From Reactions to Observations: the Directed Bigraphical Model

Davide Grohmann, **Marino Miculan**

University of Udine

IOC Tallinn, March 15, 2007

Labelled transition systems

- Labelled transition systems are relations of the form

$$P \xrightarrow{a} Q$$

  where P, Q are systems (processes, programs with state, etc. . .) and a is a *label*, that is an *observation*
- LTS are used for defining the behaviour of calculi/systems because they endorse most important techniques for verifying properties (e.g., *model checking*) and observational equivalence (e.g., *bisimulations*)
- the labels should be enough to describe faithfully the aspects we are observing, still not too many to be impractible to use.
- In general good LTS are difficult to describe, and often many ad hoc choices can be done (compare e.g. CCS, π-calculus and Ambients).

Reactions systems

- Semantics can be also specified by *reaction* (or "reduction") rules, which are pairs "(redex, reactum)". For instance:

$$(5+3, 8) \quad \text{written as} \quad 5+3 \longrightarrow 8$$
$$((\lambda x.M)N, M\{N/x\}) \quad \text{written as} \quad (\lambda x.M)N \longrightarrow M\{N/x\}$$

- A reaction system (RS) is specified by a set $\mathcal{R}$ of such rules, and possibly a family of *active contexts* where redexes have to be found in order to fire the rule.

$$\frac{(l, r) \in \mathcal{R}}{\mathcal{C}[l] \longrightarrow \mathcal{C}[r]}$$

- Only a silent, "internal" state chage.
- No interaction with the surrounding environment, thus no observation is specified.
- RS are much easier to state than LTS, but are not as useful!

Labelled Transition Systems from Reaction Systems?

Principle

What can be observed about a process P are its interactions with the surrounding environment.

Since a reaction system defines completely the behaviour of a system, it contains also the informations about interactions, although hidden.

Problem

Given a reaction system, is it possible to derive a "good" LTS?

By "good" we intend that

- the induced bisimulation must be a congruence
- labels should be not too many (otherwise it is difficult to use in practice)

Ad hoc solutions

Sometimes it can be done ad hoc, e.g, CCS: from reaction rule

$$a.P|\bar{a}.Q \longrightarrow P|Q$$

we *guess* the transitions

$$\alpha.P \xrightarrow{\alpha} P \qquad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{\bar{a}} Q'}{P|Q \xrightarrow{\tau} P'|Q'}$$

because we *recognize* labels as the (minimal) interaction with the surrounding contexts.
Ad hoc solutions are difficult, error prone and require lot of work and experience. (Cf. the plethora of LTS and bisimulations for π-calculus)

Aim

We look for a general, uniform way for deriving LTS from RS.

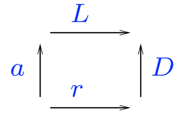
The "sledgehammer" approach

Define the labels of LTS as the contexts which may fire a rule

$$\frac{L(P) \longrightarrow Q}{P \xrightarrow{L} Q}$$

More formally:

$a \xrightarrow{L} a'$ means $L \circ a = D \circ r \rightarrow D \circ r' \simeq a'$ where (r, r') is a ground reaction rule



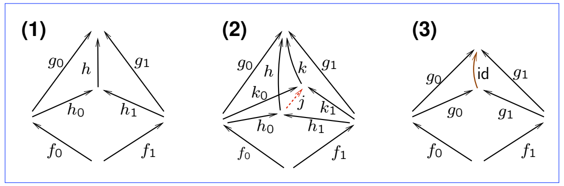
Proposition

The bisimulation induced by the contextual LTS is a congruence.

- But there are infinite labels for each process
- And also labels which do not carry any information about P, i.e., when the redex occurs in L and shares nothing with P.
- How to restrict the set of labels to only those really relevant? that is, "minimal" contexts?

Relative and Idempotent Pushouts (Leifer, Milner, 2000)

The "minimality" can be elegantly expressed as a universal categorical property.



Write $\vec{f}$ for f_0, f_1.

Call $\vec{g}$ a **bound** for $\vec{f}$ if $g_0 \circ f_0 = g_1 \circ f_1$.

(1) A **relative bound** $(\vec{h}, h)$ **for** $\vec{f}$ **to** $\vec{g}$.

