$ consists of the following elements:

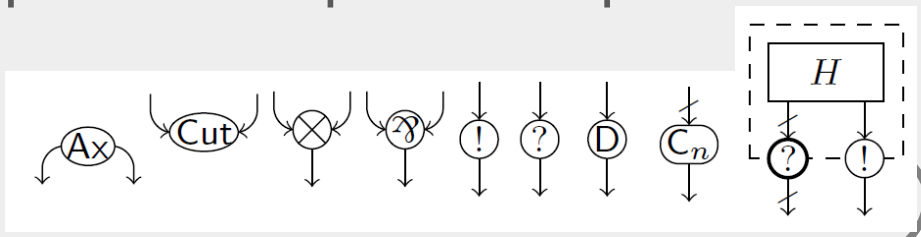
- a *named well-boxed graph* $\mathbb{G} = (G, \ell_G)$, that is a well-boxed graph G with a *naming* l_G that assigns a unique name $\alpha \in \mathcal{L}$ to each node of G
- a pair $p = (e, d)$ called *position*, of an edge e of G and a *direction* $d \in \{\uparrow, \downarrow\}$
- a *history stack* h defined by the grammar below, $\alpha \in \mathcal{L}, n \in \mathbb{N}$:

$$h ::= \square \mid \mathsf{Ax}_\alpha : h \mid \mathsf{Cut}_\alpha : h \mid \otimes_\alpha : h \mid \parr_\alpha : h \mid !_\alpha : h \mid \mathsf{D}_\alpha : h \mid \mathsf{C}^n_\alpha : h.$$

- a *multiplicative stack* m defined by the BNF grammar $m ::= \square \mid \mathsf{l} : m \mid \mathsf{r} : m$.

Rewrites-first DGoIM: states

multiplicative-exponential proof net



▶ **Definition 2.** Let $\mathcal{L}$ be a fixed countable (infinite) set of *na* ... *e* state of the transition system $s = (\mathbb{G}, p, h, m)$ consists of the following elements:

- a *named well-boxed graph* $\mathbb{G} = (G, \ell_G)$, that is a well-boxed graph G with a *naming* l_G that assigns a unique name $\alpha \in \mathcal{L}$ to each node of G
- a pair $p = (e, d)$ called *position*, of an edge e of G and a *direction* ...
- a *history stack* h defined by the grammar below, $\alpha \in \mathcal{L}$ ...

$$h ::= \square \mid \mathsf{Ax}_\alpha : h \mid \mathsf{Cut}_\alpha : h \mid \otimes_\alpha : h \mid \parr_\alpha : h \mid !_\alpha : h \mid$$

- a *multiplicative stack* m defined by the BNF grammar ...

simplified data of IAM

Rewrites-first DGoIM: transitions

pass transitions

selected GoI token passing

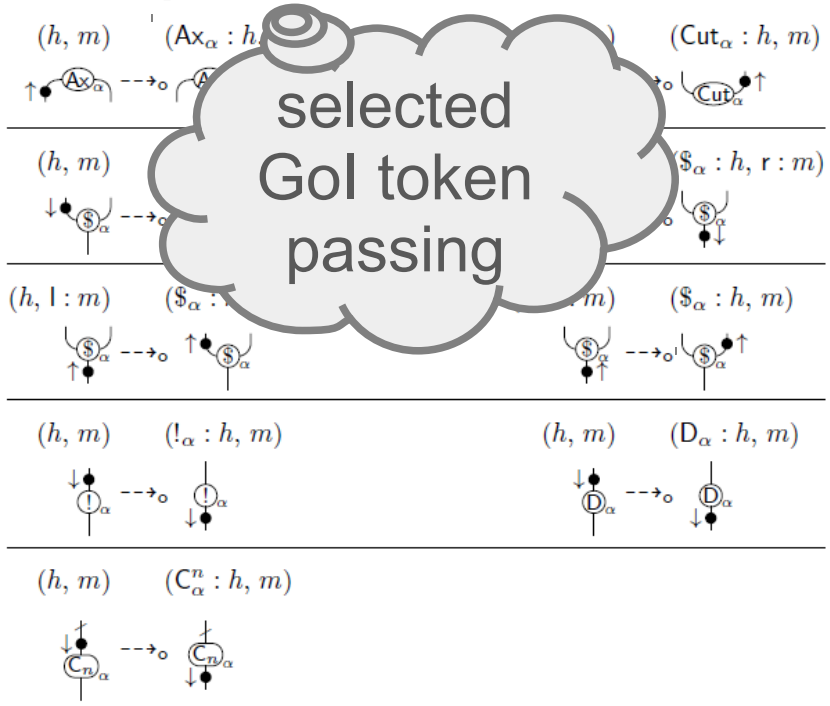


Figure 3 Pass Transitions ($\$ \in \{\otimes, \parr\}$, $n > 0$)

rewrite transitions

proof-net cut elimination

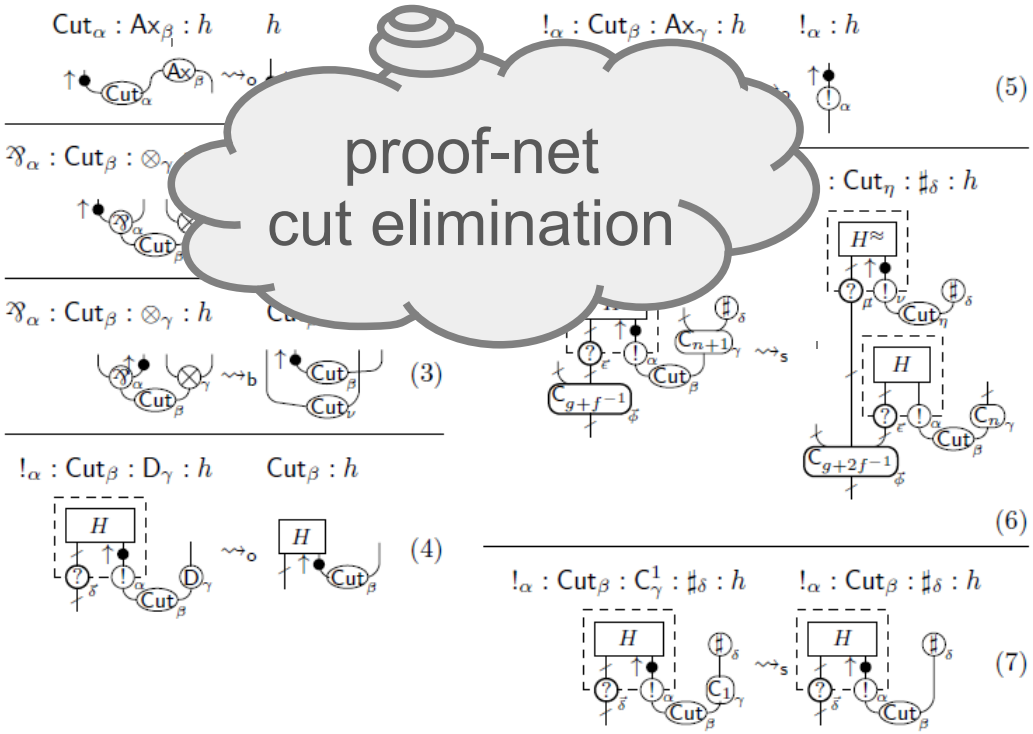


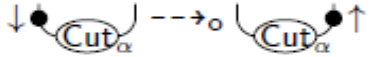
Figure 4 Rewrite Transitions ($n > 0$)

Rewrites-first DGoIM: transitions

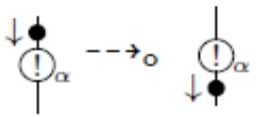
pass transitions

selected GoI token passing

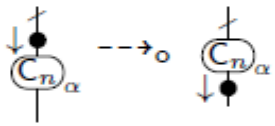
$(h, m) \qquad (\mathsf{Cut}_\alpha : h, m)$



$(h, m) \qquad (!_\alpha : h, m)$



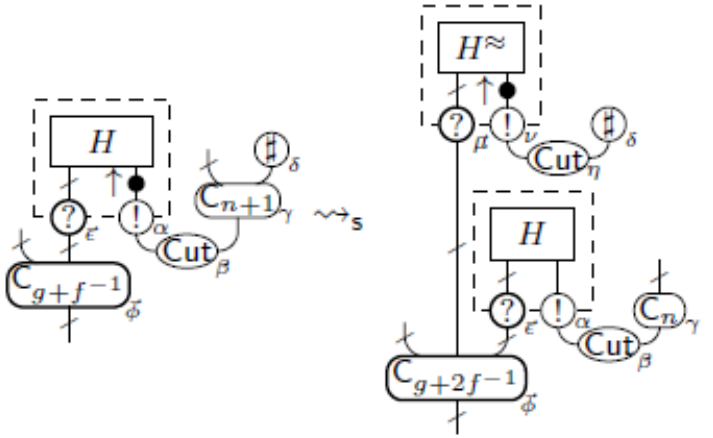
$(h, m) \qquad (\mathsf{C}^n_\alpha : h, m)$



rewrite transitions

proof-net cut elimination

$!_\alpha : \mathsf{Cut}_\beta : \mathsf{C}^{n+1}_\gamma : \sharp_\delta : h \qquad !_\nu : \mathsf{Cut}_\eta : \sharp_\delta : h$



(6)

Rewrites-first DGoIM: transitions

pass transitions

selected GoI token passing

rewrite transitions

proof-net cut elimination

$$s \to_{\mathsf{x}} s' \overset{\text{def.}}{\Longleftrightarrow} \begin{cases} s \rightsquigarrow_{\mathsf{x}} s' & \text{(if } \rightsquigarrow_{\mathsf{x}} \text{ possible)} \\ s \dashrightarrow_{\mathsf{x}} s' & \text{(if only } \dashrightarrow_{\mathsf{x}} \text{ possible).} \end{cases}$$

rewrites-first interleaving

Translate lambda-terms

"call-by-value translation" of intuitionistic logic to linear logic
[Girard '87]

only values
can be copied

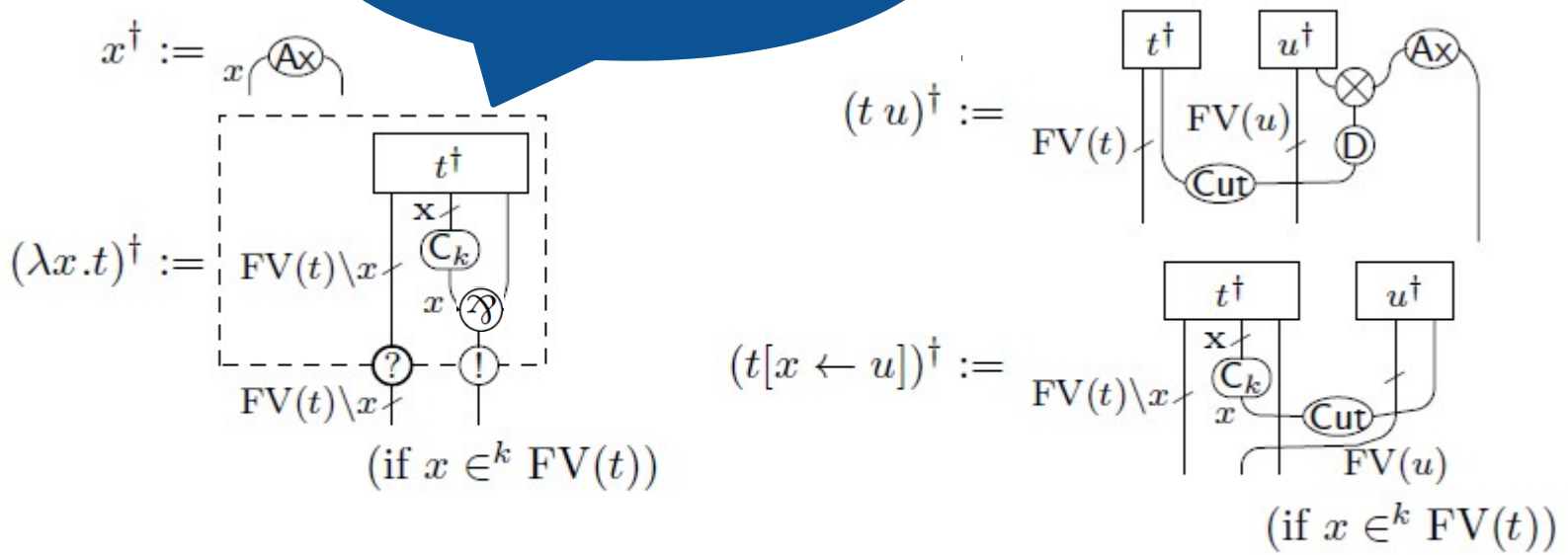


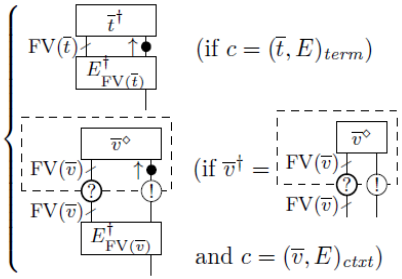
Figure 6 Inductive Translation $(\cdot)^\dagger$ of Terms to Well-boxed Graphs

Rewrites-first DGoIM: implement call-by-need

DGoIM

SAM (call-by-need)

▶ **Definition 8** (binary relation $\preceq$). A reachable configuration c and a state $((G, \ell_G), p, h, m)$ satisfies $c \preceq ((G, \ell_G), p, h, m)$ if and only if ℓ_G is an arbitrary naming, $((G, \ell_G), p, h, m)$ is rooted at the unique open edge of G, and $(G, p) =$



▶ **Theorem 9** (weak simulation). *Let a configuration c and a state s satisfy $c \preceq s$.*

1. *If a transition $c \to_{\mathsf{b}} c'$ of the SAM is possible, there exists a sequence $s \to_{\mathsf{o}}^{2} \to_{\mathsf{b}} \to_{\mathsf{o}} s'$ such that $c' \preceq s'$.*
2. *If a transition $c \to_{\mathsf{s}} c'$ of the SAM is possible, there exists a sequence $s \to_{\mathsf{s}} \to_{\mathsf{o}} s'$ such that $c' \preceq s'$.*
3. *If a transition $c \to_{\mathsf{o}} c'$ of the SAM is possible, there exists a sequence $s \to_{\mathsf{o}}^{N} s'$ such that $0 < N \leq 4$ and $c' \preceq s'$.*
4. *No transition $\to$ is possible at the state s' if $c' = (\overline{v}, A)_{ctxt}$.*

correctness via weak simulation

Rewrites-first DGoIM: implement call-by-need

time cost analysis à la [Accattoli '16]

▶ **Theorem 13** (time cost). *Let C, D be fixed natural numbers, and $r : s_0 \to_{\triangleright}^* s$ be a sequence of transitions of the $DGoIM_{\to_{\triangleright}}$. If there exists an execution $(\overline{t_0}, \langle\cdot\rangle)_{term} \to^* (\overline{t}, E)$ of the SAM such that $s_0 \preceq (\overline{t_0}, \langle\cdot\rangle)_{term}$ and $s \preceq (\overline{t}, E)$, the total time cost $T(r)$ of the sequence r satisfies:*

$$T(r) = \mathcal{O}((|\overline{t_0}| + C) \cdot (|r|_{\mathsf{b}} + D)).$$

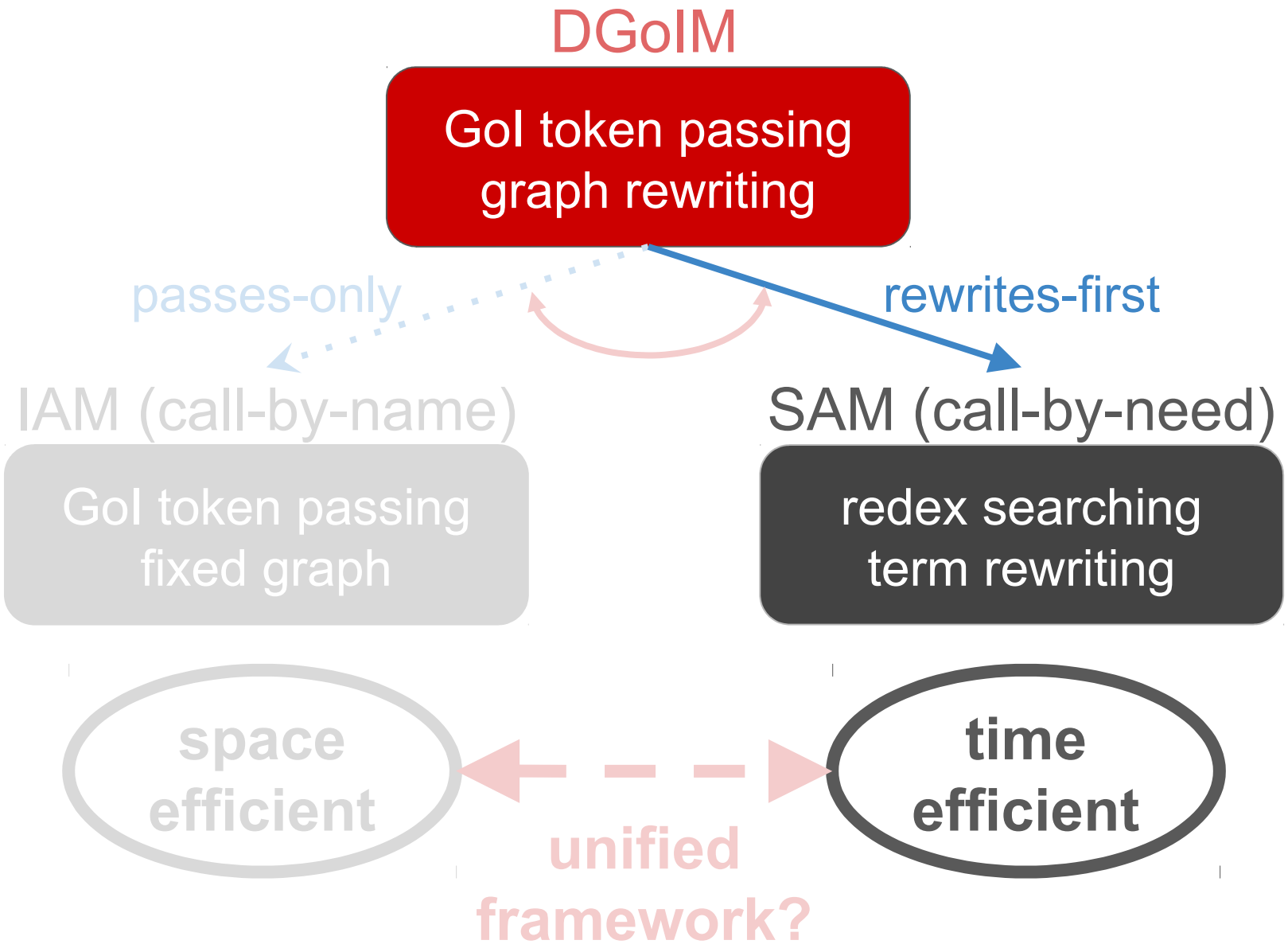
▶ **Corollary 14.** *The $DGoIM_{\to_{\triangleright}}$ is an efficient abstract machine, in the sense of Def. 11.*

time efficient

▶ **Definition 11** (classes of abstract machines [1, Def. 7.1]). **1.** An abstract machine is *efficient* if its execution time cost is linear in both the input size and the number of β-transitions.

