Lambda calculs et catégories

Paul-André Melliès

Master Parisien de Recherche en Informatique

Ecole Normale Supérieure

Synopsis of the lecture

1 – β-redexes and their residuals

2 – Finite Development Lemma

3 – An algebraic Church-Rosser theorem

4 – Event structures

5 – Asynchronous graphs

First part

β-redexes and their residuals

Rewriting in the λ-calculus

The pure λ-calculus

Terms $\quad M \quad ::= \quad x \quad | \quad App(M, N) \quad | \quad \lambda x.M$

The β-reduction:

$$App(\lambda x.M, N) \quad \longrightarrow \quad M[x := N]$$

The η-expansion:

$$M \quad \longrightarrow \quad \lambda x.App(M, x)$$

where x does not appear as free variable in M.

Remark: every term is considered up to renaming $\equiv_\alpha$ of the bound variables, typically:

$$\lambda x.\lambda y.x \quad \equiv_\alpha \quad \lambda z.\lambda y.z$$

Occurrences

The set of occurrences of a λ-term M is defined by induction:

$$
\begin{array}{lcl}
\mathbf{occ}(x) & = & \{\varepsilon\} \\
\mathbf{occ}(App(M, N)) & = & \{\varepsilon\} \uplus \{\mathsf{fun} \cdot o \mid o \in \mathbf{occ}(M)\} \uplus \{\mathsf{arg} \cdot o \mid o \in \mathbf{occ}(N)\} \\
\mathbf{occ}(\lambda x.M) & = & \{\varepsilon\} \uplus \{\mathsf{body} \cdot o \mid o \in \mathbf{occ}(M)\}
\end{array}
$$

Every occurrence $o \in \mathbf{occ}(M)$ of a λ-term M is labelled by

- an application node App
- a binder λx
- a variable x

Restriction

Every occurrence $o \in \mathbf{occ}(M)$ of a λ-term M defines a λ-term

$$M_{|o}$$

by structural induction on the occurrence:

$$M_{|\varepsilon} = M$$

$$App(M, N)_{|\mathsf{fun}\cdot o} = M_{|o} \qquad App(M, N)_{|\mathsf{arg}\cdot o} = N_{|o}$$

$$(\lambda x.M)_{|\mathsf{body}\cdot o} = M_{|o}$$

The λ-term $M_{|o}$ is the restriction of M along the occurrence o.

Concatenation of occurrences

Every occurrence

$$o \in \mathbf{occ}\,(M)$$

induces a concatenation function

$$\mathbf{occ}\,(M_{|o}) \quad \longrightarrow \quad \mathbf{occ}\,(M)$$

which transports every occurrence

$$o' \in \mathbf{occ}\,(M_{|o})$$

to the occurrence

$$o \cdot o' \in \mathbf{occ}\,(M)$$

β-redexes

Definition.

A β-redex is a pair

$$u \quad = \quad (M, o)$$

consisting of a λ-term M and of an occurrence $o \in \mathbf{occ}(M)$ such that the restriction

$$M_{|o} \quad = \quad App(\lambda x.P, Q)$$

is a *Beta*-rule pattern.

Notation: $\mathbf{Redex}(M)$ denotes the set of β-redexes of M.

β-redexes

Every β-redex

$$u \in \mathbf{Redex}(M)$$

induces a β-reduction

$$M = C[\,App(\,\lambda x.\,P\,,\,Q\,)\,] \quad \longrightarrow_\beta \quad C[\,P[\,x := Q\,]\,] = N$$

what we write

$$u \;:\; M \;\longrightarrow_\beta\; N$$

The graph of β-redexes

The graph

$$\mathbf{G}_\lambda = (V_\lambda, E_\lambda)$$

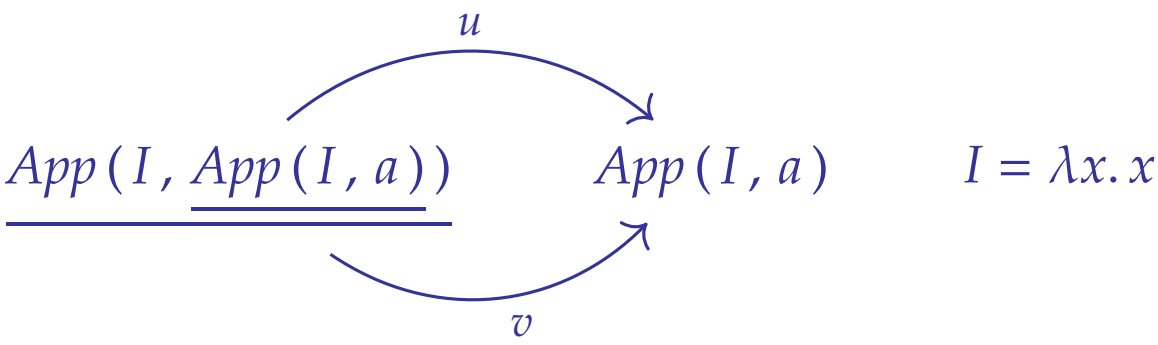
has λ-terms as vertices and β-redexes

$$u : M \longrightarrow_\beta N$$

as edges.

Illustration

There are two β-redexes u and v in



where

the β-redex u rewrites at occurrence ε
the β-redex v rewrites at occurrence $\arg \cdot \varepsilon$.

Church-Rosser theorem

Also called confluence theorem.