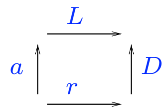
(2) A **relative pushout (RPO)** $(\vec{h}, h)$ **for** $\vec{f}$ **to** $\vec{g}$ **:** For any other relative bound $(\vec{k}, k)$, there is a unique mediator j.

(3) An **idem pushout (IPO)** $\vec{g}$ **for** $\vec{f}$ **:** $(\vec{g}, \mathrm{id})$ is an RPO for $\vec{f}$ to $\vec{g}$.

Labelled transition systems from IPOs

A **transition** $a \xrightarrow{L} a'$ is such that, for some r, r' and D:

- $(r, r' : J)$ is a ground reaction rule
- D is active
- (L, D) is an IPO for (a, r)
- $a' = D \circ r'$



- Remarkably, the bisimulation induced by IPO LTS is the same of contextual LTS.
- Notice that only contexts which form an IPO for the rule are considered as labels. Thus if the reaction takes place "outside" a, it means that the redex r appears in L and hence the square cannot be "minimal"

The Plan: Metamodels with RPOs

For reaching our Aim ("general methodologies for turning RS into LTS"), we need to find general *metamodels* with RPO and IPO constructions

- A category where RPO exist and can be calculated
- Conditions for establishing when a span $\vec{A}$ has IPOs, and how to calculate these IPOs
- Encoding metodologies, that is, how to represent calculi and systems (with reaction semantics) in these categories.

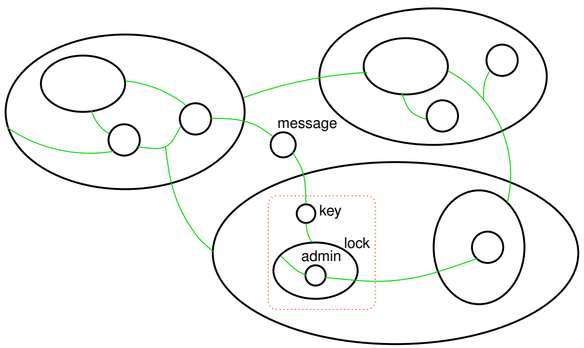
Then we obtain an "reduced" LTS (whose bisimulation is a congruence) automatically.

Bigraphical Models

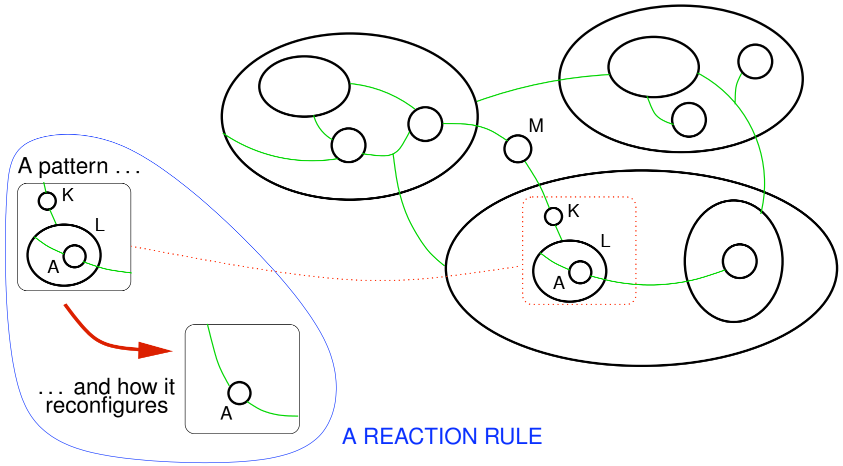
- Long term aim: "to express as much as possible of worldwide distributed computing in one mathematical model."
- Bigraphs (Milner 2001) aim to be a unifying model of computations based on communications and locality.
- Fundamental: they have RPO and IPO constructions
- References:
  - Pure bigraphs: structure and dynamics, R.Milner. (2005)
  - Bigraphs and mobile processes (revised), O.-H.Jensen and R.Milner. (2003)