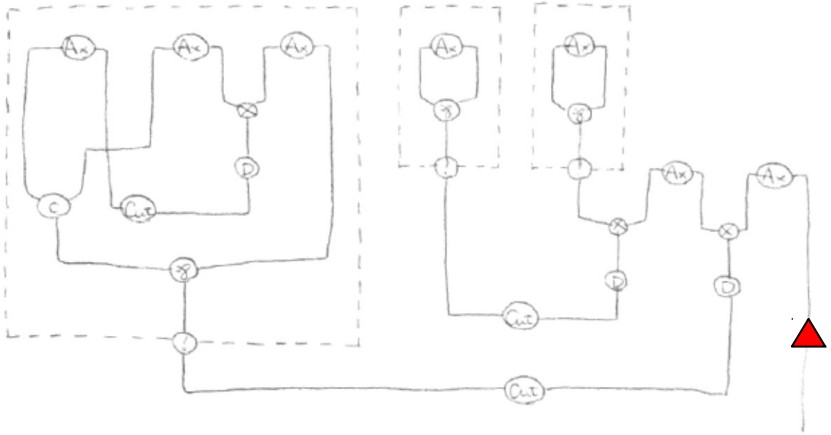
2. An abstract machine is *reasonable* if its execution time cost is polynomial in the input size and the number of β-transitions.
3. An abstract machine is *unreasonable* if it is not reasonable.

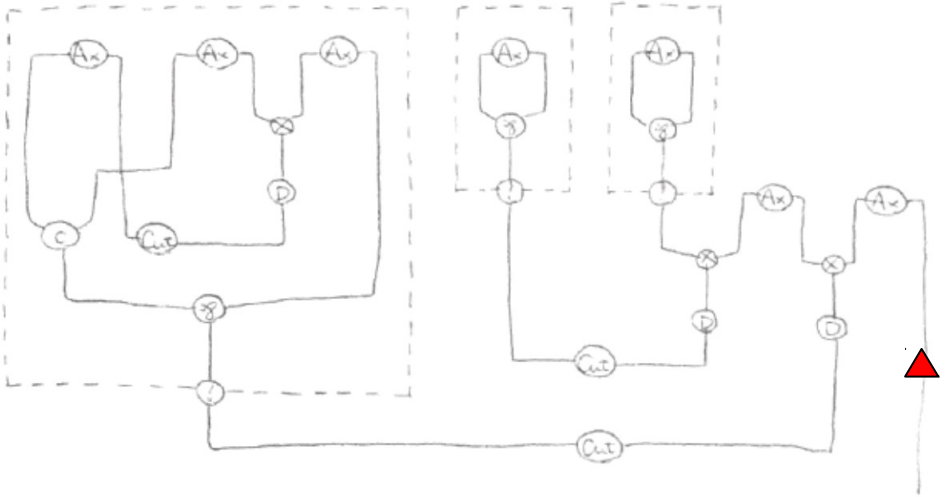
DGoIM: *rewrites-first interleaving*



IAM vs. rewrites-first DGoIM

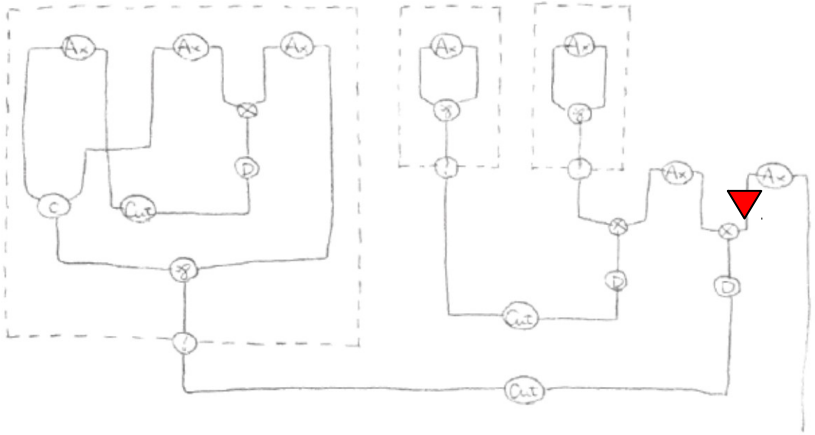
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

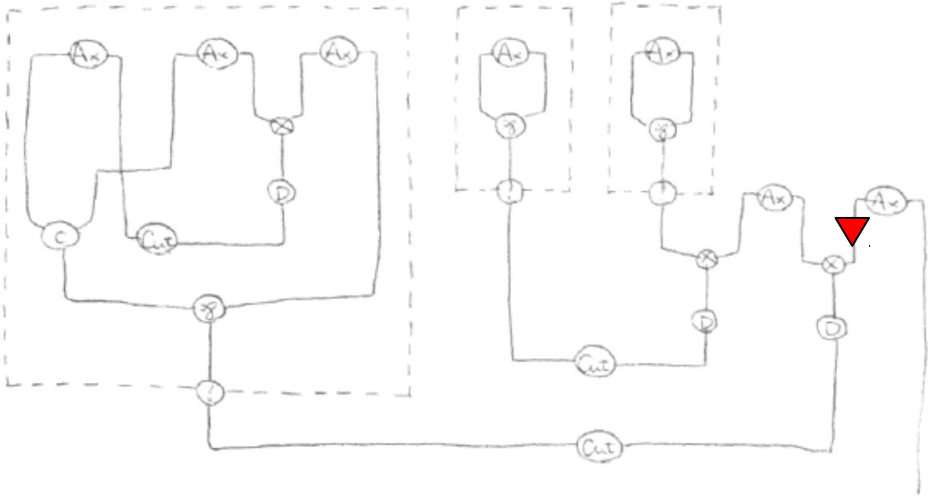




IAM vs. rewrites-first DGoIM

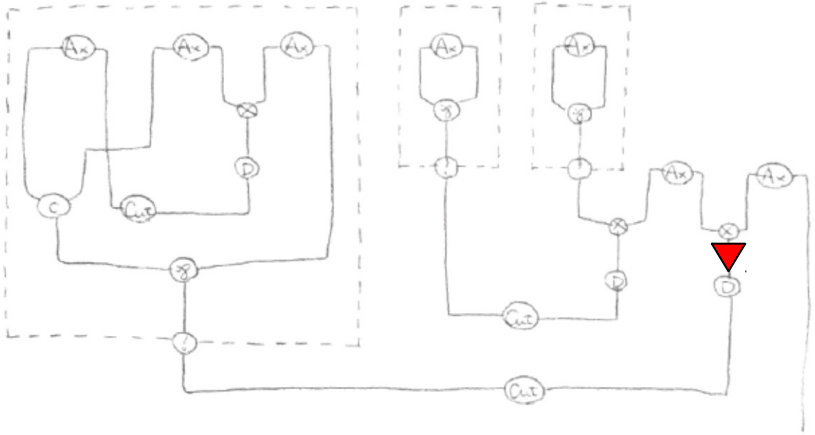
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

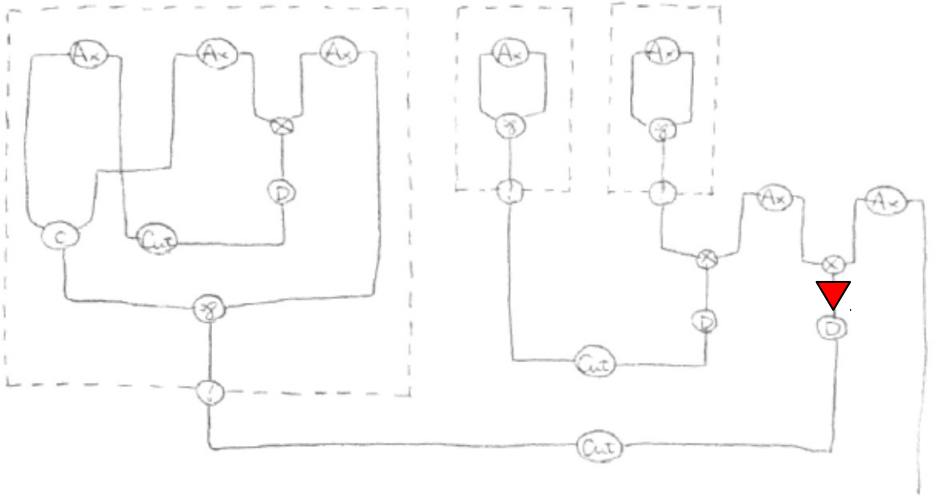




IAM vs. rewrites-first DGoIM

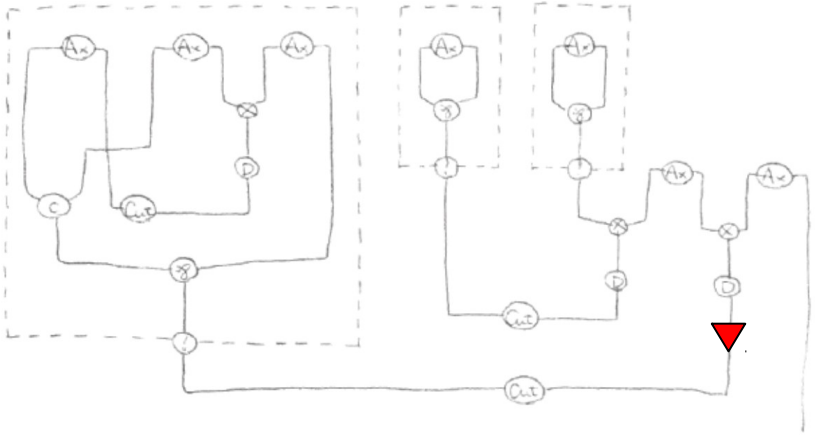
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

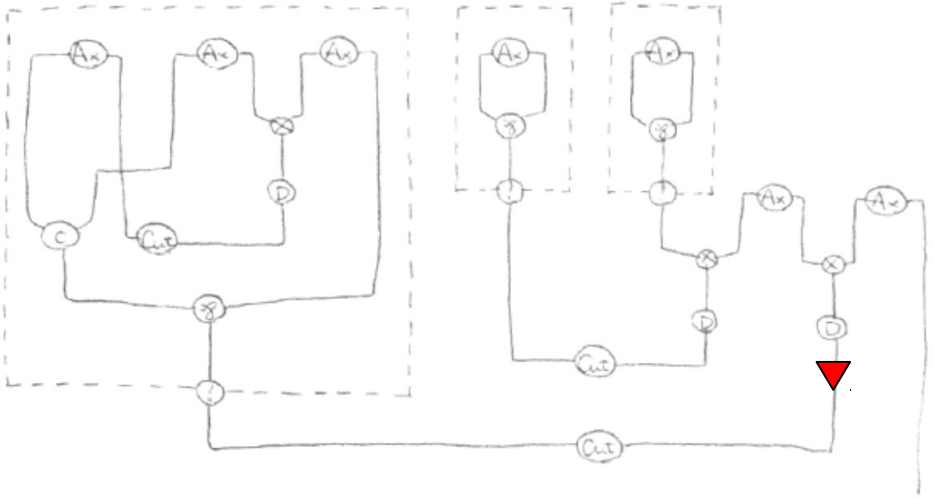




IAM vs. rewrites-first DGoIM

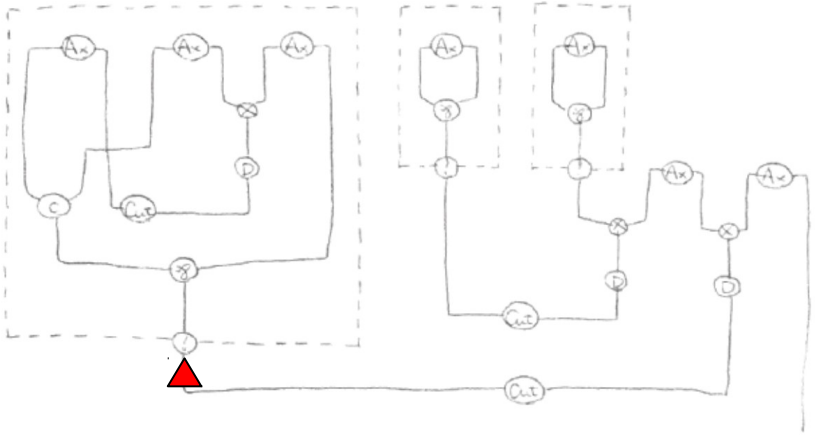
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

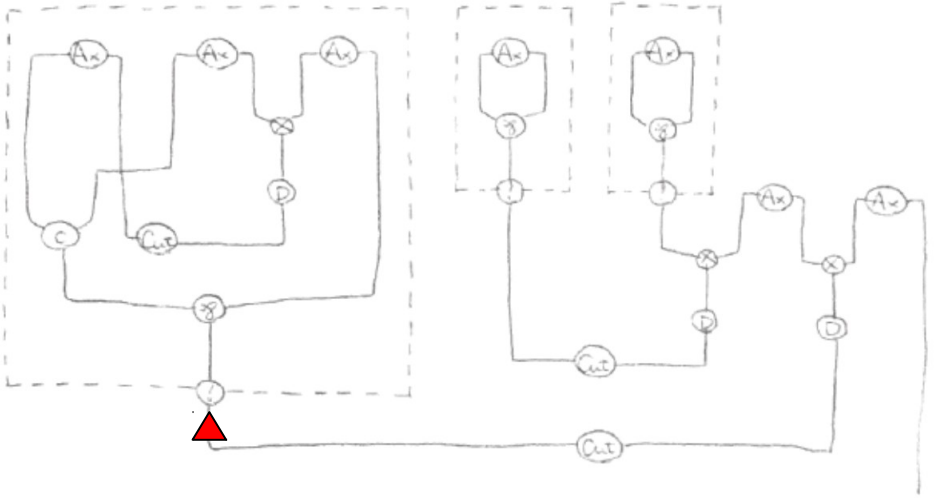




IAM vs. rewrites-first DGoIM

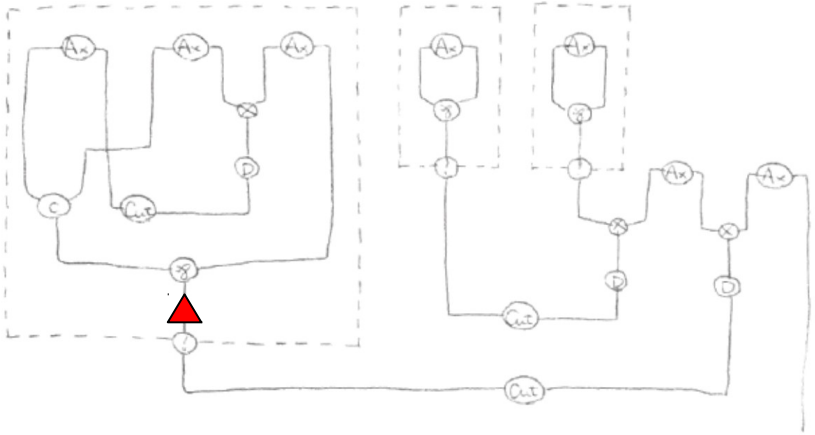
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