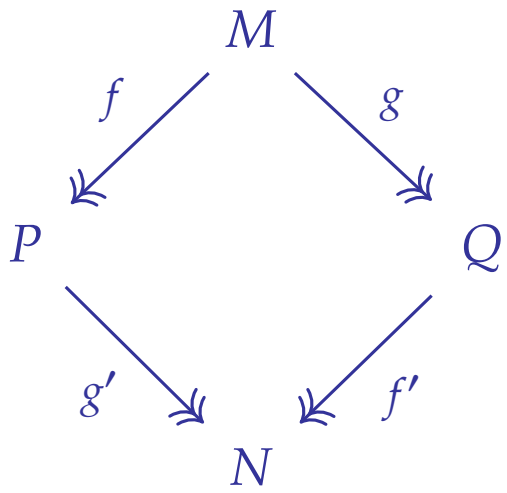
Given two β-rewriting paths

$$f : M \twoheadrightarrow P \qquad\qquad g : M \twoheadrightarrow Q$$

there exists a λ-term N and two β-rewriting paths

$$f' : Q \twoheadrightarrow N \qquad\qquad g' : P \twoheadrightarrow N$$

completing the span into a square diagram:



Residuals

Every β-redex

$$r \ : \ M \longrightarrow_\beta N$$

induces a binary relation

$$[r] \quad \subseteq \quad \mathbf{Redex}(M) \times \mathbf{Redex}(N)$$

which relates every β-redex

$$u \in \mathbf{Redex}(M)$$

to its **residuals**

$$v \in \mathbf{Redex}(N)$$

along the β-redex r.

Residuals

The residual relation

$$[r] \quad \subseteq \quad \mathbf{Redex}(M) \times \mathbf{Redex}(N)$$

is defined by induction on occurrence of $r = (M, o)$.

We will use the fact that every occurrence

$$o \in \mathbf{occ}\,(M)$$

induces a function

$$\mathbf{Redex}(M_{|o}) \quad \longrightarrow \quad \mathbf{Redex}(M)$$

defined as follows:

$$u = (\,M_{|o}\,,\,o_u\,) \quad \mapsto \quad o \cdot u = (\,M\,,\,o \cdot o_u\,)$$

Residuals

Base case. The β-redex

$$r \quad : \quad M = App\,(\,\lambda x.\,P\,,\,Q\,) \quad \longrightarrow_\beta \quad N = P\,[\,x := Q\,]$$

rewrites at the root occurrence ε. In that case:

$$u\,[r]\,v \qquad u = (M, o_u) \quad v = (N, o_v)$$

precisely when one of the two following cases occurs:

1. there exists an occurrence $o \in \mathbf{occ}\,(P)$
such that $o_u = \mathsf{fun} \cdot \mathsf{body} \cdot o$ and $o_v = o$

2. there exists an occurrence $o \in \mathbf{occ}\,(Q)$ and
an occurrence $o_x \in \mathbf{occ}\,(P)$ of a free variable x in P
such that $o_u = \mathsf{arg} \cdot o$ and $o_v = o_x \cdot o$.

Residuals

Abstraction case. The β-redex

$$r \quad : \quad M = \lambda x.\, P \quad \longrightarrow_\beta \quad N = \lambda x.\, Q$$

has occurrence $o_r = \mathsf{body} \cdot o$ and is thus of the form

$$r = \mathsf{body} \cdot r_P \qquad \text{for a } \beta\text{-redex } r_P \in \mathbf{Redex}(P)$$

In that case,

$$u\,[r]\,v$$

precisely when

$$u = \mathsf{body} \cdot u_P \qquad v = \mathsf{body} \cdot v_Q \qquad u_P\,[r_P]\,v_Q$$

for β-redexes $u_P \in \mathbf{Redex}(P)$ and $v_Q \in \mathbf{Redex}(Q)$.

Residuals

Application case A. The β-redex

$$r \quad : \quad M = App(P, Q) \quad \longrightarrow_\beta \quad N = App(P', Q)$$

has occurrence $o_r = \mathsf{fun} \cdot o$ and is thus of the form

$$r = \mathsf{fun} \cdot r_P \qquad \text{for a } \beta\text{-redex } r_P \in \mathbf{Redex}(P)$$

In that case,

$$u\,[r]\,v$$

precisely when one of the following cases occurs:

0. both β-redexes u and v have root occurrence ε

1. $u = \mathsf{fun} \cdot u_P \qquad v = \mathsf{fun} \cdot v_{P'} \qquad u_P\,[r_P]\,v_{P'}$
   for β-redexes $u_P \in \mathbf{Redex}(P)$ and $v_{P'} \in \mathbf{Redex}(P')$

2. $u = \mathsf{arg} \cdot w \qquad v = \mathsf{arg} \cdot w$
   for a β-redex $w \in \mathbf{Redex}(Q)$

Residuals

Application case B. The β-redex

$$r \quad : \quad M = App(P, Q) \quad \longrightarrow_\beta \quad N = App(P, Q')$$

has occurrence $o_r = \mathsf{arg} \cdot o$ and is thus of the form

$$r = \mathsf{arg} \cdot r_Q \qquad \text{for a } \beta\text{-redex } r_Q \in \mathbf{Redex}(Q)$$

In that case,

$$u\,[r]\,v$$

precisely when one of the following cases occurs:

0. both β-redexes u and v have root occurrence ε

1. $u = \mathsf{fun} \cdot w \qquad v = \mathsf{fun} \cdot w$
   for a β-redex $w \in \mathbf{Redex}(P)$

2. $u = \mathsf{arg} \cdot u_Q \qquad v = \mathsf{arg} \cdot v_{Q'} \qquad u_Q\,[r_Q]\,v_{Q'}$
   for β-redexes $u_Q \in \mathbf{Redex}(Q)$ and $v_{Q'} \in \mathbf{Redex}(Q')$

Residuals along a rewriting path

Every rewriting path

$$f \quad : \quad M \longrightarrow\!\!\!\!\!\rightarrow N$$

induces a residual relation

$$[f] \subseteq \mathbf{Redex}(M) \times \mathbf{Redex}(N)$$

defined by induction on the length of f:

1. when f is the empty path $id_M : M \twoheadrightarrow M$

$$u\,[id_M]\,v \iff u = v$$

2. when f factors as $f = r \cdot g$ where $r : M \longrightarrow_\beta P$ and $g : P \twoheadrightarrow N$,

$$u\,[r \cdot g]\,v \iff \exists\, w \in \mathbf{Redex}(P),\ u\,[r]\,w \text{ and } w\,[g]\,v$$

Second part

Finite development lemma

Rewriting parallel β-redexes in the λ-calculus

Refined λ-terms

Definition.