Example

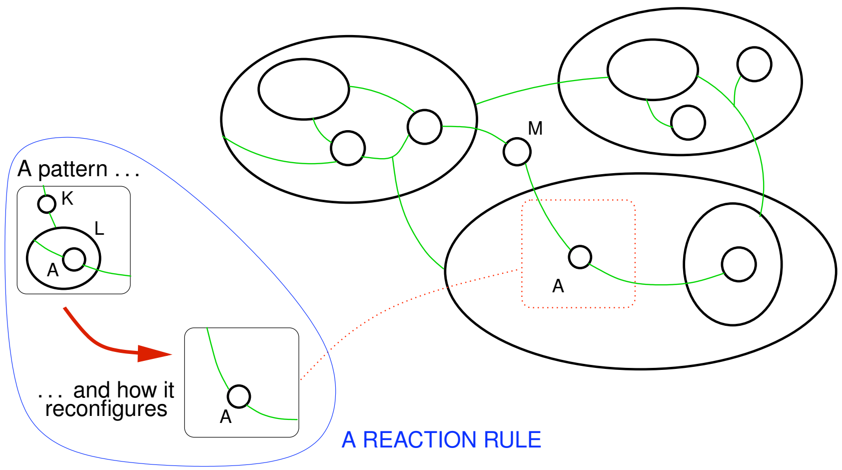
The ovals and circles are the nodes; the places, which are nested, are the interiors of nodes, while the links (the thin lines) connect ports that lie on the periphery of each node



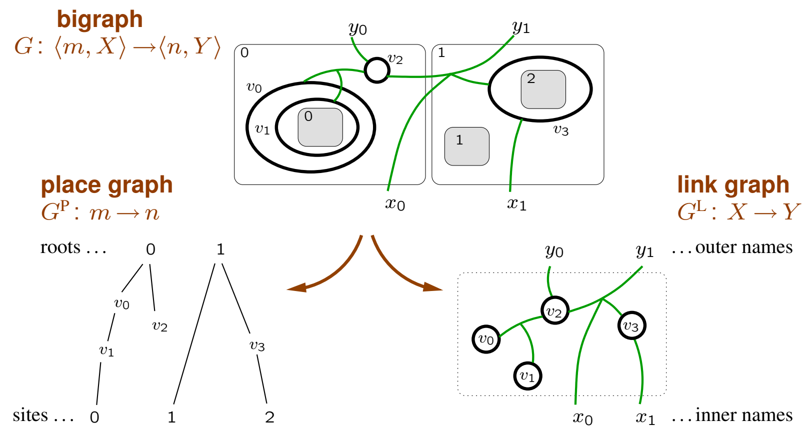
How a system evolves: a set of local reaction rules



Result of the reaction



A bigraph = a place graph + a link graph

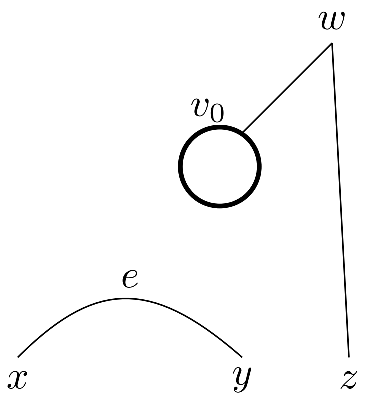


Output Linear Link Graph

'OLG

- An algorithm for the construction of RPOs;
- consistency conditions on the existence of bounds;
- an algorithm for the construction of IPOs.

[O. H. Jensen and R. Milner. Bigraphs and mobile processes (revised). Technical Report, University of Cambridge, 2004]

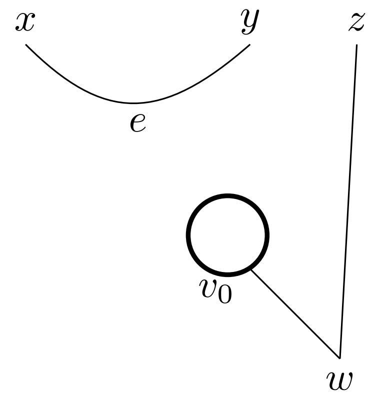


Input Linear Link Graph

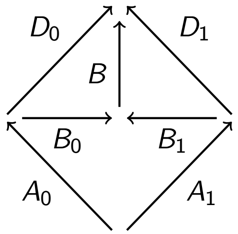
'ILG

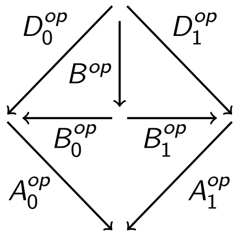
- An algorithm for the construction of (G)RPOs, as an instance of general construction for ILC(*PLGraphs*).