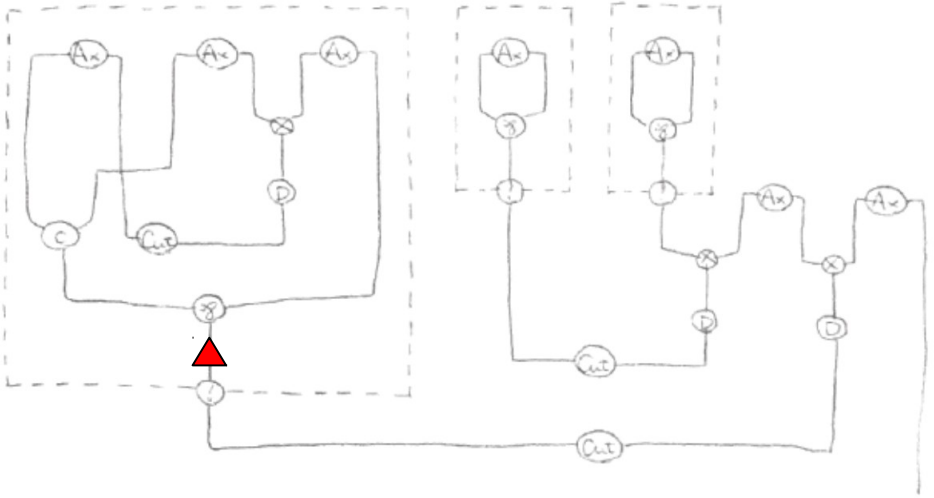




IAM vs. rewrites-first DGoIM

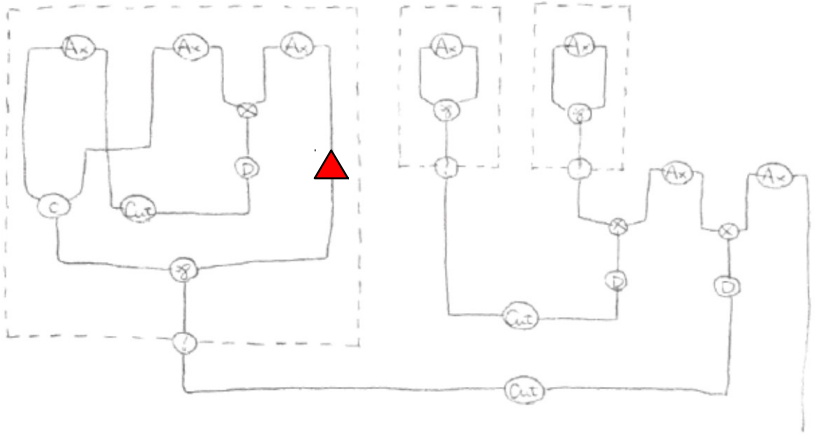
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

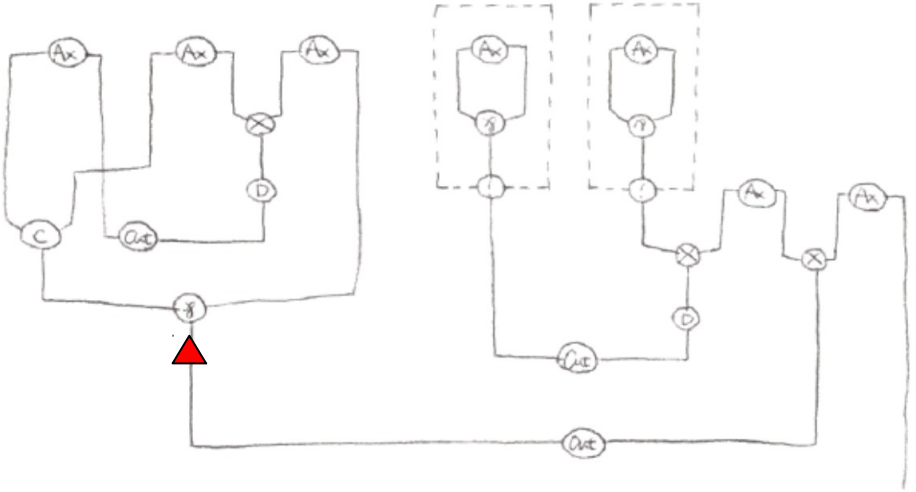




IAM vs. rewrites-first DGoIM

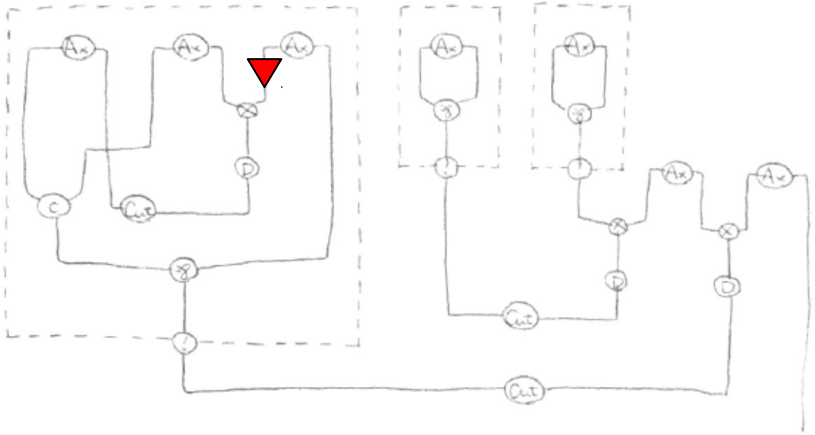
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

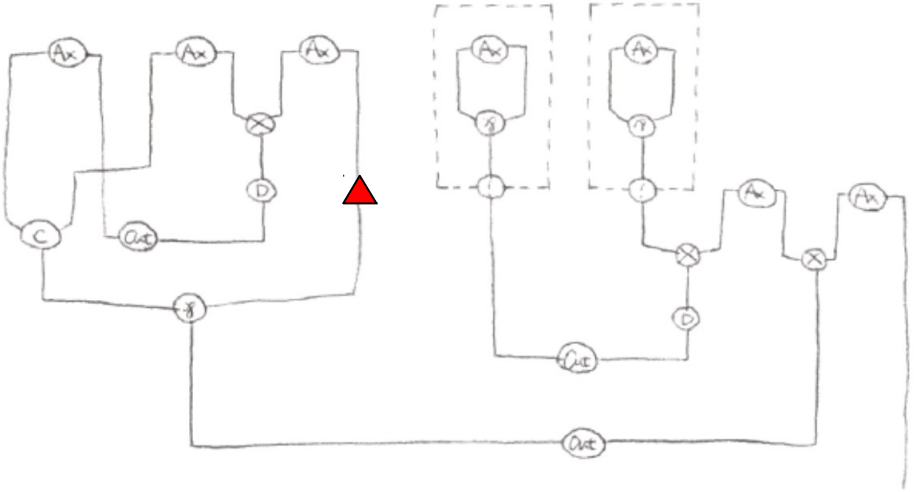




IAM vs. rewrites-first DGoIM

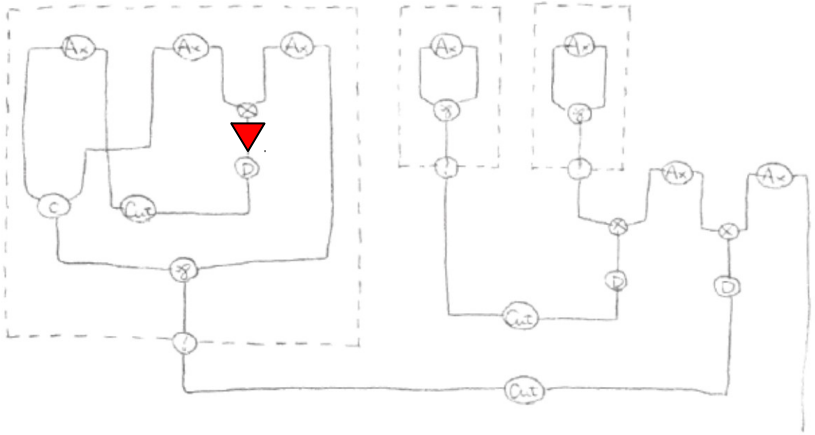
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

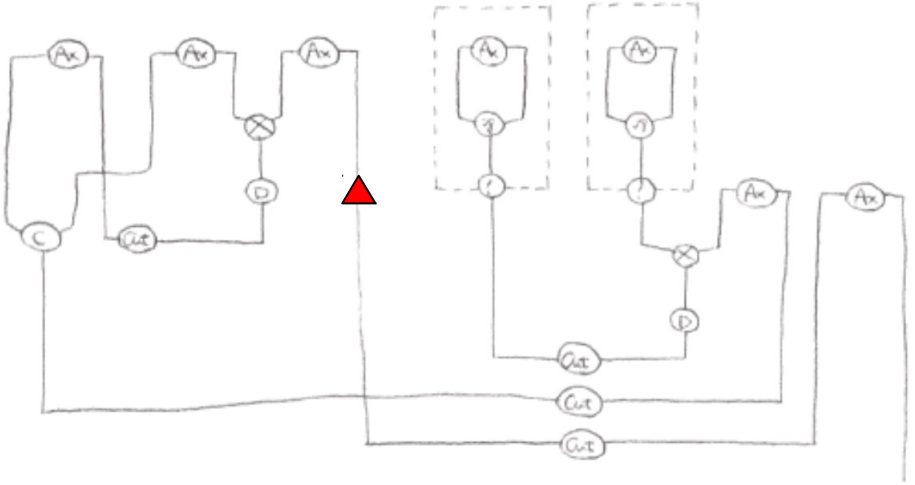




IAM vs. rewrites-first DGoIM

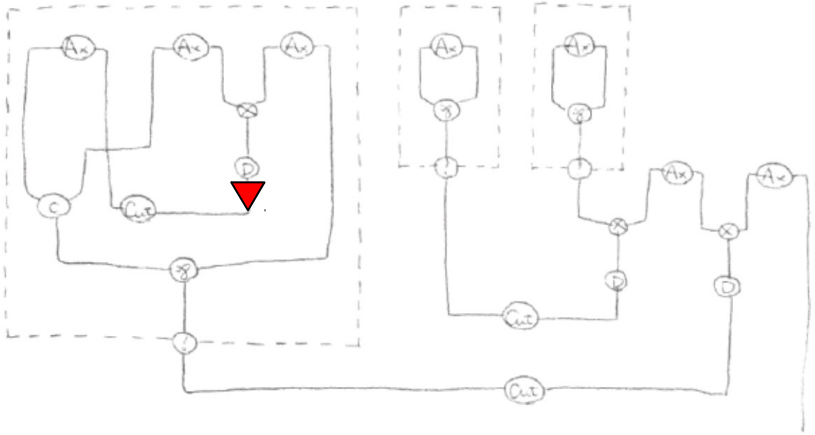
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

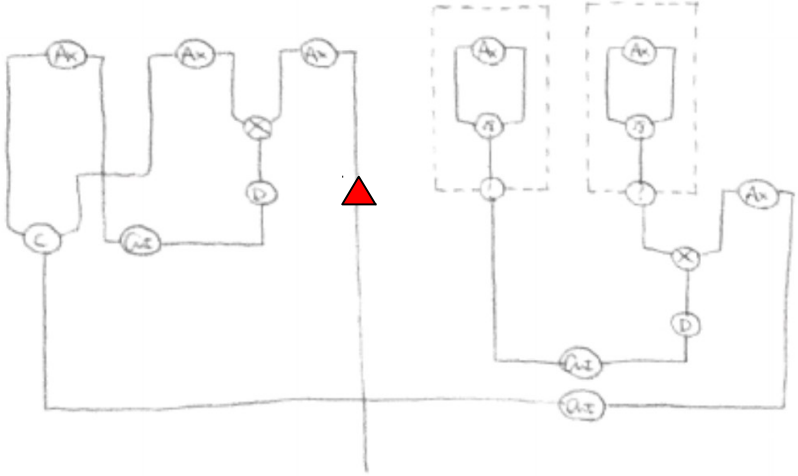




IAM vs. rewrites-first DGoIM

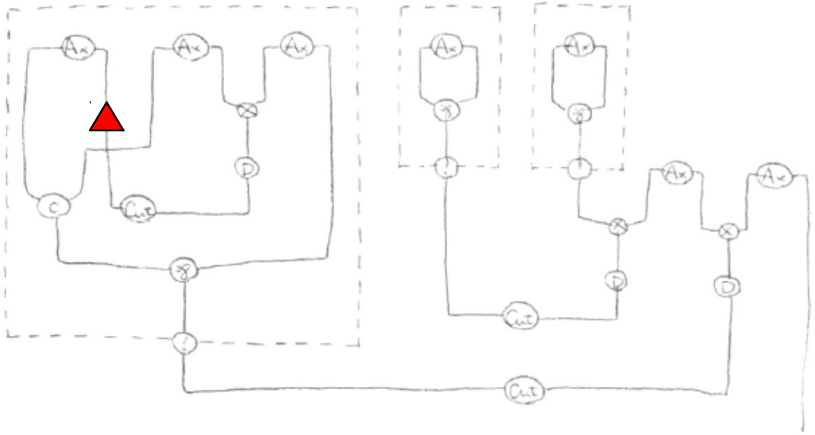
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

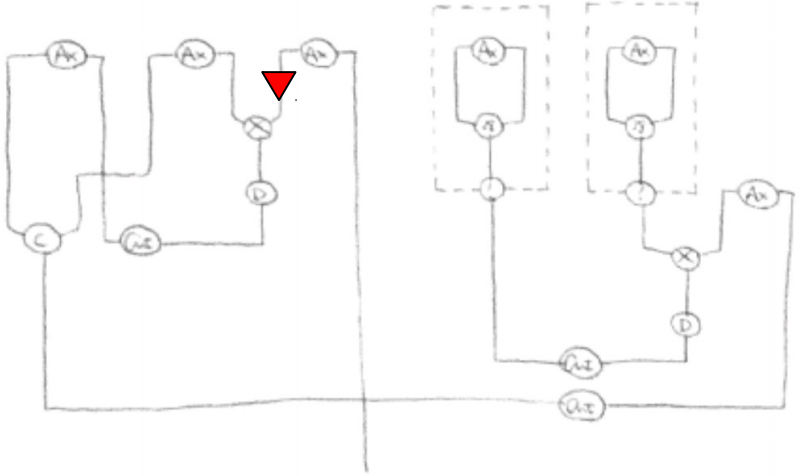




IAM vs. rewrites-first DGoIM

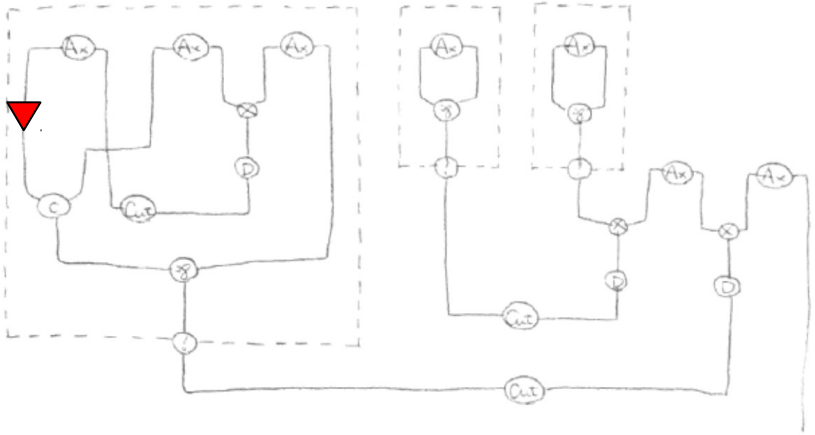
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