A refined λ-term is a pair

$$(M, \mathbf{U})$$

of a λ-term M and of a finite set $\mathbf{U}$ of β-redexes of M.

Remark.

Such a finite set $\mathbf{U}$ of β-redexes is often called a **multiredex**.
The idea is that one wishes to rewrite the elements of $\mathbf{U}$ in parallel...

The graph of refined λ-terms

Observation. Every β-redex

$$v \quad : \quad M \quad \longrightarrow \quad N$$

induces a β-redex of refined λ-term

$$(v, \mathbf{U}) \quad : \quad (M, \mathbf{U}) \quad \longrightarrow \quad (N, \mathbf{U}[v])$$

where $\mathbf{U}[v]$ denotes the set of β-redexes

$$\mathbf{U}[v] \quad = \quad \left\{ \quad u' \in \mathbf{Redex}(N) \quad | \quad \exists\, u \in \mathbf{U},\ u[v]u' \quad \right\}$$

Definition. The graph $\mathbf{G}_{\mathrm{ref}}$ of refined λ-terms has

- ▷ refined λ-terms as vertices
- ▷ refined β-redexes as edges

The graph of developments

Definition.

The graph of developments $\mathbf{G}_{\mathrm{dev}}$ is the graph $\mathbf{G}_{\mathrm{ref}}$ restricted to the refined β-redexes

$$(u, \mathbf{U}) \quad : \quad (M, \mathbf{U}) \quad \longrightarrow \quad (N, \mathbf{U}[u])$$

such that the β-redex u is an element of $\mathbf{U}$.

A development of $(M, \mathbf{U})$ is a path

$$f \quad : \quad (M, \mathbf{U}) \longrightarrow\!\!\!\!\!\rightarrow (N, \mathbf{V})$$

in the graph $\mathbf{G}_{\mathrm{dev}}$ of developments.

Finite development lemma

Key property.

There are no infinite path in the graph $\mathbf{G}_{\mathrm{dev}}$ of developments.

Proof : the idea is to associate an ordinal

$$\omega(M, \mathbf{U})$$

to every refined λ-term

$$(M, \mathbf{U})$$

in such a way that for every refined β-redex

$$(M, \mathbf{U}) \quad \longrightarrow_\beta \quad (N, \mathbf{V})$$

the associated ordinals strictly decrease:

$$\omega(M, \mathbf{U}) \quad > \quad \omega(N, \mathbf{V})$$

The multiset ordering

Suppose given a partially ordered set $(S, \leq_S)$,

Two finite multisets

$$M, N \quad : \quad S \quad \longrightarrow \quad \mathbb{N}$$

are ordered by the multiset ordering

$$M >_{\mathsf{mset}} N$$

precisely when there exists two finite multisets X, Y such that

$$N \quad = \quad (M \setminus X) \uplus Y$$

and

$$\forall y \in Y, \exists x \in X, \quad x >_S y.$$

Nesting and gripping

The difficulty is that the number of β-redexes in $\mathbf{U}$ may increase in the course of development, typically when a β-redex

$$u \quad : \quad App\,(\,\lambda x.\,App\,(\,x\,,\,x\,)\,,\,P\,) \quad \longrightarrow_\beta \quad App\,(\,P\,,\,P\,)$$

lies in $\mathbf{U}$ and duplicates the β-redexes of $\mathbf{U}$ in the argument P.

Nesting and gripping

In order to establish the finite development lemma,
one thus needs to understand the structure of β-redexes in λ-terms...

To that purpose, one introduces the two relations of

nesting and **gripping**

between β-redexes of a same λ-term.

Nesting and gripping

Suppose given two β-redexes

$$u, v \in \mathbf{Redex}(M)$$

of the same λ-term, with respective occurrences o_u and o_v.

One writes

$$M_{|o_u} = App(\lambda x.P, Q) \quad \longrightarrow_\beta \quad P[x := Q]$$

and

$$M_{|o_v} = App(\lambda y.R, S) \quad \longrightarrow_\beta \quad R[y := S]$$

the β-reductions corresponding to the β-redexes u and v.

Nesting

One declares that u nests v and writes

$$u \prec_M v$$

when

$$o_v = o_u \cdot \mathsf{arg} \cdot o$$

for some occurrence $o \in \mathbf{occ}(Q)$ of the argument of u.

Gripping

Similarly, one declares that u grips v and writes

$$u \lessdot_M v$$

when

$$o_v = o_u \cdot \mathsf{fun} \cdot o$$

for some occurrence $o \in \mathbf{occ}(P)$ of the body of u
and an occurrence of the variable x bound by the β-redex u
appears in the argument S of the β-redex v.

Nesting and gripping

In other words,

$$u \prec_M v$$

when the β-redex v appears in the argument of u,
and

$$u \prec\prec_M v$$

when

1. the β-redex v appears in the body of u
2. an occurrence of variable x bound by the β-redex u appears in the argument of v.