[P. Sobociński. Deriving process congruences from reaction rules. PhD thesis, University of Aarhus, 2004]



Duality





Proposition

$'\mathrm{OLG} \cong {'\mathrm{ILG}}^{op}$.

Corollary

Let $\vec{A}$ be a span in $'\mathrm{OLG}$, with a bound $\vec{D}$. $(\vec{B}, B)$ is an RPO for $(\vec{A}, \vec{D})$ in $'\mathrm{OLG}$ iff $(\vec{B}^{op}, B^{op})$ is an RPB for $(\vec{A}^{op}, \vec{D}^{op})$ in $'\mathrm{ILG}$.

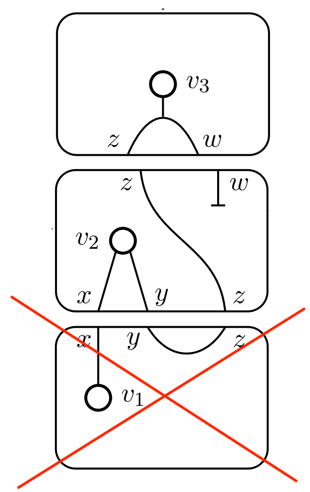
Limitations of Input or Output Linear Link Graphs

In Output Linear

- From outside to inside, links can fork, but cannot join.
- Two names of a component can be unified by the context;
- but two names in the context cannot be unified by a component.
- Once a name is created, it is known and unique in all subcomponents.

In Input Linear

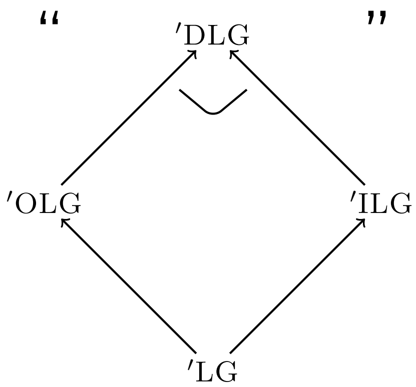
Vice versa.



Subsuming both input- and output-linear link graphs?

We look for a (pre)category $'\mathrm{DLG}$ (of link graphs), which satisfies the following conditions:

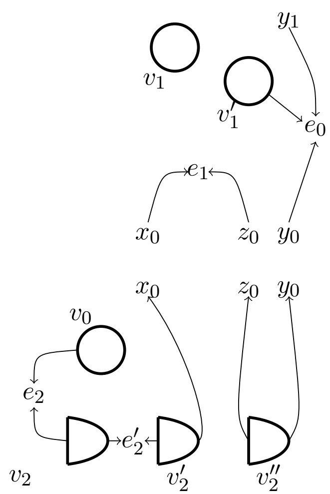
- $'\mathrm{ILG}$ and $'\mathrm{OLG}$ are two sub-precategories of $'\mathrm{DLG}$;
- $'\mathrm{DLG}$ is self-dual, that is $'\mathrm{DLG} = {'\mathrm{DLG}}^{op}$;
- there is a unique algorithm for the construction of RPO and RPB (hence of IPO and IPB).



Directed Link Graph I

Edges and Links

- edges become new resources;
- links have a direction from points (i.e. ports and names) to links (i.e. edges and names);
- direction represents the “flow of resource access”;
- names are “ports” through which resources are requested or offered;
- composition must respect the direction of requests.



Directed Link Graph II

Definition

A *polarized interface* X is a pair of sets of names $X = (X^-, X^+)$; the two components are called *downward* and *upward* interfaces, respectively.
A *directed link graph* $A : X \to Y$ is $A = (V, E, ctrl, link)$ where X and Y are the *inner* and *outer* interfaces, V is the set of *nodes*, E is the set of *edges*, $ctrl : V \to \mathcal{K}$ is the *control map*, and $link : \mathrm{Pnt}(A) \to \mathrm{Lnk}(A)$ is the *link map*, where the *ports*, the *points* and the *links* of A are defined as follows:

$$\mathrm{Prt}(A) \triangleq \sum_{v \in V} ar(ctrl(v)) \qquad \mathrm{Pnt}(A) \triangleq X^+ \uplus Y^- \uplus \mathrm{Prt}(A)$$