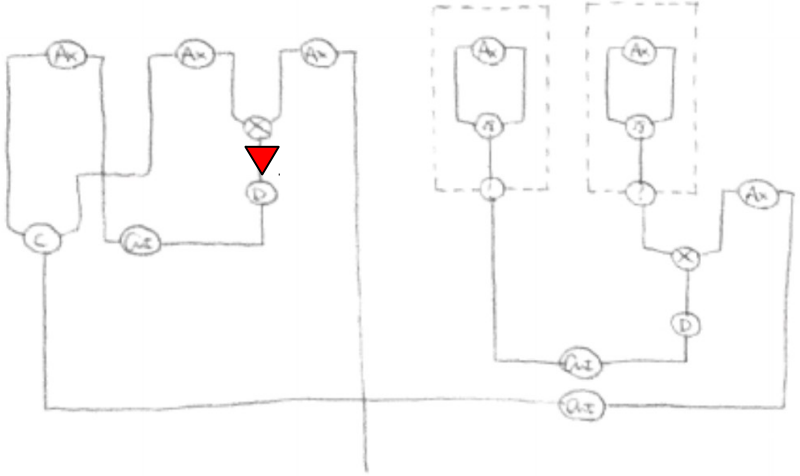




IAM vs. rewrites-first DGoIM

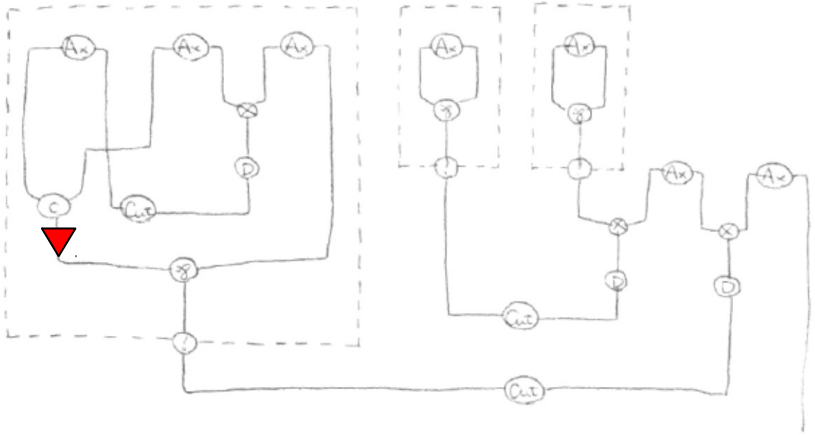
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

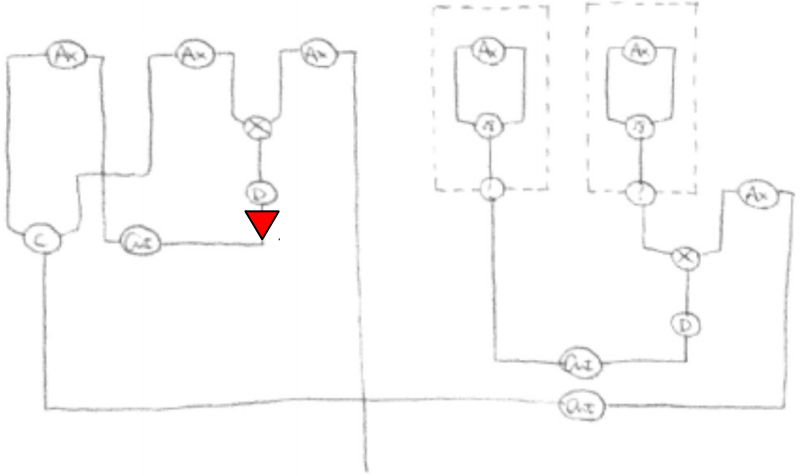




IAM vs. rewrites-first DGoIM

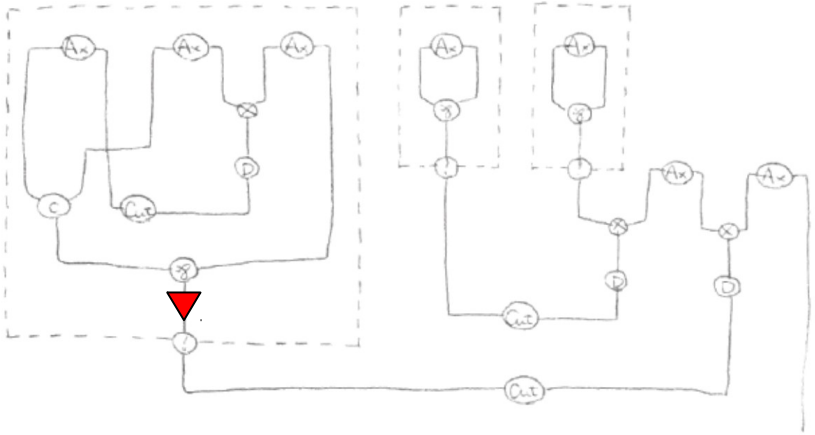
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

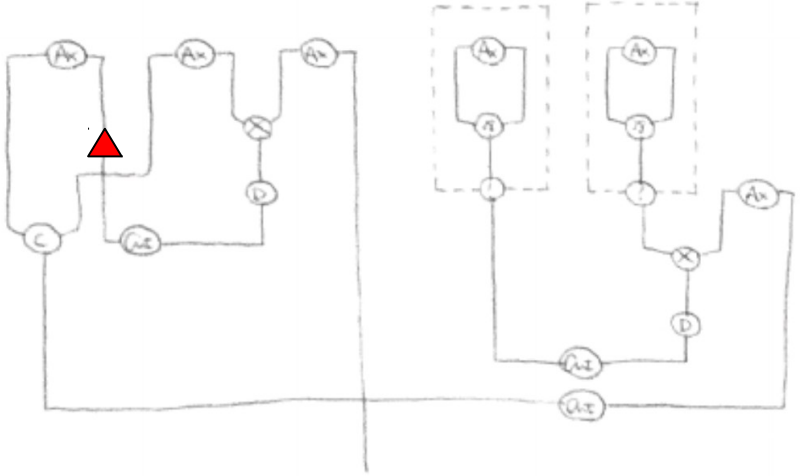




IAM vs. rewrites-first DGoIM

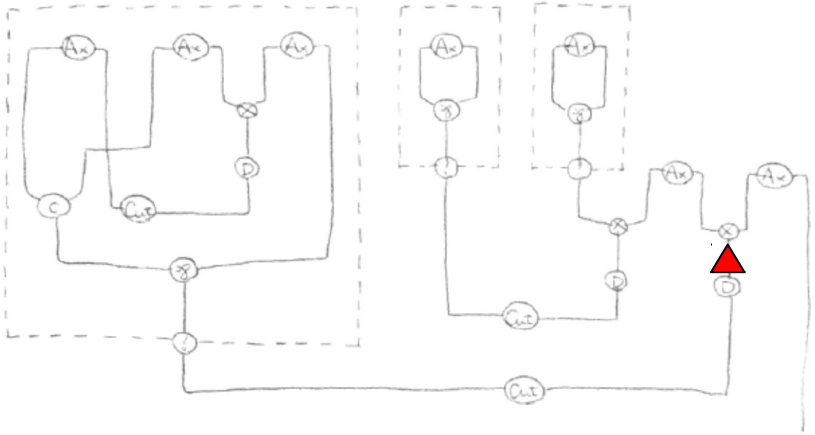
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

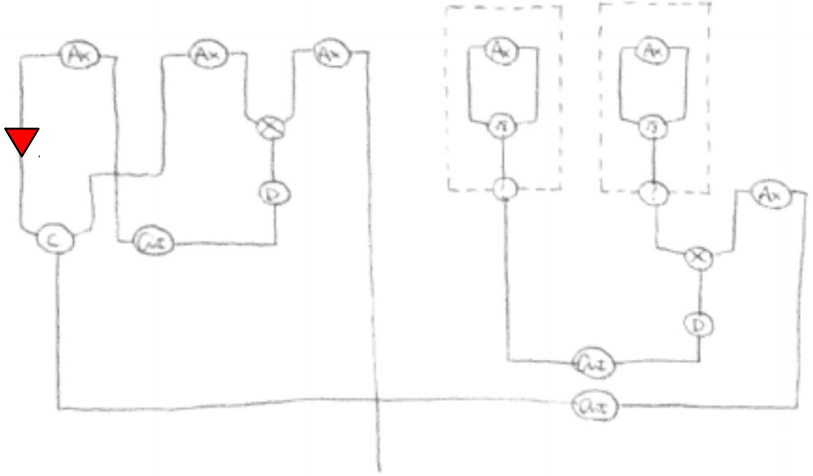




IAM vs. rewrites-first DGoIM

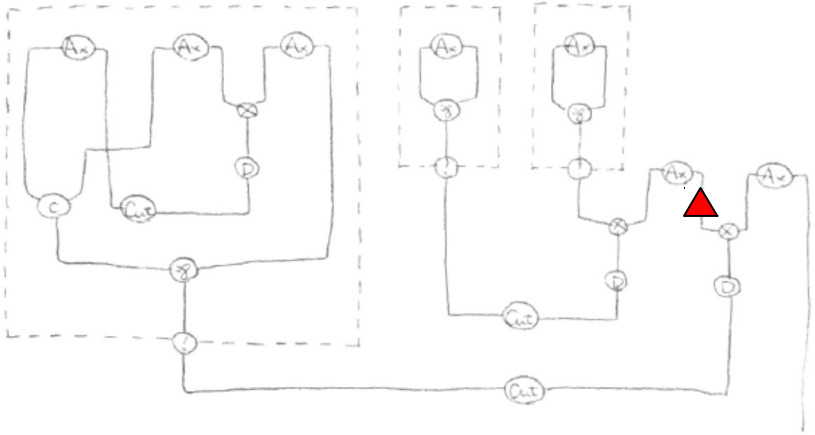
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

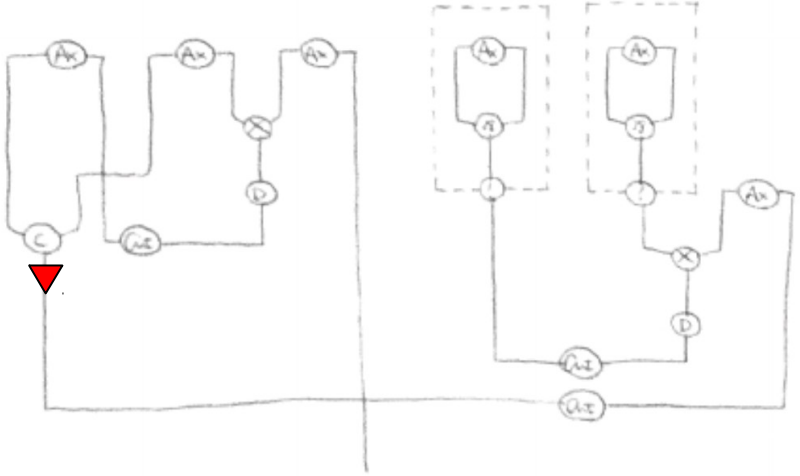




IAM vs. rewrites-first DGoIM

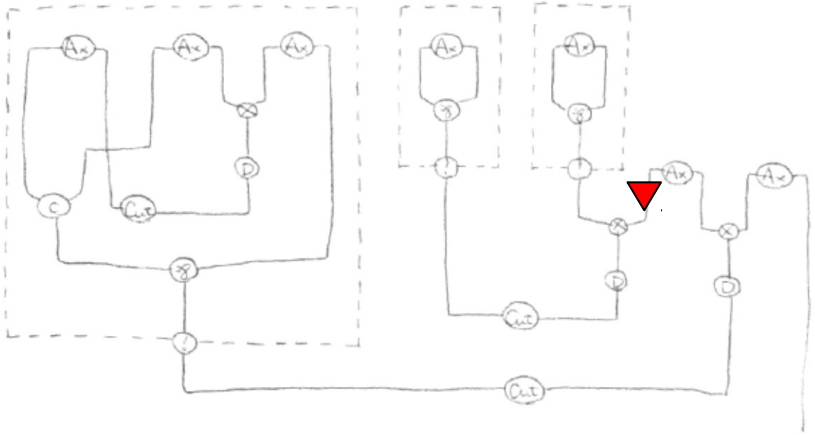
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

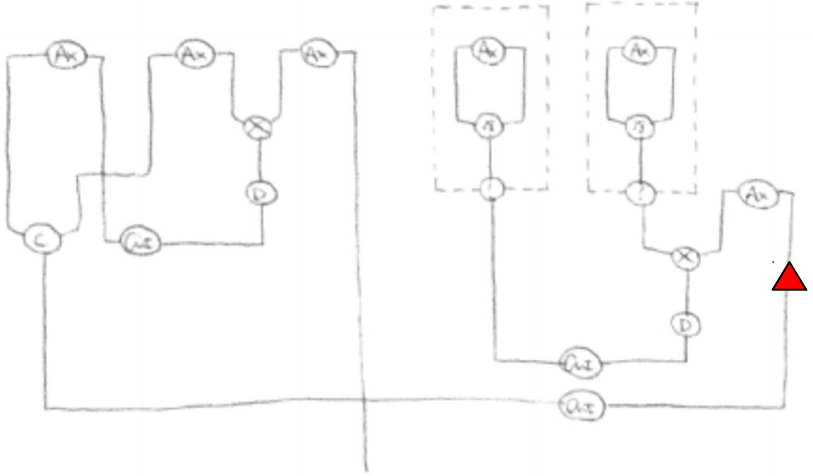




IAM vs. rewrites-first DGoIM

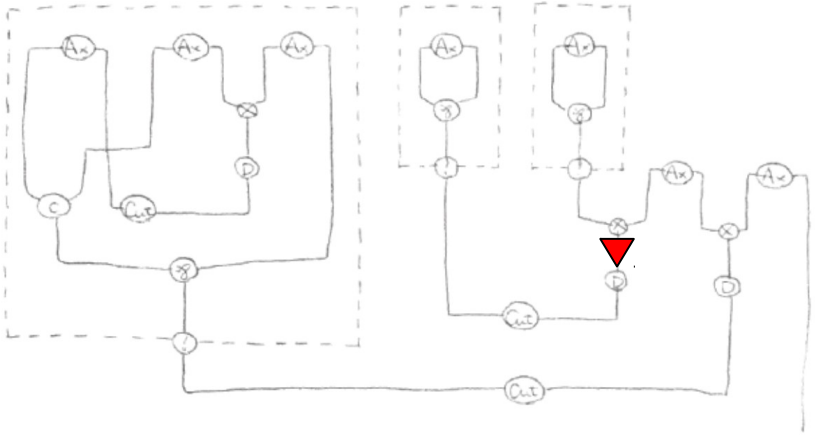
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

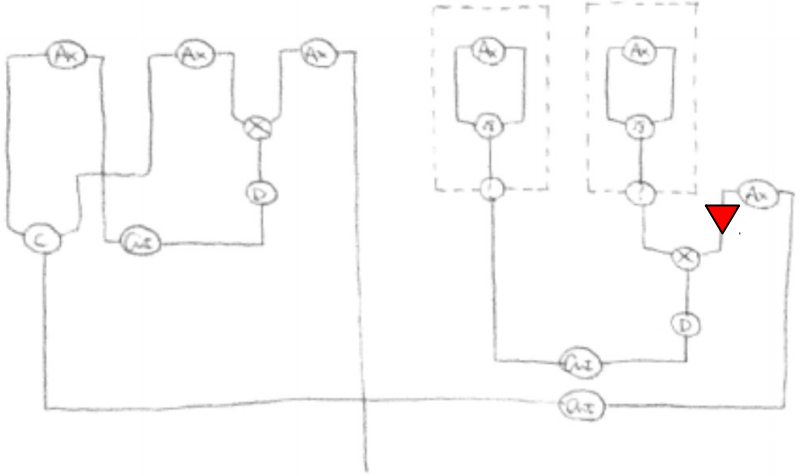




IAM vs. rewrites-first DGoIM

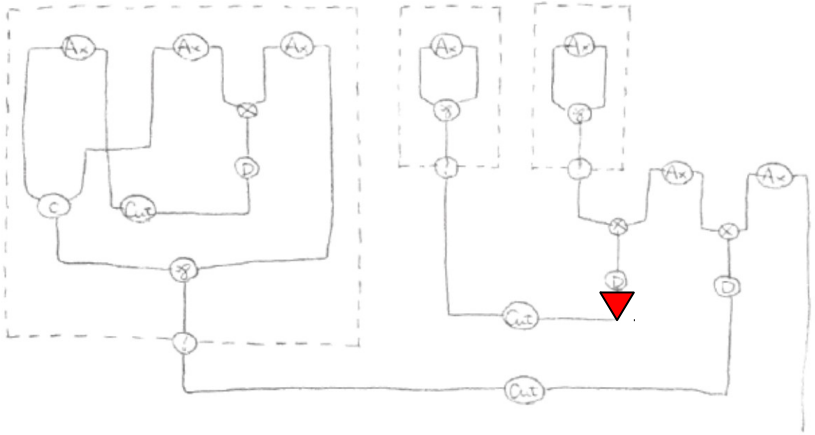
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