Key properties of nesting and gripping

Property 1. Consider a β-redex

$$r \quad : \quad M \quad \longrightarrow_\beta \quad N$$

and β-redexes $u, v \in \mathbf{Redex}(M)$ and $u', v' \in \mathbf{Redex}(N)$ such that

$$u \,[r]\, u' \qquad \text{and} \qquad v \,[r]\, v'$$

If

$$u' \prec_N v'$$

then either

$$u \prec_M v$$

or

$$r \lessdot_M u \quad \text{and} \quad r \prec_M v.$$

Key properties of nesting and gripping

Property 2. Consider a β-redex

$$r \quad : \quad M \quad \longrightarrow_\beta \quad N$$

and β-redexes $u, v \in \mathbf{Redex}(M)$ and $u', v' \in \mathbf{Redex}(N)$ such that

$$u \,[r]\, u' \qquad \text{and} \qquad v \,[r]\, v'$$

If

$$u' \lessdot_N v'$$

then

$$u \lessdot_M v$$

or

$$u \lessdot_M r \lessdot_M v.$$

Key properties of nesting and gripping

Property 3. The nesting relation $\prec_M$ is transitive.

Property 4. The gripping relation $\ll_M$ does not contain any loop

$$u_1 \ll_M u_2 \ll_M \cdots \ll_M u_n = u_1$$

Gripping height

Definition. The gripping height

$$|u|_{\mathbf{U}} \quad \in \quad \mathbb{N}$$

of a β-redex $u \in \mathbf{U}$ in a refined λ-term

$$(M, \mathbf{U})$$

is the length $n \in \mathbb{N}$ of the longest sequence of grippings:

$$u = u_1 \lessdot_M u_2 \lessdot_M \cdots \lessdot_M u_n$$

where the β-redexes $u_1, \ldots, u_n$ are all elements of $\mathbf{U}$.

Preliminary observation

Property. Consider a refined β-redex

$$(r, \mathbf{U}) \quad : \quad (M, \mathbf{U}) \quad \longrightarrow_\beta \quad (N, \mathbf{V})$$

where $r \in \mathbf{U}$ and $\mathbf{V} = \mathbf{U}[r]$.

In that case, for all β-redexes $u \in \mathbf{U}$ and $v \in \mathbf{V}$,

$$u[r]\,v \quad \Rightarrow \quad |v|_{\mathbf{V}} \leq |u|_{\mathbf{U}}$$

Preliminary observation

Sketch of the proof. Every sequence of gripping

$$v = v_1 \lll_N v_2 \lll_N \cdots \lll_N v_n$$

with all v_i's in $\mathbf{V}$ induces a sequence of gripping of the form

$$u = u_1 \lll_M \cdots \lll_M u_n$$

or of the form

$$u = u_1 \lll_M \cdots \lll_M u_k \lll_M r \lll_M u_{k+1} \lll_M \cdots \lll_M u_n$$

with all u_i's in $\mathbf{U}$. Note that moreover

$$u_i \, [r] \, v_i$$

for all indices $1 \leq i \leq n$.

Nesting depth

Definition. The nesting depth

$$\| u \|_{\mathbf{U}}$$

of a β-redex

$$u \in \mathbf{Redex}(M)$$

is the multiset

$$\| u \|_{\mathbf{U}} \quad = \quad \left\langle \quad | v |_{\mathbf{U}} \quad \middle| \quad v \prec_M u \quad \right\rangle$$

of gripping heights of the β-redexes $v \in \mathbf{Redex}(M)$ nesting u.

Second observation

Key property. Consider a refined β-redex

$$(r, \mathbf{U}) \quad : \quad (M, \mathbf{U}) \quad \longrightarrow_\beta \quad (N, \mathbf{V})$$

where $r \in \mathbf{U}$ and $\mathbf{V} = \mathbf{U}[r]$.

In that case, for all β-redexes $u \in \mathbf{U}$ and $v \in \mathbf{V}$,

$$u[r]v \quad \Rightarrow \quad \|v\|_{\mathbf{V}} \leq \|u\|_{\mathbf{U}}$$

Moreover, the inequality

$$\|v\|_{\mathbf{V}} \leq \|u\|_{\mathbf{U}}$$

is strict when $r \prec_M u$.

Second observation

Proof by case analysis.

▷ suppose that the β-redex r does not nest u.

In that case, the residual relation

$$[r] \quad \subseteq \quad \mathbf{U} \times \mathbf{V}$$

defines a one-to-one relationship between the two sets

$$\left\{ \quad x \in \mathbf{U} \quad \middle| \quad x \prec_M u \quad \right\} \qquad \text{and} \qquad \left\{ \quad y \in \mathbf{V} \quad \middle| \quad y \prec_N v \quad \right\}$$

From this follows that

$$\| v \|_{\mathbf{V}} \leq \| u \|_{\mathbf{U}}$$

because the gripping height of each $x \prec_M u$ is larger than the gripping height of its unique residual $y \prec_N v$.

Second observation

Proof by case analysis.

▷ suppose that the β-redex r does nest u.