$$\mathrm{Lnk}(A) \triangleq X^- \uplus Y^+ \uplus E$$

Composition in 'DLG

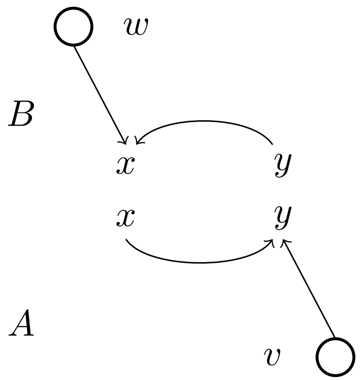
Given two directed link graphs $A_i = (V_i, E_i, ctrl_i, link_i) : X_i \to X_{i+1}$ $(i = 0, 1)$, the composition $A_1 \circ A_0 : X_0 \to X_2$ is defined as follows: $A_1 \circ A_0 \triangleq (V, E, ctrl, link)$, where $V \triangleq V_0 \uplus V_1$, $ctrl \triangleq ctrl_0 \uplus ctrl_1$, $E \triangleq E_0 \uplus E_1$ and $link : X_0^+ \uplus X_2^- \uplus P \to E \uplus X_0^- \uplus X_2^+$ is defined as follows (where $P = \mathrm{Prt}(A_0) \uplus \mathrm{Prt}(A_1)$):

$$link(p) \triangleq \begin{cases} link_0(p) & \text{if } p \in X_0^+ \uplus \mathrm{Prt}(A_0) \text{ and } link_0(p) \in E_0 \uplus X_0^- \\ link_1(x) & \text{if } p \in X_0^+ \uplus \mathrm{Prt}(A_0) \text{ and } link_0(p) = x \in X_1^+ \\ link_1(p) & \text{if } p \in X_2^- \uplus \mathrm{Prt}(A_1) \text{ and } link_1(p) \in E_1 \uplus X_2^+ \\ link_0(x) & \text{if } p \in X_2^- \uplus \mathrm{Prt}(A_1) \text{ and } link_1(p) = x \in X_1^- . \end{cases}$$

The identity link graph of X is $id_X \triangleq (\emptyset, \emptyset, \emptyset_{\mathcal{K}}, Id_{X^- \uplus X^+}) : X \to X$.

Avoiding loops (i.e. vacuous definitions)

We must forbid connections between names of the same interface in order to avoid undefined link maps after compositions. Hence, the link map cannot connect downward and upward names of the same interface, i.e., the following condition must hold:

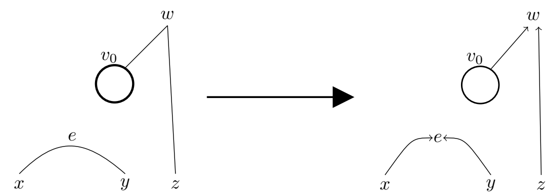


$$(link(X^+) \cap X^-) \cup (link(Y^-) \cap Y^+) = \emptyset$$

Embeddings

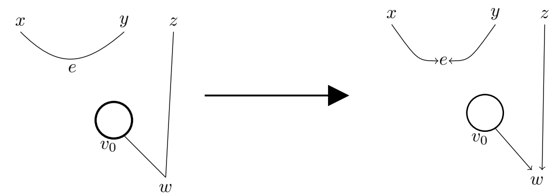
Output Linear Link Graphs

$\mathcal{F}_O$: 'OLG $\to$ 'DLG



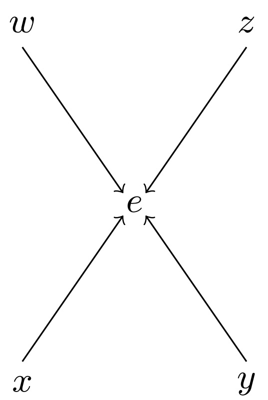
Input Linear Link Graphs

$\mathcal{F}_I$: 'ILG $\to$ 'DLG



New Link Graphs

There are directed link graphs which are neither input-linear nor output-linear, nor any combination of these.



$$C \triangleq (\emptyset, \{e\}, \emptyset, \{(x,e),(y,e),(z,e),(w,e)\}) : \{x,y\} \to \{z,w\}$$

RPO and RPB

RPO and RPB exist

In $'\mathrm{DLG}$ all RPO and RPB exist, and there is a unique method for constructing RPO's and RPB's.