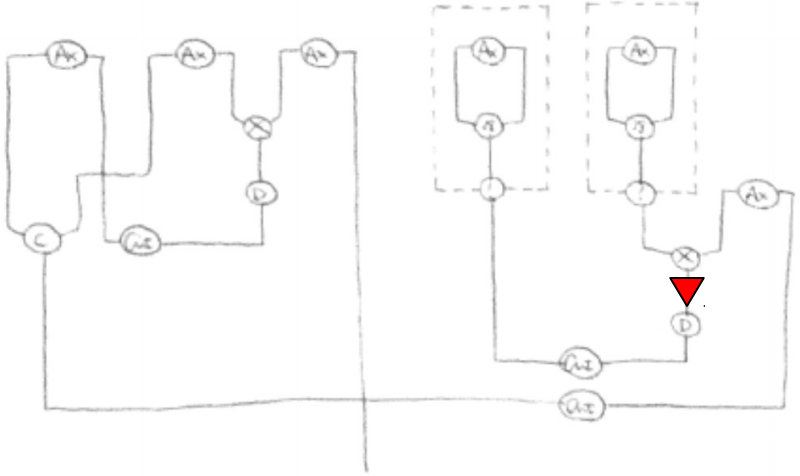




IAM vs. rewrites-first DGoIM

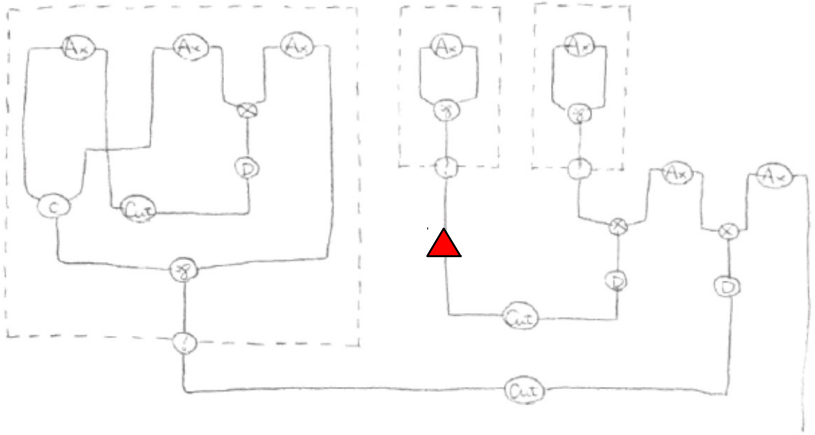
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

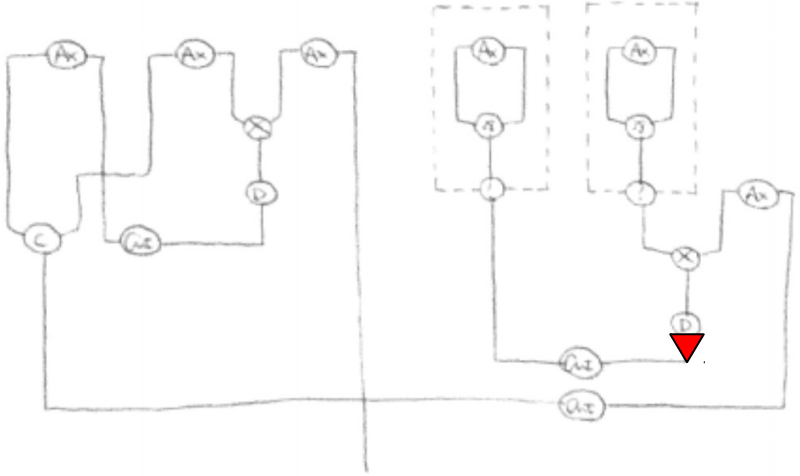




IAM vs. rewrites-first DGoIM

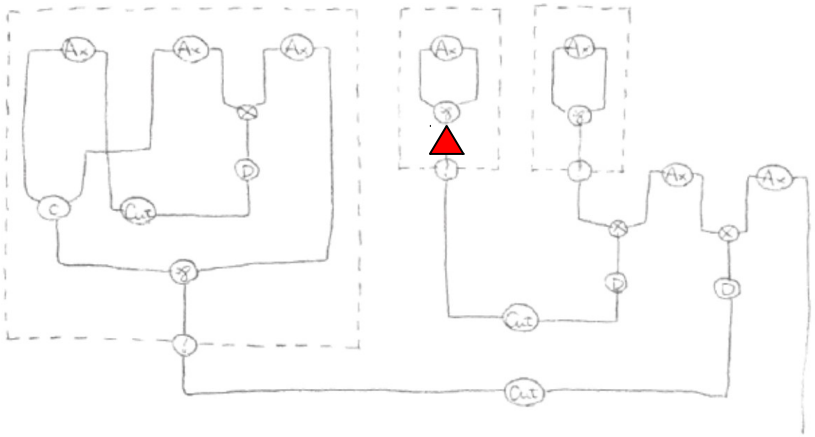
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

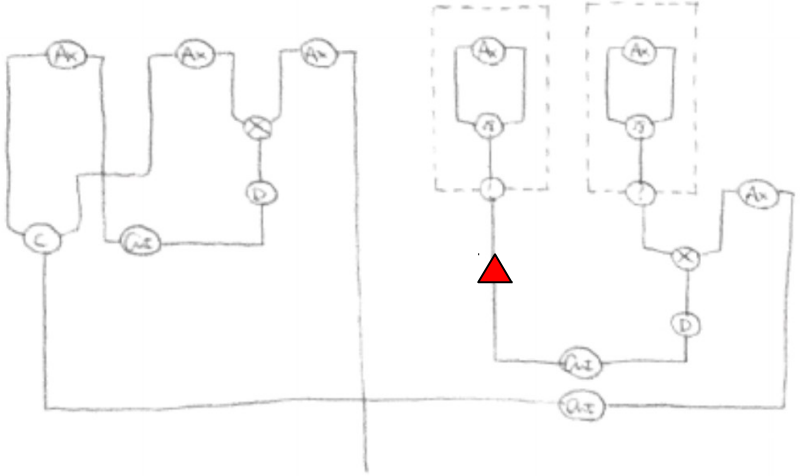




IAM vs. rewrites-first DGoIM

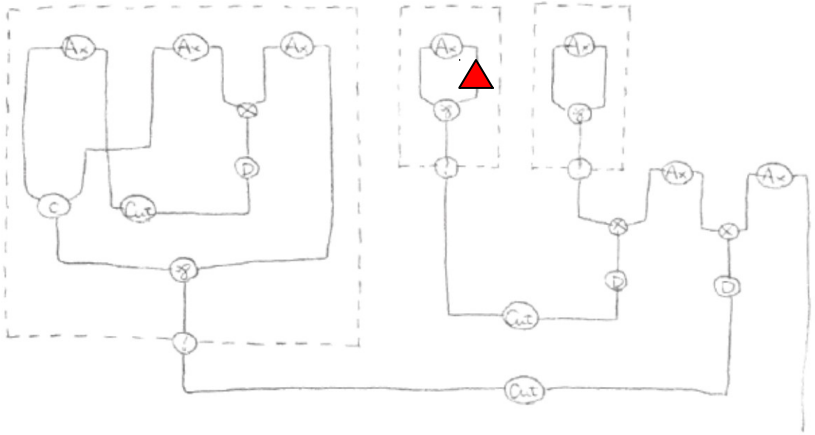
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

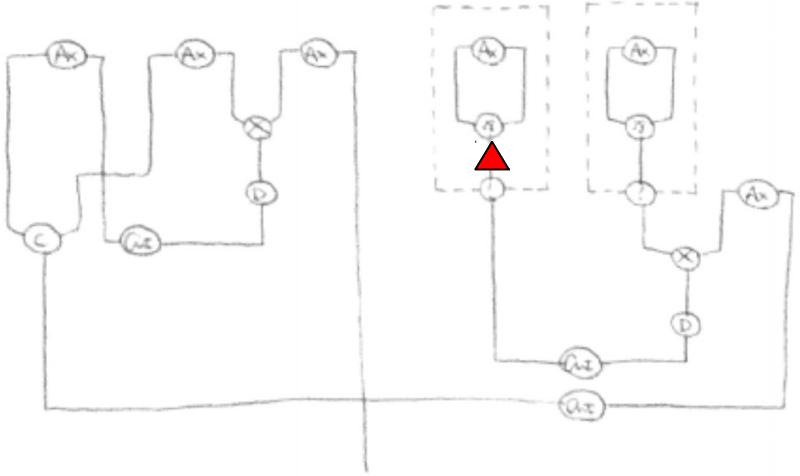




IAM vs. rewrites-first DGoIM

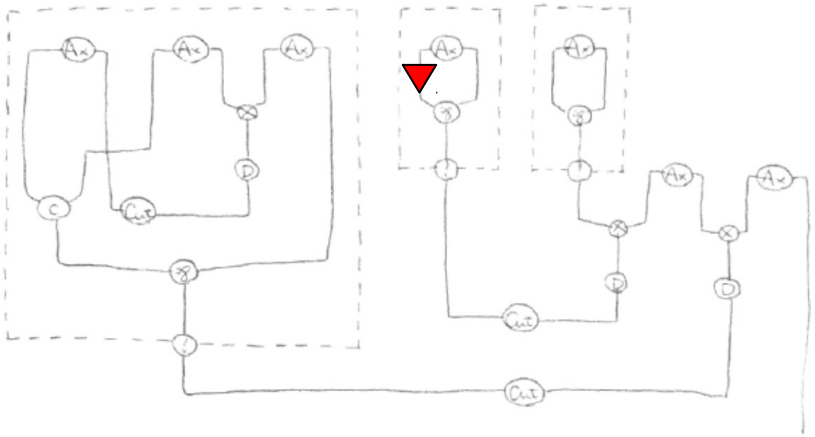
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

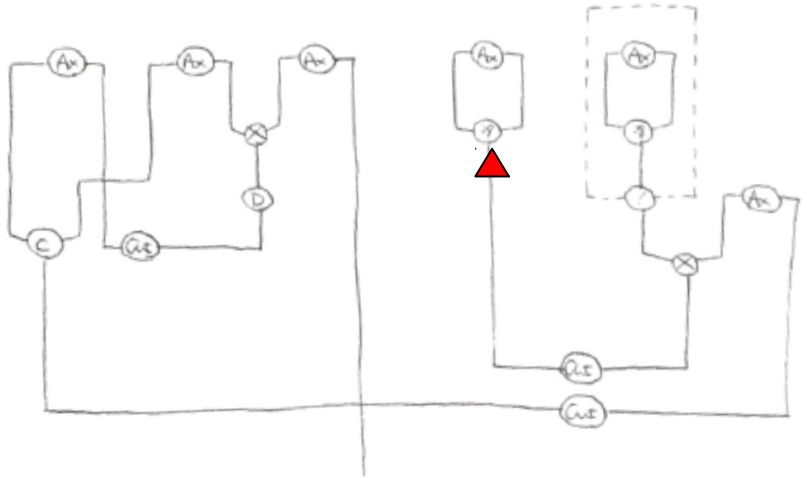




IAM vs. rewrites-first DGoIM

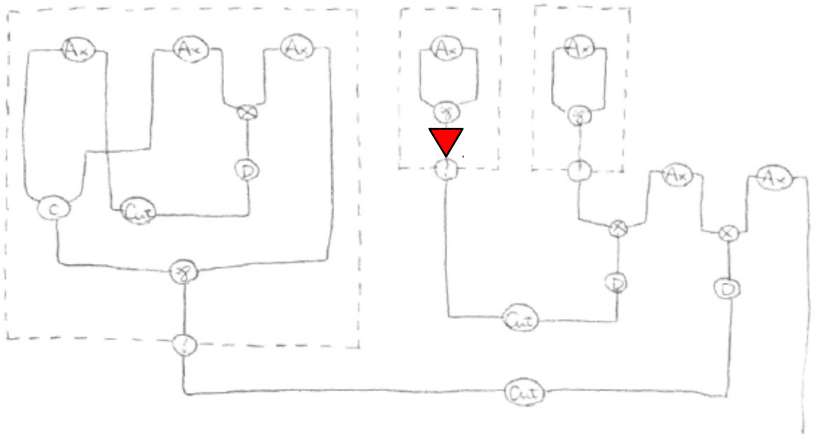
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

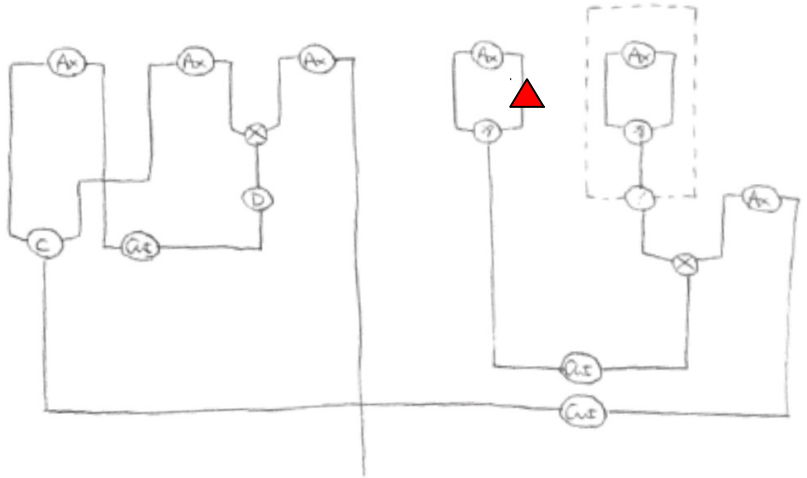




IAM vs. rewrites-first DGoIM

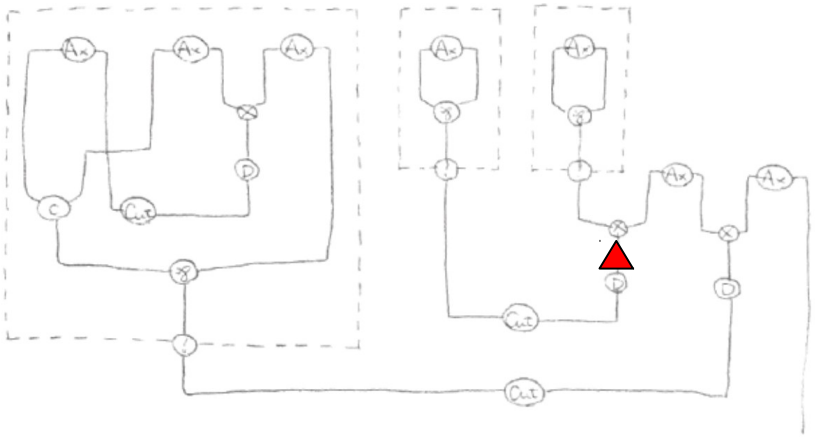
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

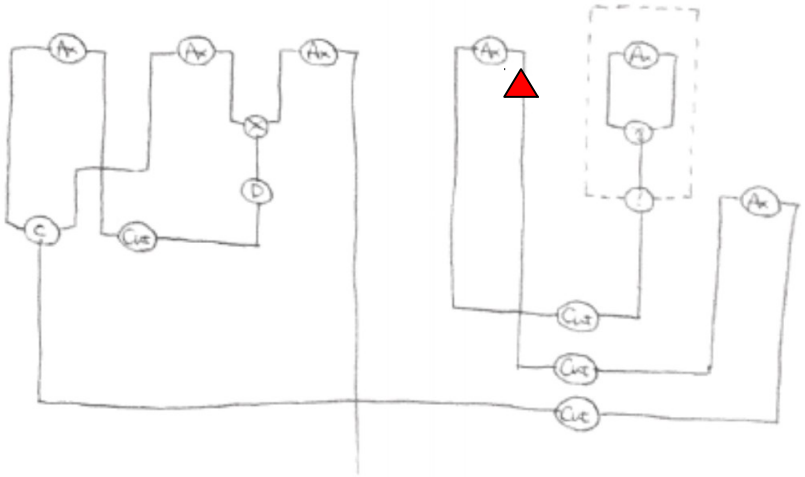




IAM vs. rewrites-first DGoIM

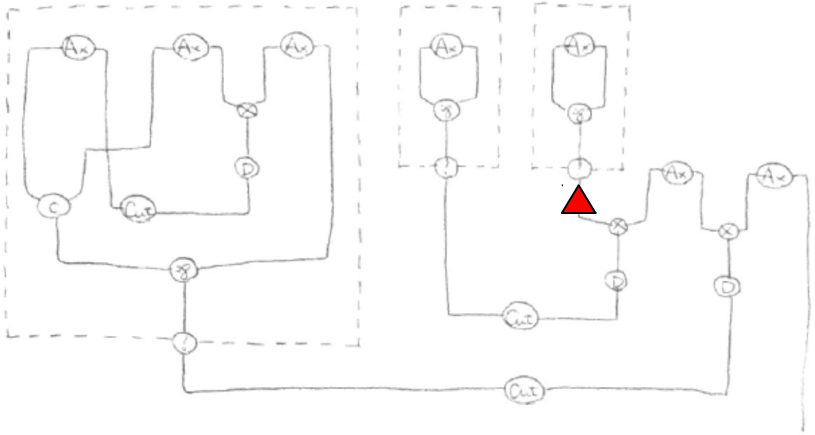
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