In that case, **Prop. 2** implies that every β-redex in the set

$$\left\{ \quad y \in \mathbf{V} \quad \middle| \quad y \prec_N v \quad \right\}$$

is either **(case a.)** the residual of a β-redex in the set

$$\left\{ \quad x \in \mathbf{U} \quad \middle| \quad x \prec_M u \quad \right\}$$

or **(case b.)** the residual of a β-redex $x \in \mathbf{U}$ whose gripping height

$$|x|_{\mathbf{U}}$$

is strictly smaller than the gripping height of r because

$$r \lll_M x.$$

Second observation

From this follows that

$$\|v\|_{\mathbf{V}} < \|u\|_{\mathbf{U}}.$$

because the residual relation

$$[r] \quad \subseteq \quad \mathbf{U} \times \mathbf{V}$$

defines a one-to-one relationship between the two sets

$$\left\{ \quad x \in \mathbf{U} \quad \middle| \quad x <_M u \quad \right\} \qquad \text{and} \qquad \left\{ \quad y \in \mathbf{V} \quad \middle| \quad y <_N v \quad \right\}$$

restricted to the β-redexes x and y related by the residual relation ; while the other β-redexes appearing in

$$\left\{ \quad y \in \mathbf{V} \quad \middle| \quad y <_N v \quad \right\}$$

have a gripping height strictly smaller than the gripping height of

$$r \quad \in \quad \left\{ \quad x \in \mathbf{U} \quad \middle| \quad x <_M u \quad \right\}$$

which does not have any residual along itself.

Finite development lemma

Lemma.

There are no infinite developments

$$(M_1, \mathbf{U}_1) \xrightarrow{u_1} (M_2, \mathbf{U}_2) \xrightarrow{u_2} (M_3, \mathbf{U}_3) \xrightarrow{u_3} \cdots$$

in the graph $\mathbf{G}_{\mathrm{dev}}$ of developments.

Proof. The multiset of nesting depths

$$\omega(\mathbf{U}_i) \quad = \quad \left\langle \quad \|u\| \quad | \quad u \in \mathbf{U}_i \quad \right\rangle$$

strictly decreases at each step of the development.

Development

Definition. A development of a finite set

$$\mathbf{U} \subseteq \mathbf{Redex}(M)$$

of β-redexes in M is defined as a path

$$f \quad : \quad (M, \mathbf{U}) \longrightarrow\!\!\!\!\!\rightarrow (N, \varnothing)$$

in the refinement graph $\mathbf{G}_{\mathrm{ref}}$.

Note that the development may be equivalently seen as a path

$$f \quad : \quad M \longrightarrow\!\!\!\!\!\rightarrow N$$

in the underlying rewriting graph $\mathbf{G}_{\lambda}$.

We write in that case

$$f \propto (M, \mathbf{U})$$

Local confluence with residuals

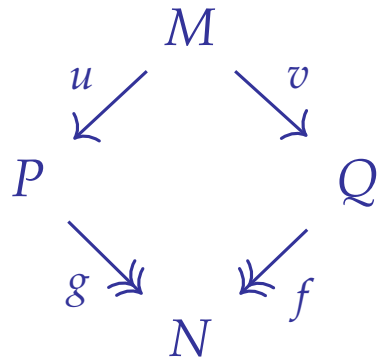
Additional property. For every coinitial pair of β-redexes

$$u : M \longrightarrow_\beta P \qquad v : M \longrightarrow_\beta Q$$

there exists a λ-term N and two developments

$$f \propto u\,[v] \qquad g \propto v\,[u]$$

completing the span into a square diagram:



and such that the residual relations along the two borders coincide:

$$[u \cdot g] \;=\; [v \cdot f] \quad\subseteq\quad \mathbf{Redex}(M) \times \mathbf{Redex}(N)$$

Finite development theorem

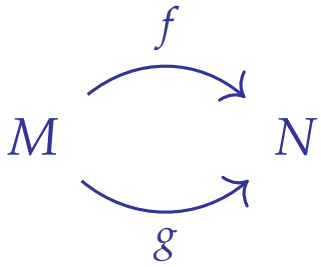
Theorem. Every two developments

$$f, g \propto \mathbf{U}$$

of the same finite set of β-redexes

$$\mathbf{U} \quad \subseteq \quad \mathbf{Redex}(M)$$

are coinitial and cofinal

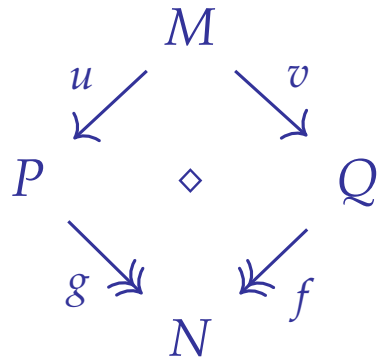


and induce the same residual relation

$$[f] \;=\; [g] \quad \subseteq \quad \mathbf{Redex}(M) \times \mathbf{Redex}(N).$$

Permutation tiles

Definition. A permutation tile $u \cdot f \diamond v \cdot g$



is a pair of developments $u \cdot f$ and $v \cdot g$ where

$$u : M \longrightarrow_\beta P \qquad v : M \longrightarrow_\beta Q$$

are two coinitial β-redexes and

$$f \propto u\,[v] \qquad g \propto v\,[u]$$

are developments of the respective residuals of u and v.

Finite development theorem

Stronger version. Every two developments

$$f, g \propto \mathbf{U}$$

of the same finite set of β-redexes

$$\mathbf{U} \quad \subseteq \quad \mathbf{Redex}(M)$$

are equivalent modulo a series of permutation tiles.