Theorem

In $'\mathrm{DLG}$, whenever a span $\vec{A}$ of link graphs has a bound $\vec{D}$, there exists an RPO $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$.

Corollary

In $'\mathrm{DLG}$, whenever a co-span $\vec{D}$ of link graphs has a co-bound $\vec{A}$, there exists an RPB $(\vec{B}, B)$ for $\vec{A}$ to $\vec{D}$.

IPO and IPB

Consistency and IPOs

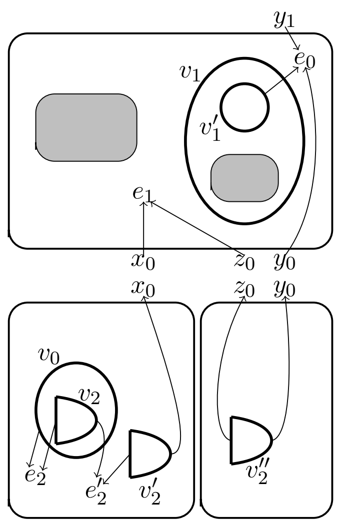
- There are consistency conditions for the existence of bounds and co-bounds;
- there is a unique algorithm to compute IPO and IPB.

Note

- These conditions and construction subsume those given by Jensen and Milner for output linear link graphs.
- We can derive consistency conditions and an algorithm to compute IPO in input linear link graphs.

Directed Bigraphs

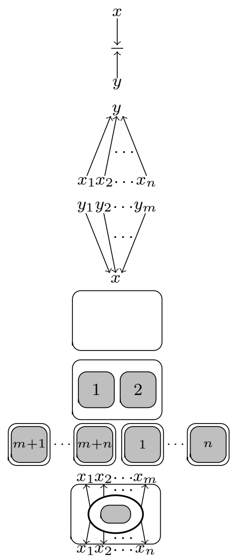
- The *directed bigraphs* can be defined as the composition of standard place graphs (i.e. Milner's one) and directed link graphs.
- An RPO (IPO) in $'\mathrm{DBig}$ is constructed by combining an RPO (IPO) in $'\mathrm{DLG}$ with an RPO (IPO) in $'\mathrm{PLG}$.



Abstract Directed Bigraphs

- In many situations we do not want to distinguish bigraphs differing only on the identity of nodes and edges;
- the category DBIG is constructed from 'DBIG forgetting the identity of nodes and edges and any idle edge;
- two directed bigraphs G and H are *lean-support equivalent*, written $G \asymp H$, if they are support equivalent after removing any idle edges.

Elementary Bigraphs



$\mathbf{I}^x_y : (\emptyset, y) \to (x, \emptyset)$	closure
$\triangle^y_X : (\emptyset, X) \to (\emptyset, y)$	substitution
$\nabla^Y_x : (x, \emptyset) \to (Y, \emptyset)$	fusion
$1 : \epsilon \to 1$	a barren root
$merge : 2 \to 1$	mapping 2 sites in 1 root
$\gamma_{m,n} : m + n \to n + m$	swapping m with n
$K^{\vec{x}^+}_{\vec{x}^-} : \langle(\vec{x}^-, \emptyset)\rangle \to \langle(\emptyset, \vec{x}^+)\rangle$	a discrete ion

Discrete Normal Form in DBIG

Proposition

In DBIG every bigraph G, discrete D, discrete and prime Q and discrete molecule N can be described by an expression of the respective following form:

$$G = (\omega \otimes id_n) \circ D \circ (\omega' \otimes id_m)$$
$$D = \alpha \otimes ((Q_0 \otimes \cdots \otimes Q_{n-1}) \circ (\pi \otimes id_{dom(\vec{Q})}))$$
$$Q = (merge_{n+p} \otimes id_{\emptyset,Y^+}) \circ (id_n \otimes N_0 \otimes \cdots \otimes N_{p-1}) \circ (\pi \otimes id_{Y^-,\emptyset})$$
$$N = (K^{\vec{x}^+}_{\vec{x}^-} \otimes id_{\emptyset,Y^+}) \circ Q.$$

Furthermore, the expression is unique up to isomorphisms on the parts.