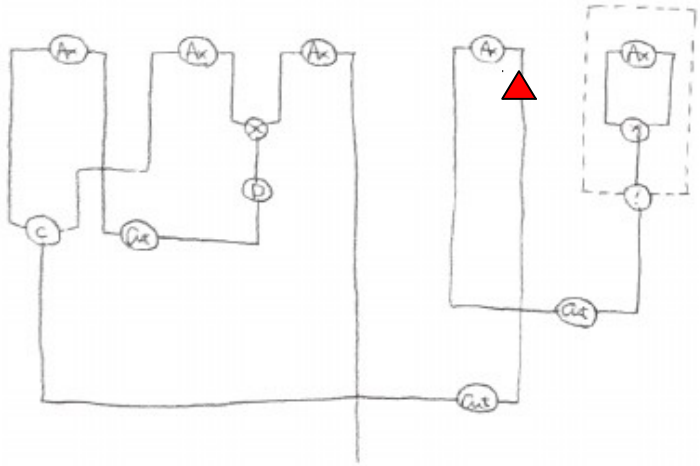




IAM vs. rewrites-first DGoIM

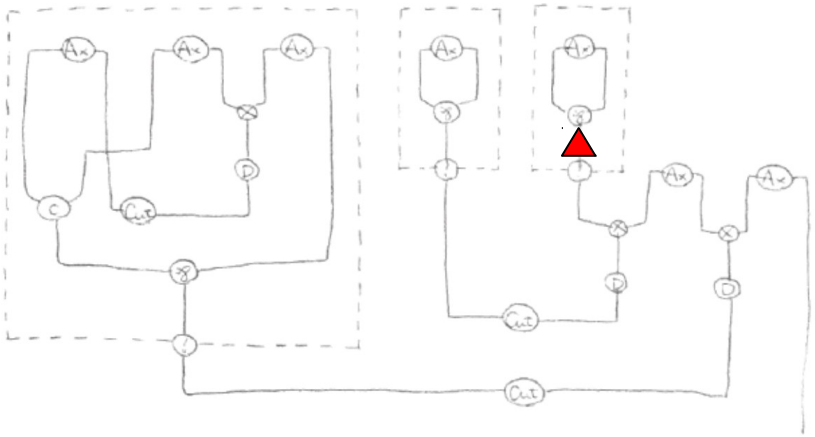
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

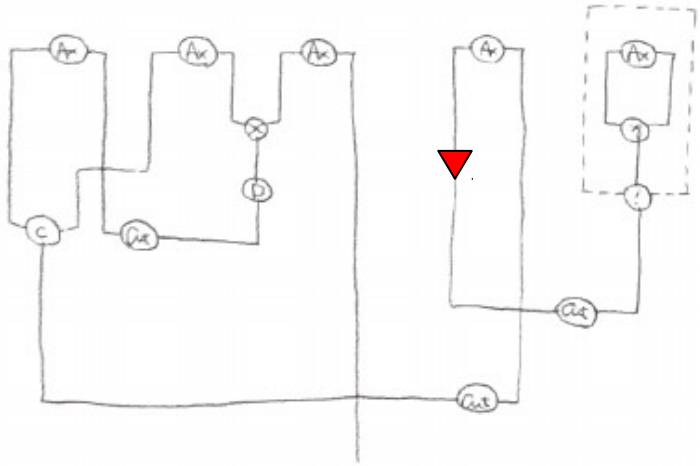




IAM vs. rewrites-first DGoIM

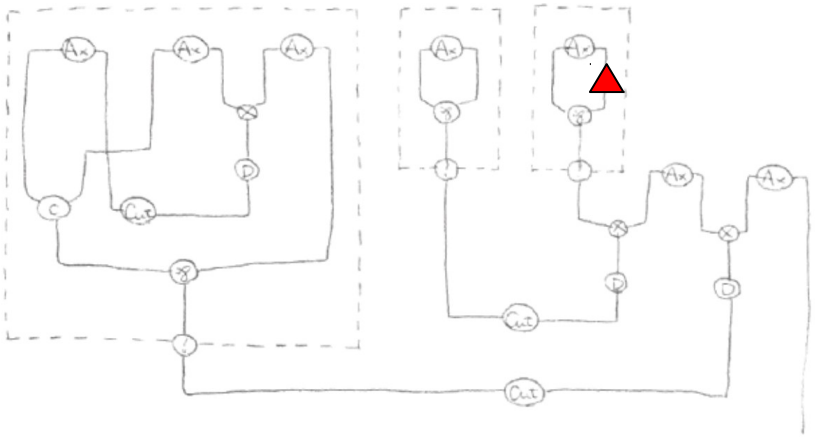
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

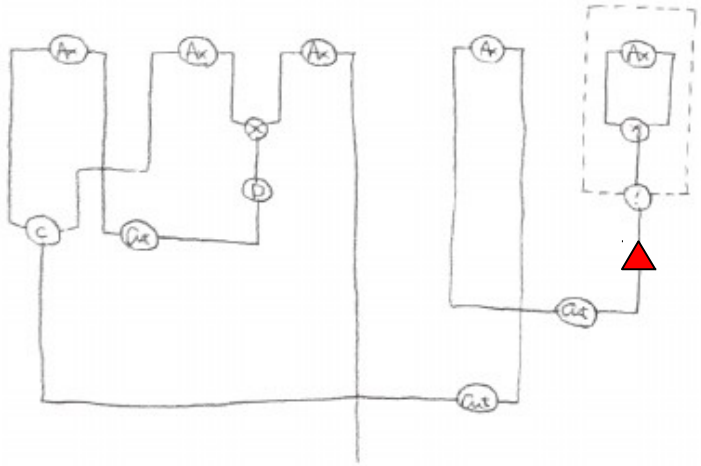




IAM vs. rewrites-first DGoIM

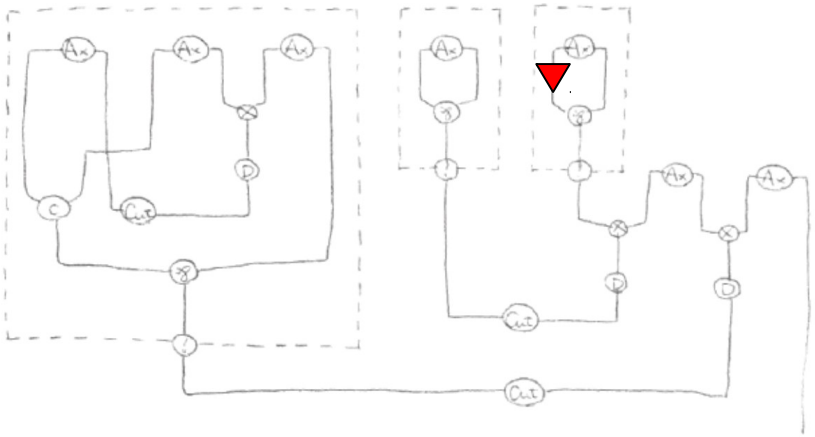
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

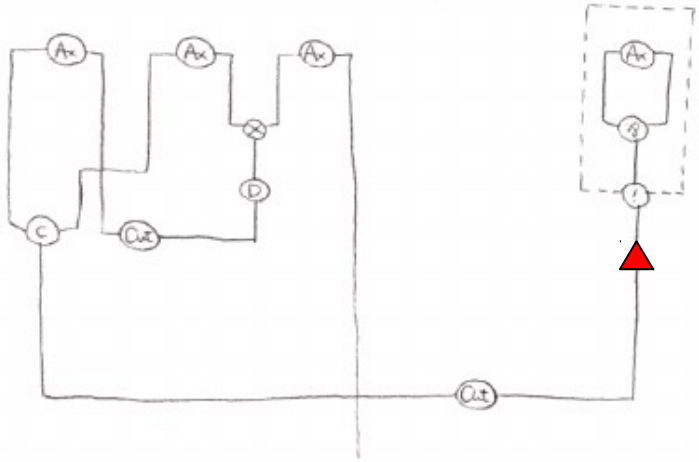




IAM vs. rewrites-first DGoIM

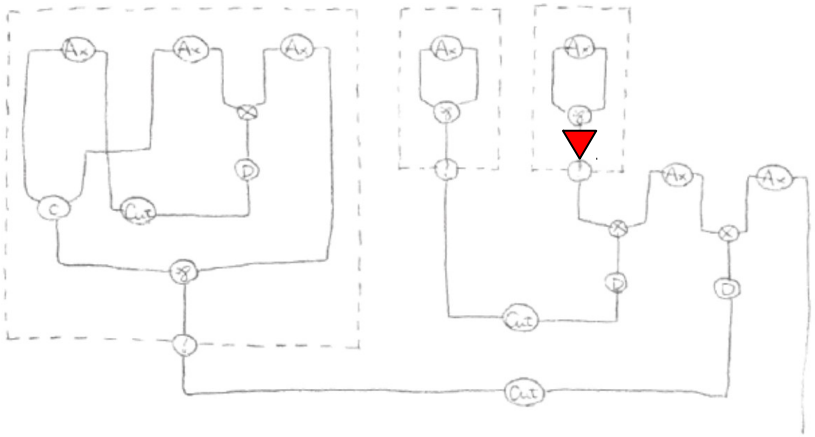
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

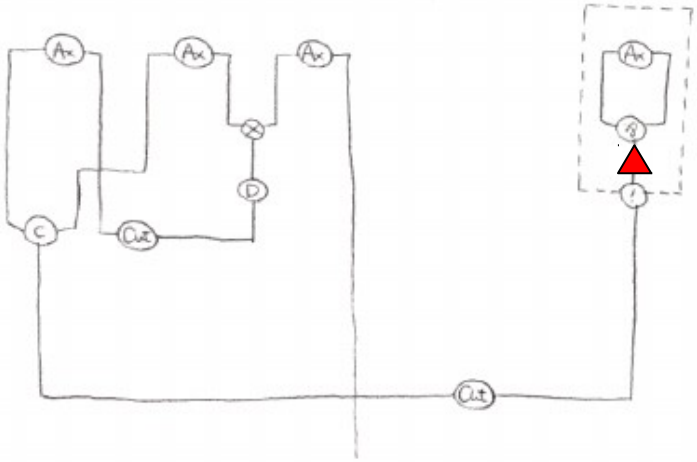




IAM vs. rewrites-first DGoIM

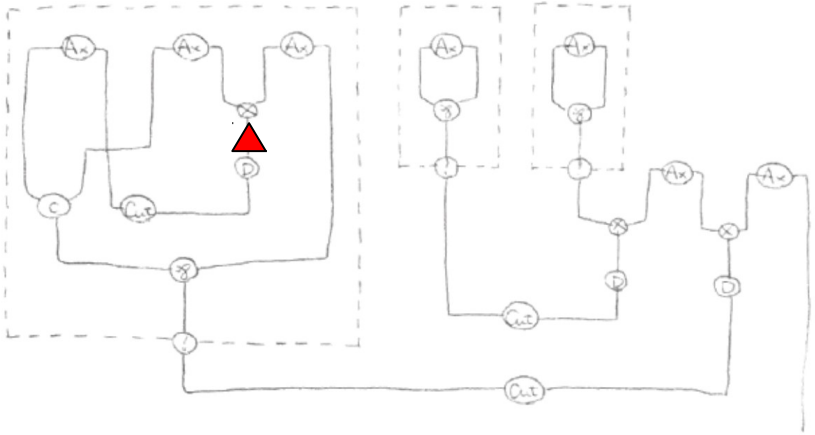
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

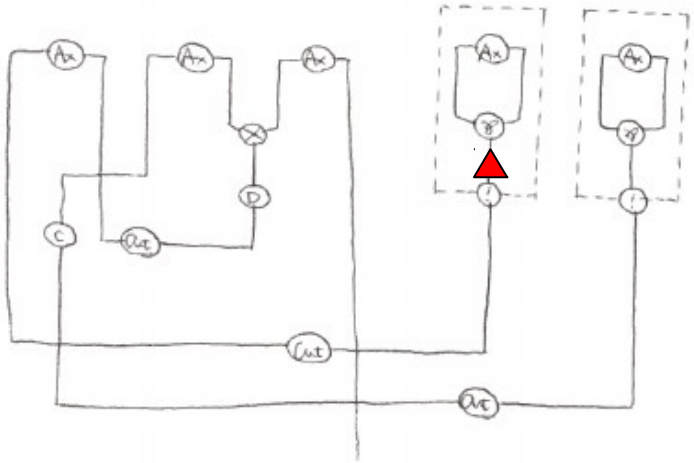




IAM vs. rewrites-first DGoIM

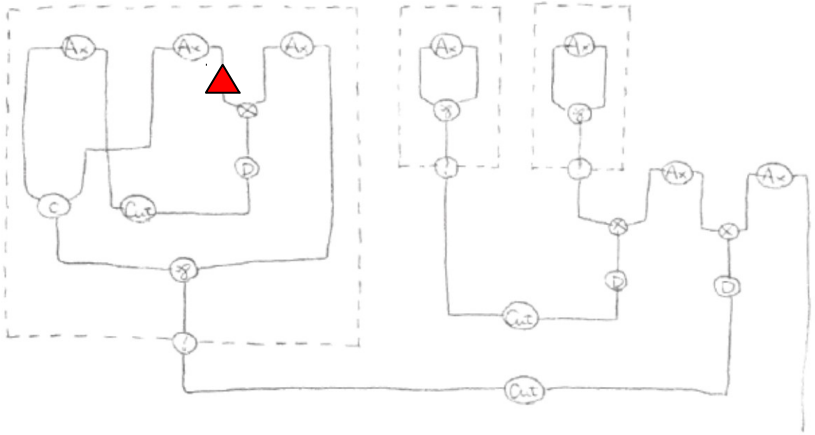
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

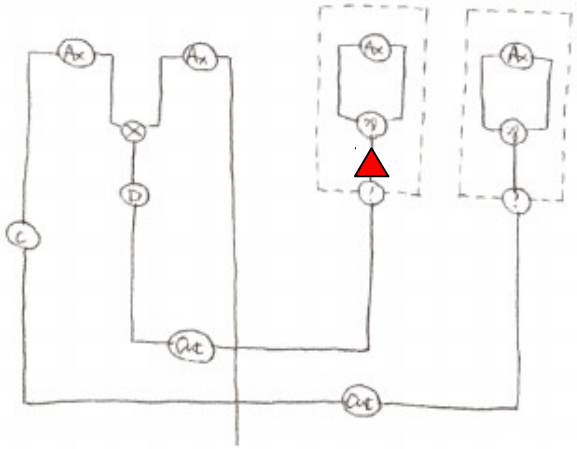




IAM vs. rewrites-first DGoIM

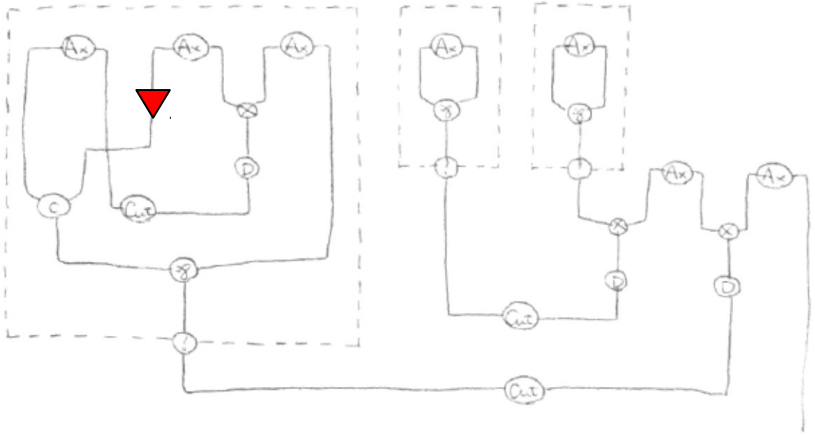
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

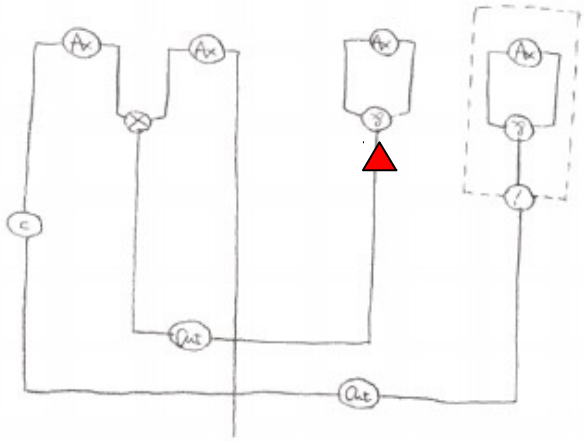




IAM vs. rewrites-first DGoIM

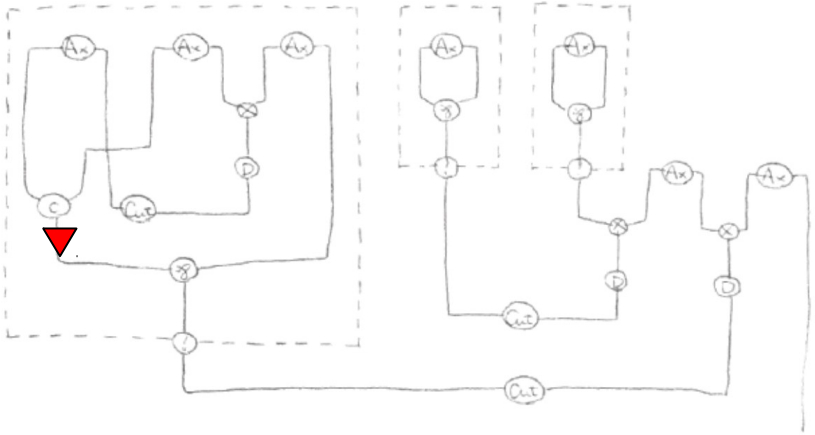
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