Third part

An algebraic Church-Rosser theorem

A pushout at the heart of Rewriting Theory

Pointed graph

Definition. A graph $\mathbf{G}$

$$E \quad \underset{\text{target}}{\overset{\text{source}}{\rightrightarrows}} \quad V$$

equipped with a specific edge

$$\varnothing_M : M \longrightarrow M$$

for every vertex

$$M \in V.$$

Concurrent graphs

Definition. A pointed graph $\mathbf{G}$

$$E \quad \underset{\text{target}}{\overset{\text{source}}{\rightrightarrows}} \quad V$$

equipped with a set of permutation tiles

$$f \diamond g \quad : \quad P \longrightarrow\!\!\!\!\!\rightarrow Q$$

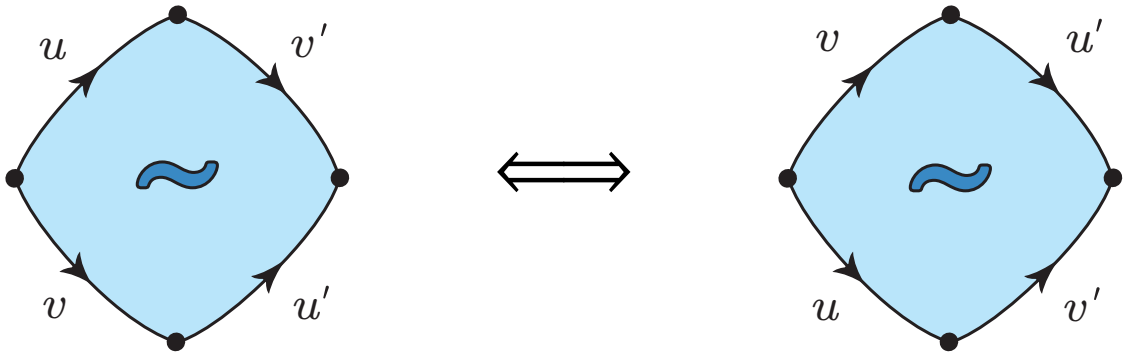
between coinitial and cofinal paths of length 2, satisfying the following axioms or properties.

1. Reversibility

Every tile is reversible:

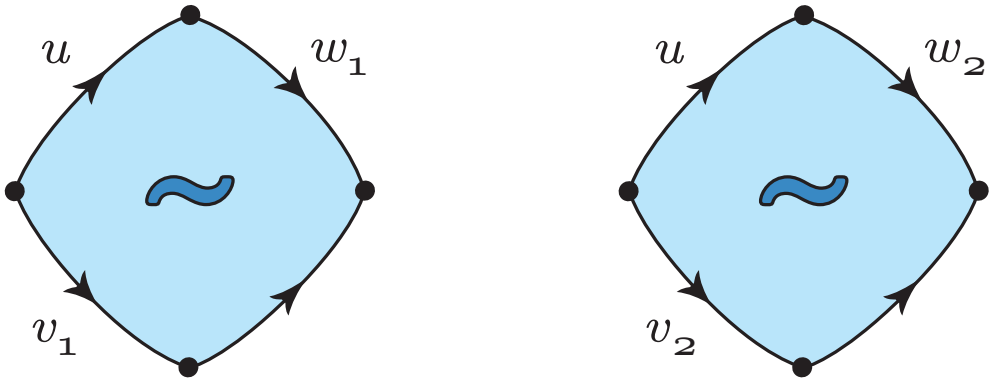
$$f \diamond g \quad \Rightarrow \quad g \diamond f$$

This property can be represented in this way:



2. Structure axiom

Suppose that one has two reversible tiles of the form



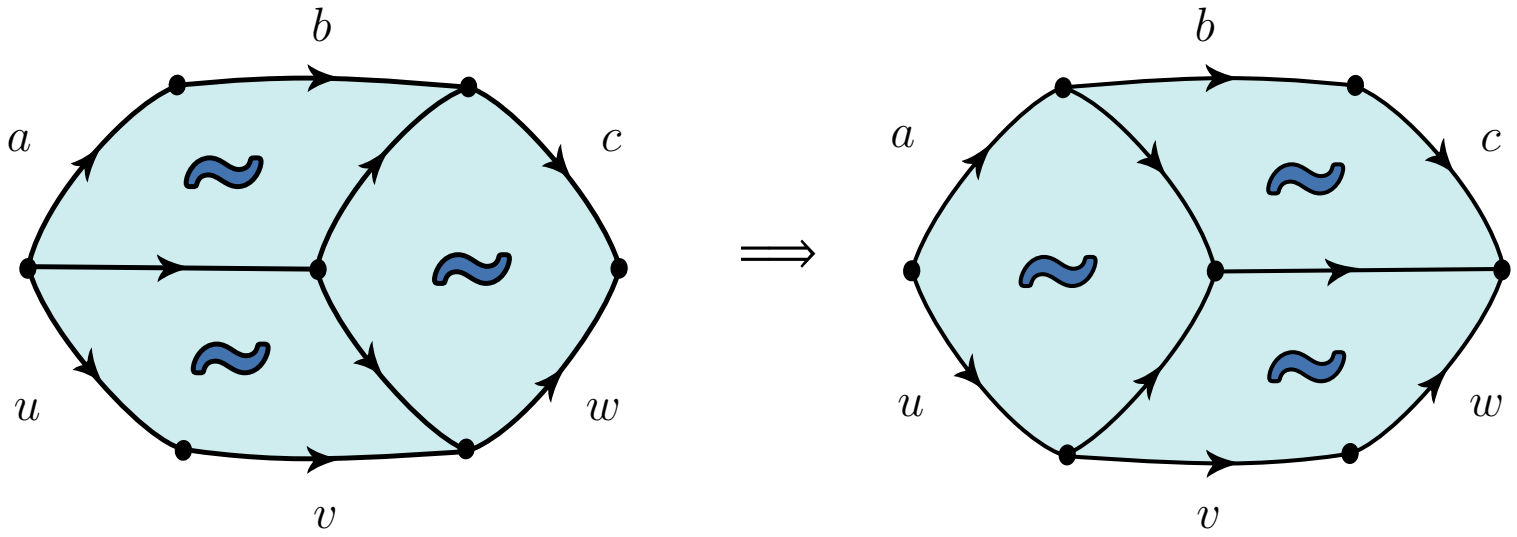
in the concurrent graph **G**.