The λ-calculus

Syntax

$$M, N ::= x \mid \lambda x.M \mid MN$$

Call-by-name Semantics

$$(\lambda x.M)N \to M[N/x]$$
$$\frac{M \to M'}{MN \to M'N}$$
$$\frac{N \to N'}{MN \to MN'}$$

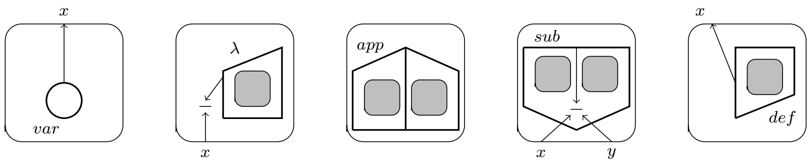
Call-by-value Semantics

$$(\lambda x.M)V \to M[V/x]$$
$$\frac{M \to M'}{MN \to M'N}$$
$$\frac{N \to N'}{MN \to MN'}$$

Values

A value is either a λ-abstraction or a variable.

Signature for the λ-calculus



var_x lam_x app $sub_{x,y}$ def_x

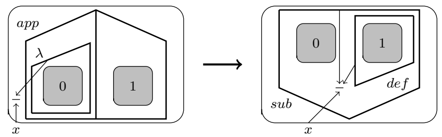
Single Substitutions

We give a signature for representing the λ-calculus "with single substitutions", that is where a substitution is performed once for each variable occurrence.

Translator operator

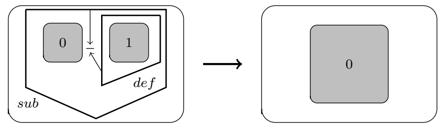
$$[\![x]\!] = var_x$$
$$[\![\lambda x.M]\!] = lam_x \circ ([\![M]\!] \curlywedge \Delta^x)$$
$$[\![MN]\!] = app \circ ([\![M]\!] \curlywedge [\![N]\!])$$

Call-by-name reactions



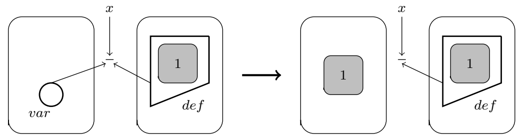
App_{cbn}

$app \circ (lam_x \curlywedge id_1) \to sub_{x,y} \circ (id_1 \curlywedge def_y)$



$Sub_{dispose}$

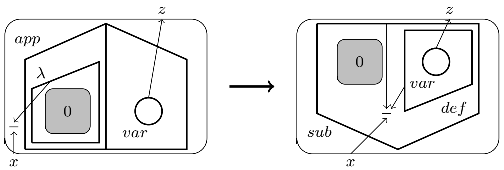
$sub_{x,y} \circ (id_1 \curlywedge \Delta^x \curlywedge def_y) \to id_1$



Sub_{var}

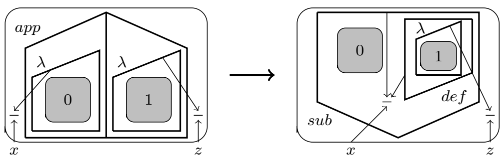
$\blacktriangledown^x_y \circ (var_y \curlywedge def_y) \to \blacktriangledown^x_y \circ (id_1 \curlywedge def_y)$

Call-by-value reactions



$App_{cbv\text{-}var}$

$app \circ (lam_x \curlywedge var_z) \to sub_{x,y} \circ (id_1 \curlywedge (def_y \circ var_z))$



$App_{cbv\text{-}lam}$

$app \circ (lam_x \curlywedge lam_z) \to sub_{x,y} \circ (id_1 \curlywedge (def_y \circ lam_z))$

Conclusions and Future Works

- We introduce directed bigraphs, a more general model that subsumes both input- and output-linear bigraphs;
- we present a unique algorithm to compute both RPO (IPO) and RPB (IPB);
- finally we have an algebra for directed bigraphs, based on a set of elementary bigraphs;
- we show an encoding of the λ-calculus in the directed bigraphs (without bindings).

- We want to derive a weak lts for the λ-calculus, using a construction defined by Jensen and compare the corresponding weak bisimilarity with known equivalences;
- we want to apply the model to fusion calculus and ν-calculus.