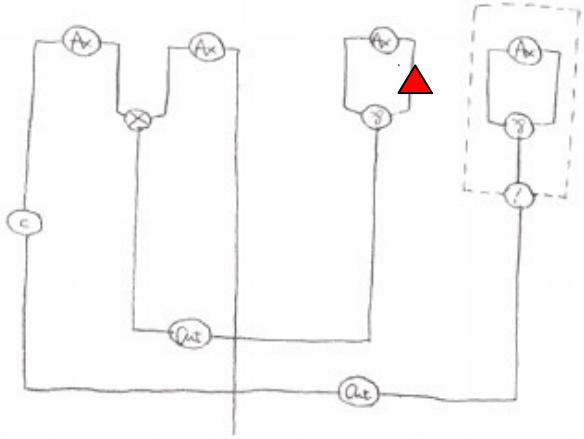




IAM vs. rewrites-first DGoIM

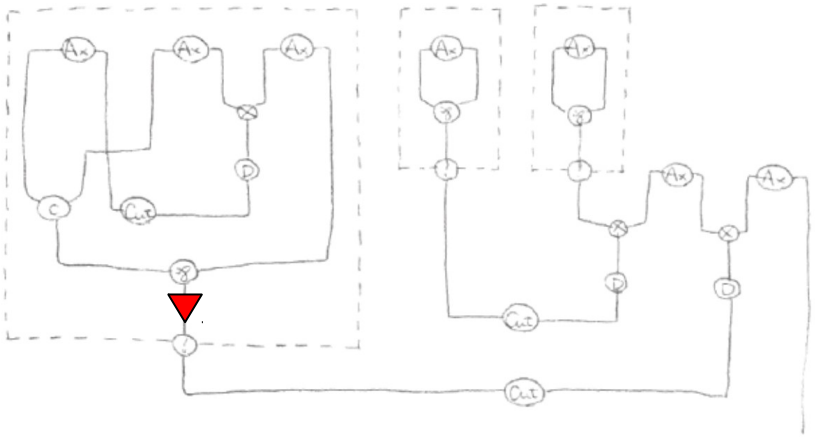
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

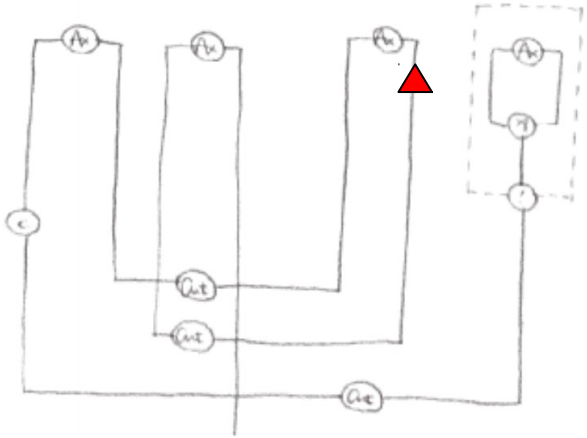




IAM vs. rewrites-first DGoIM

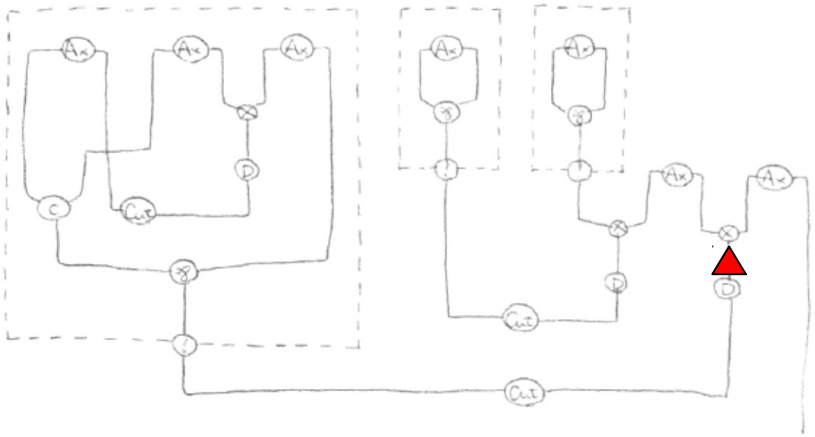
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

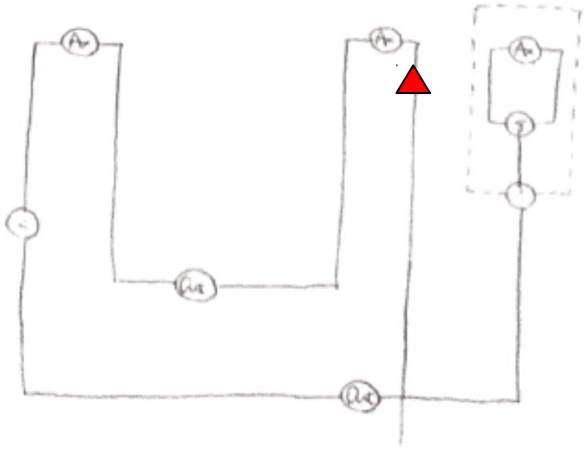




IAM vs. rewrites-first DGoIM

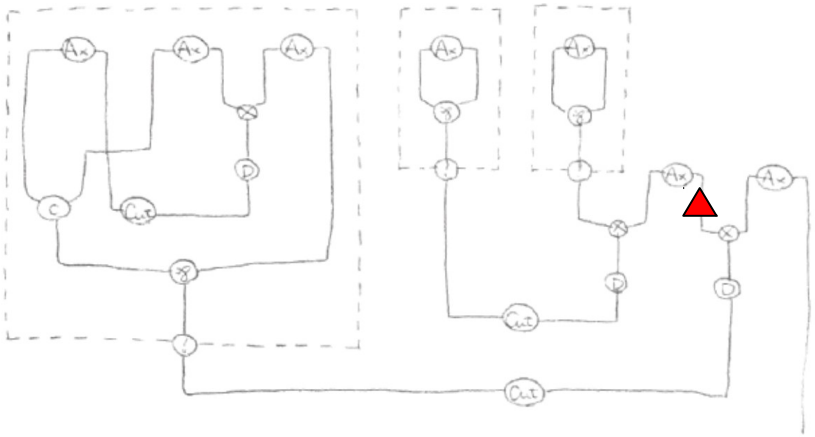
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

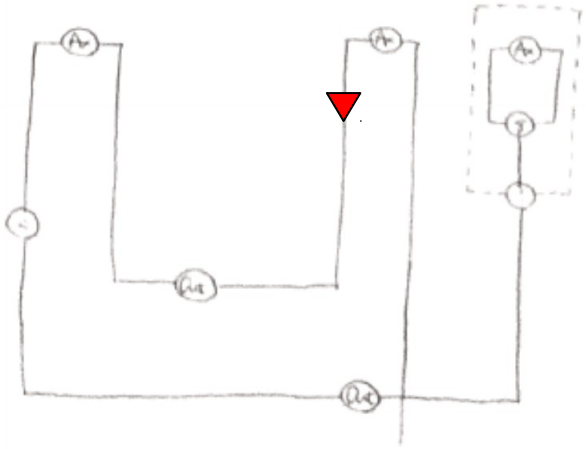




IAM vs. rewrites-first DGoIM

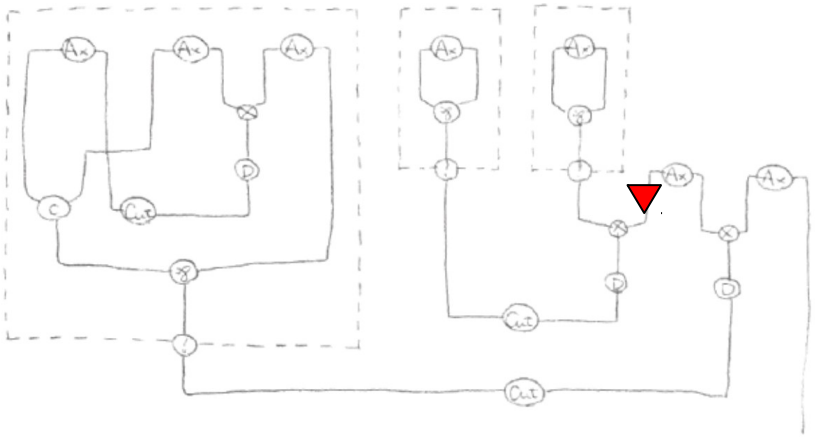
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

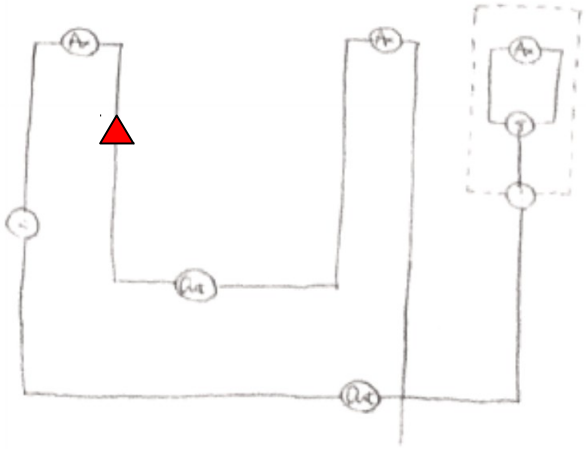




IAM vs. rewrites-first DGoIM

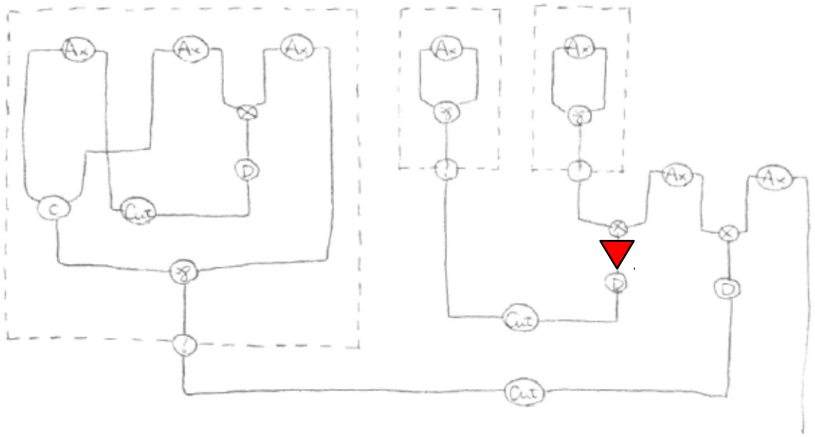
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

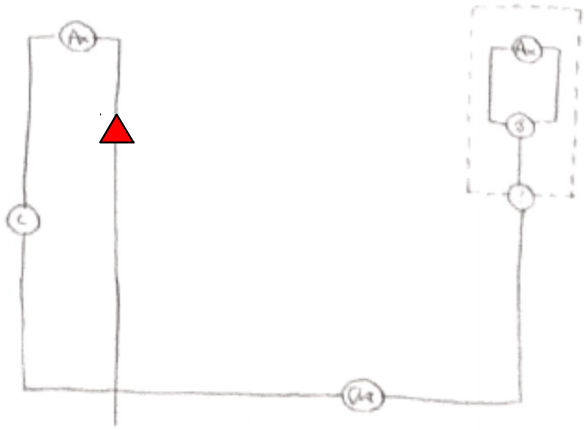




IAM vs. rewrites-first DGoIM

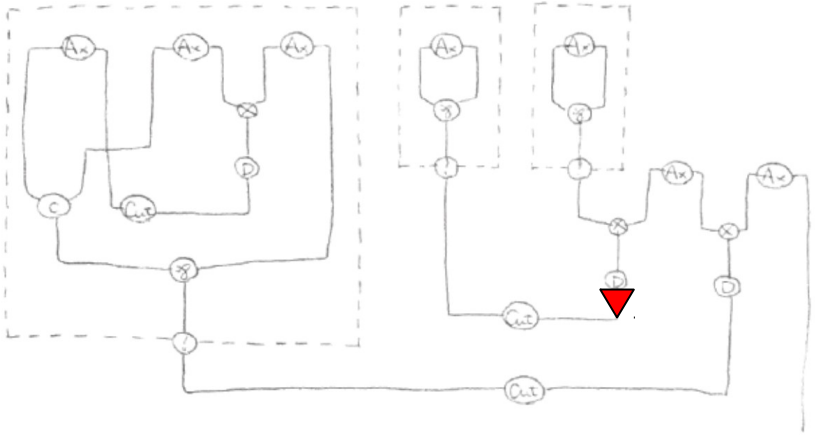
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

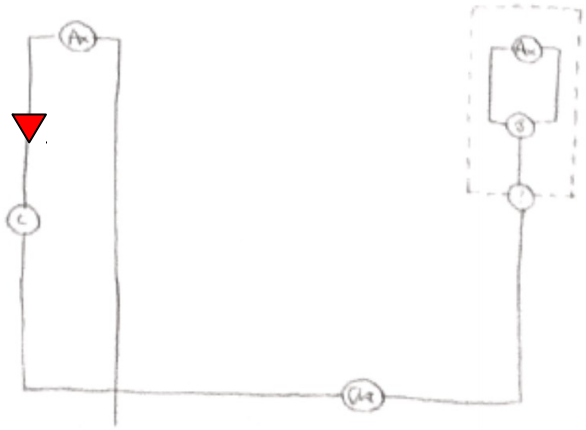




IAM vs. rewrites-first DGoIM

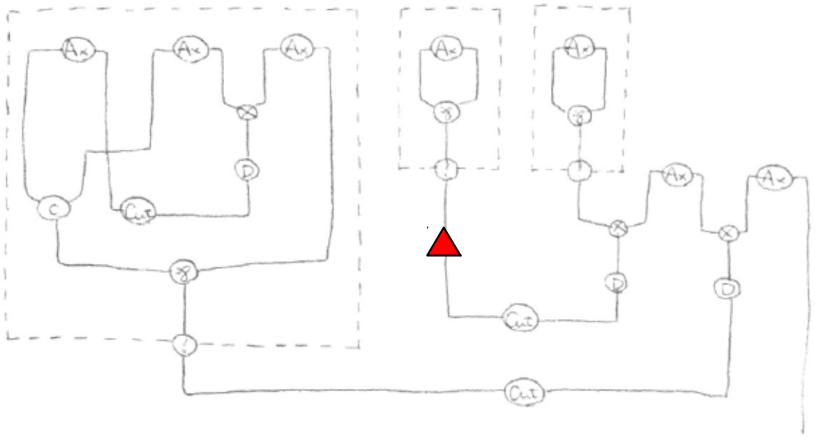
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

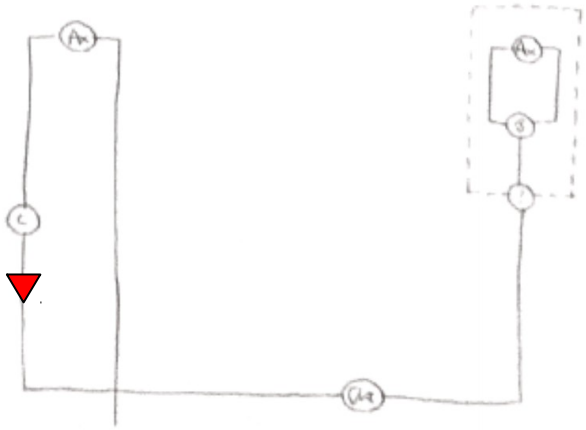




IAM vs. rewrites-first DGoIM

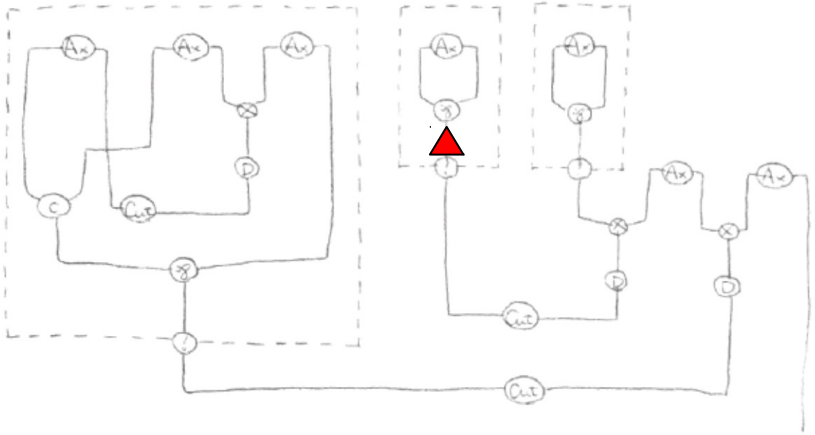
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