In that case, one requires that

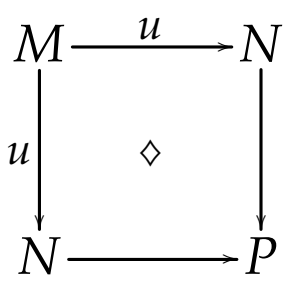
$$v_1 = v_2 \quad \Rightarrow \quad w_1 = w_2.$$

3. Forward cube axiom

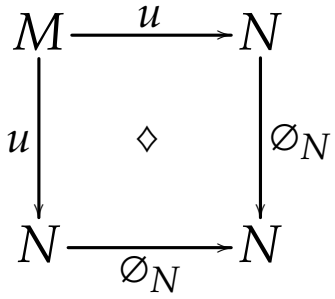
The asynchronous graph $\mathbf{G}$ satisfies the forward cube property:

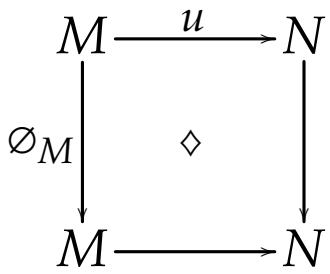


4. Two pointed axioms

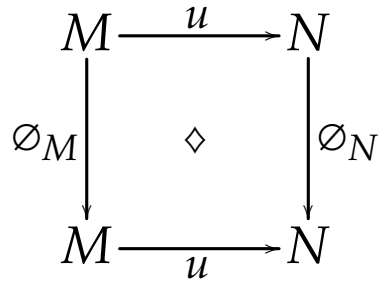


implies





implies



The rewriting category

Every concurrent graph

$$(\mathbf{G}, \diamond)$$

induces a category

$$\mathbf{Cat}(\mathbf{G}, \diamond)$$

defined in the following way:

- its objects are the vertices of $\mathbf{G}$,
- its morphisms are the finite paths of $\mathbf{G}$ modulo
  a. the permutation tiles $f \diamond g$
  b. the equations $\varnothing_M = id_M$

Confluence graphs

Definition. A confluence graph $(\mathbf{G}, \diamond)$ is a concurrent graph where for every pair of edges

$$u : M \to P \qquad v : M \to Q$$

there exists a permutation tile of the form

$$u \cdot v' \diamond v \cdot u'$$

for a vertex N and edges u', v' of the form:

$$u' : Q \to N \qquad v' : P \to N$$

Residual of a path after an edge

In a confluence graph $(\mathbf{G}, \diamond)$, every edge

$$u : M \longrightarrow N$$

defines a partial function

$$f \mapsto f/u$$

from M-paths to N-paths.

The path f/u is defined by structural induction on f

$$id_M/u = id_N$$

$$(v \cdot g)/u = (v/u) \cdot g/(u/v)$$

Residual of a path after a path

In a confluence graph $(\mathbf{G}, \diamond)$, every path

$$f : M \longrightarrow N$$

defines a partial function

$$h \mapsto h/f$$

from M-paths to N-paths.

The path h/f is defined by structural induction on f

$$h/id_M = h$$

$$h/(v \cdot g) = (h/v)/g$$

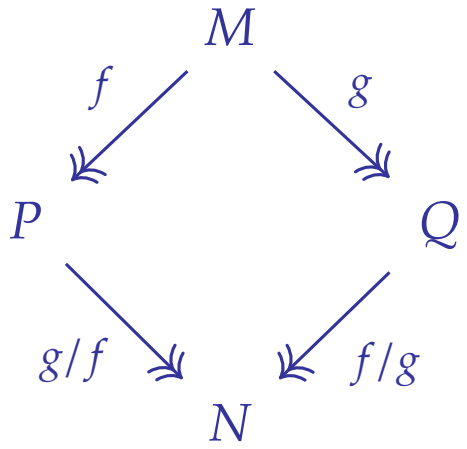
An algebraic Church-Rosser theorem

Main theorem. The rewriting category

$$\mathbf{Cat}(\mathbf{G}, \diamond)$$

associated to a confluence graph $(\mathbf{G}, \diamond)$ has pushouts.

Idea of the proof: every diagram of the form



defines a pushout diagram in the rewriting category $\mathbf{Cat}(\mathbf{G}, \diamond)$.

The confluence graph of the λ-calculus

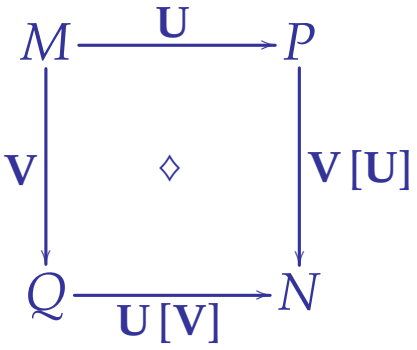
The vertices of $\mathbf{G_{MRed}}$ are the λ-terms and its edges

$$M \xrightarrow{\mathbf{U}} N$$

are the complete developments of a multiredex

$$\mathbf{U} \quad \subseteq \quad \mathbf{Redex}(M)$$

The permutation tiles $\diamond$ are of the form



An algebraic Church-Rosser theorem

Main theorem for the λ-calculus.

The category $\mathscr{C}_\lambda$ with

- λ-terms as objects
- rewriting paths modulo permutation tiles as morphisms

is a category with pushouts.