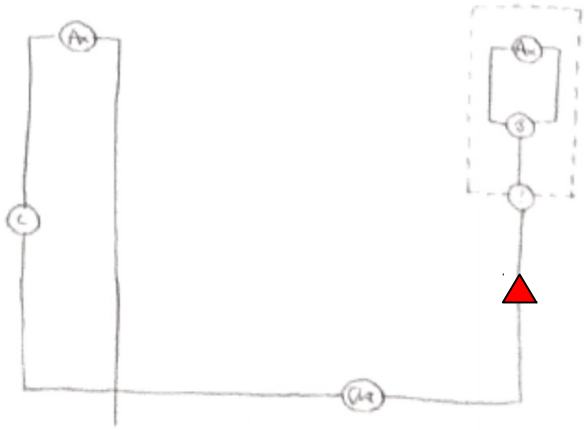




IAM vs. rewrites-first DGoIM

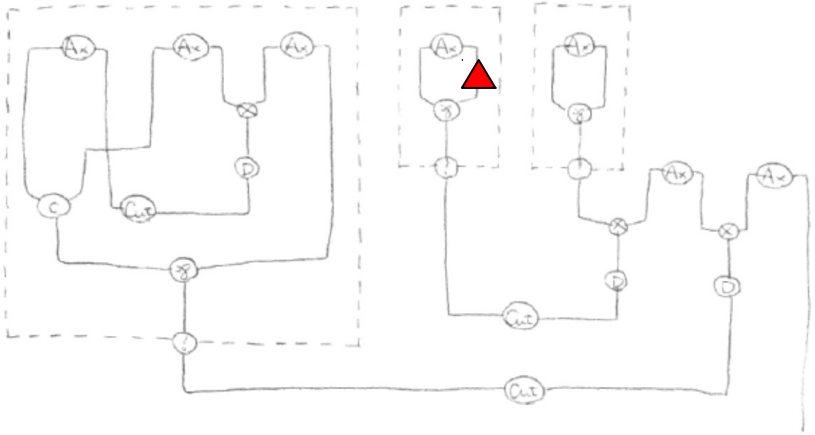
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

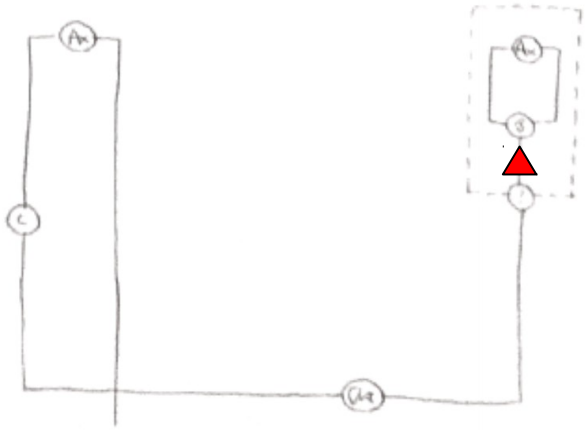




IAM vs. rewrites-first DGoIM

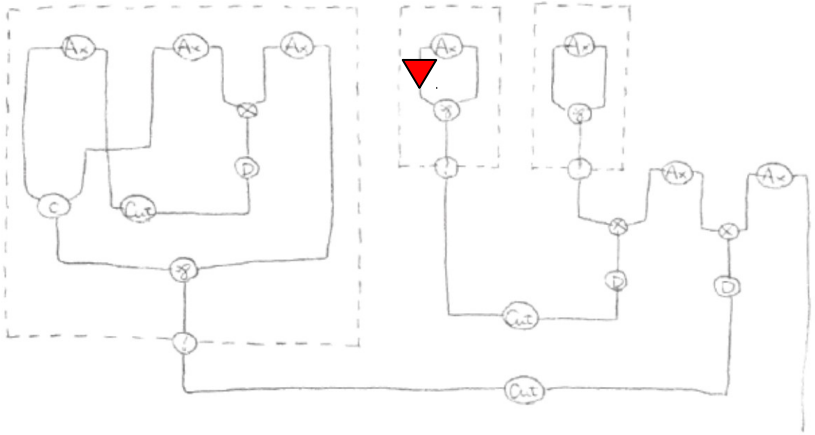
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

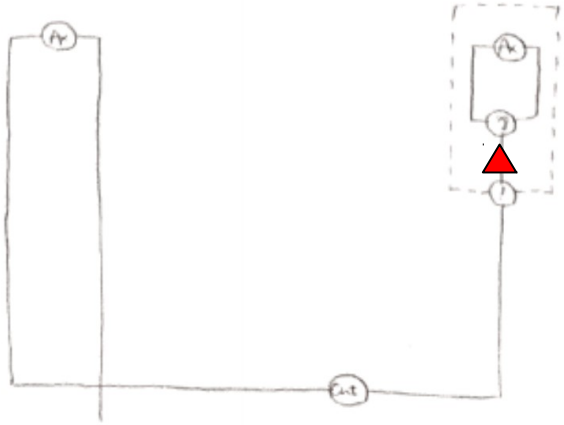




IAM vs. rewrites-first DGoIM

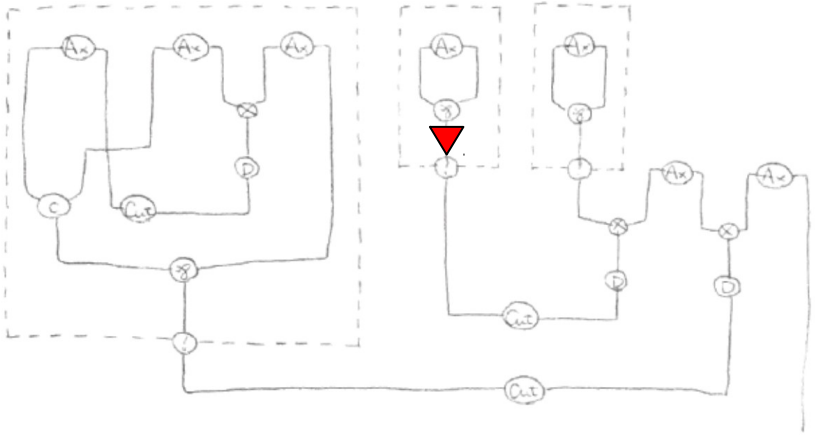
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

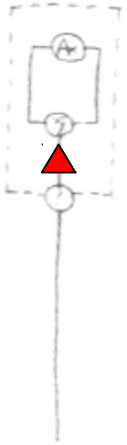




IAM vs. rewrites-first DGoIM

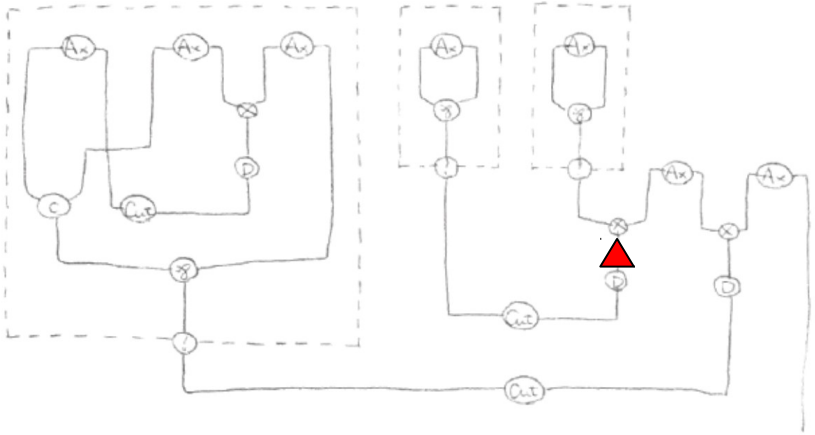
$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

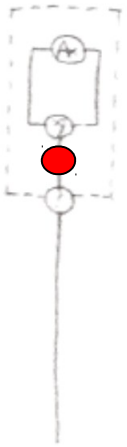




IAM vs. rewrites-first DGoIM

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

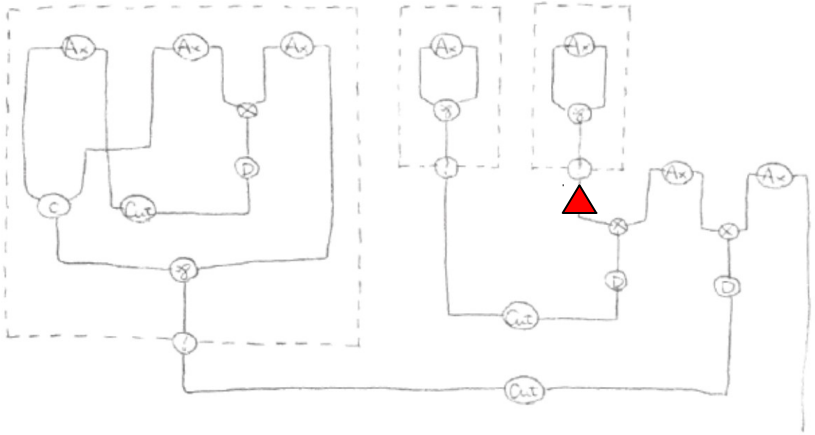


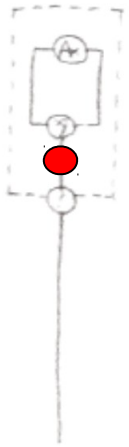


29 passes
16 rewrites

IAM vs. rewrites-first DGoIM

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

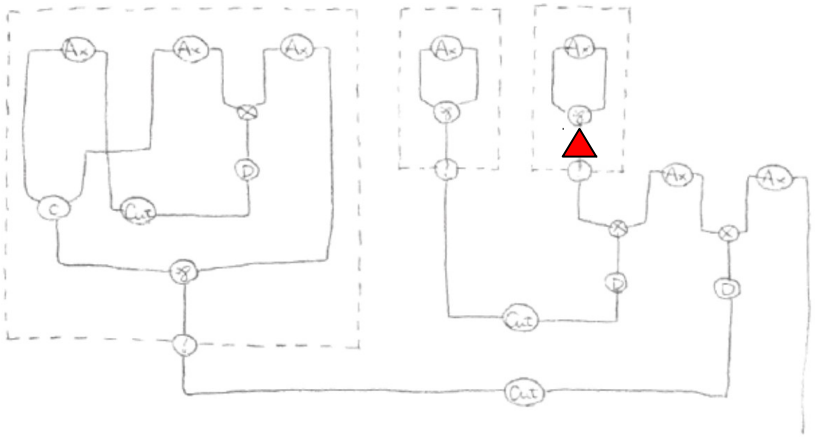


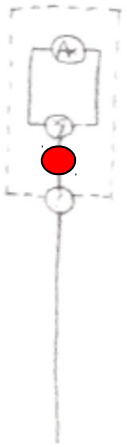


29 passes
16 rewrites

IAM vs. rewrites-first DGoIM

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$

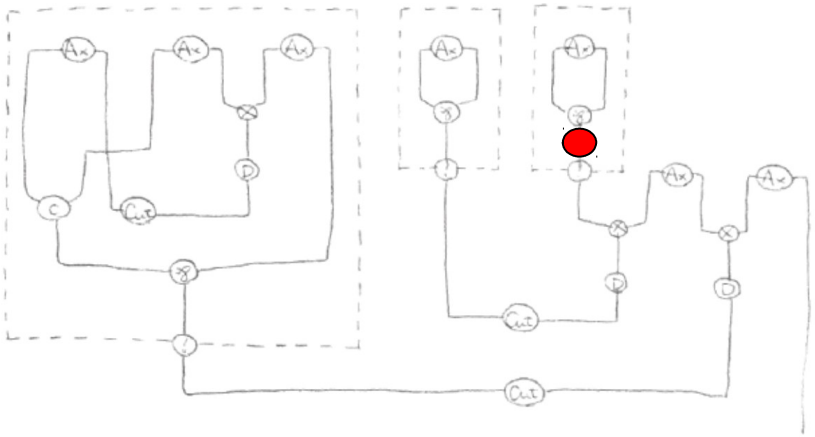




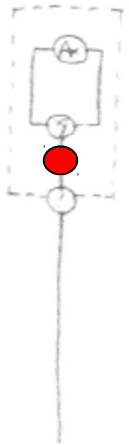
29 passes
16 rewrites

IAM vs. rewrites-first DGoIM

$$(\lambda x.x\,x)\,((\lambda y.y)\,(\lambda z.z)) \;\Downarrow\; \lambda z.z$$



48 passes



29 passes
16 rewrites

Quantitative analysis by GoI-style token passing

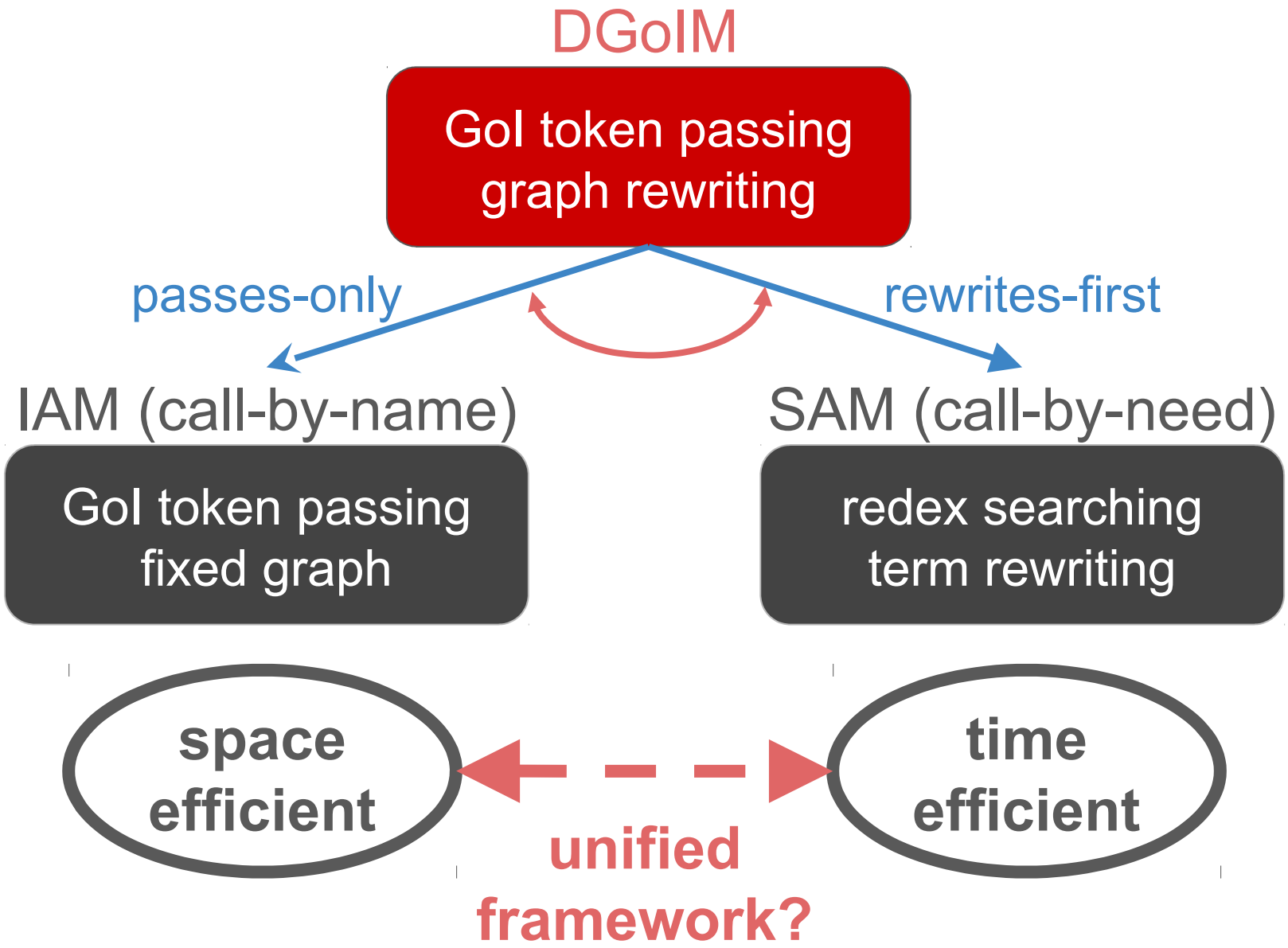
of

semantics of linear logic

space-time trade-off of program execution cost

abstract machines for lambda-calculus

DGoIM: *flexible interleaving*



Future work

- flexible interleaving
  - IAM (call-by-name) -- SAM (call-by-need) spectrum
  - call-by-value?
- token-guided rewriting of “higher-order” data-flow graphs?
  - TensorFlow (https://www.tensorflow.org/)
  - self-adjusting computation (http://www.umut-acar.org/self-adjusting-computation)