Fourth part

Event structures

An asynchronous model of computations

Event structures

Definition.

A prime event structure is a triple

$$(E, \leq, \#)$$

consisting of

- a partially ordered set $(E, \leq)$ of events
- a symmetric and irreflexive binary relation $\#$ between events.

Event structures

An event structure is required to satisfy the two conditions:

Finiteness condition : the set of events

$$\downarrow e \quad := \quad \left\{ \quad f \in E \quad | \quad f \leq e \quad \right\}$$

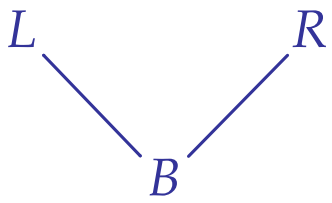
is finite for every event $e \in E$

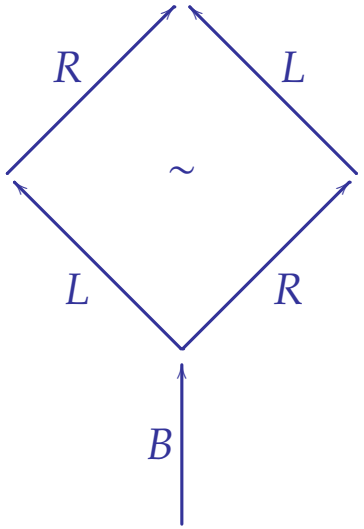
Inheritance condition:

$$\forall e_1, e_2, e_3 \in E, \qquad e_1 \# e_2 \quad \text{and} \quad e_2 \leq e_3 \quad \Rightarrow \quad e_1 \# e_3.$$

True concurrency

Event structure $\mapsto$ Transition system

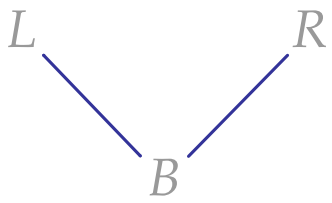


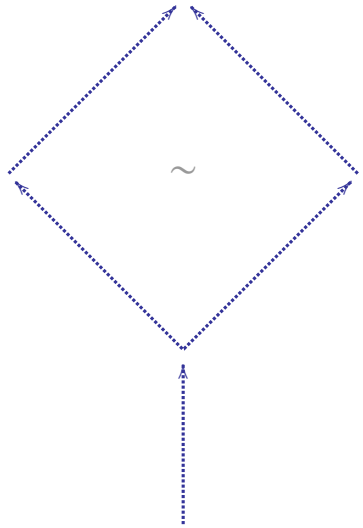


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

True concurrency

Event structure $\mapsto$ Transition system

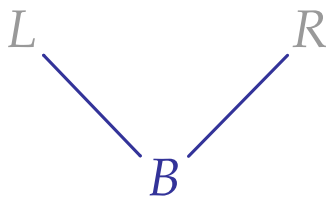


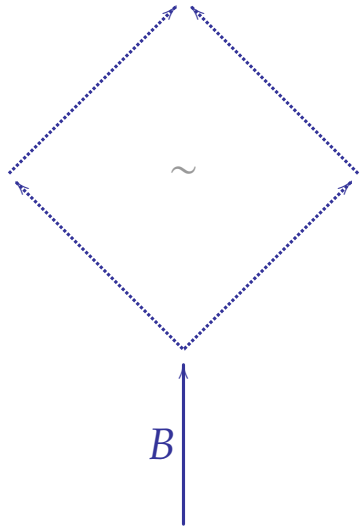


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

True concurrency

Event structure $\mapsto$ Transition system

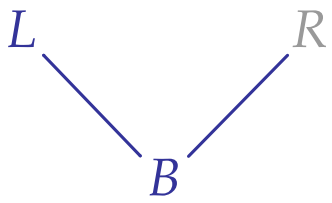


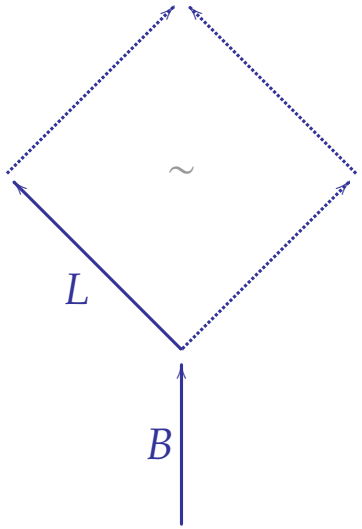


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

True concurrency

Event structure $\mapsto$ Transition system

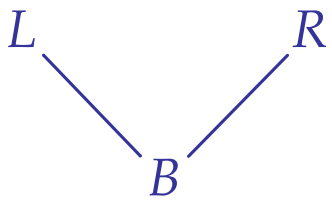


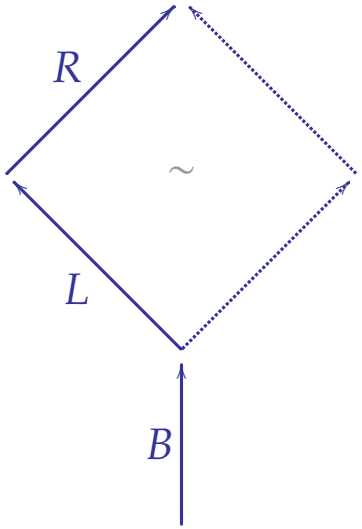


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

True concurrency

Event structure $\mapsto$ Transition system

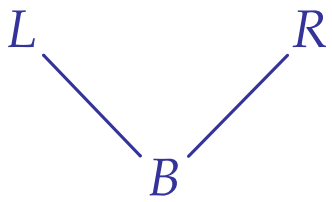


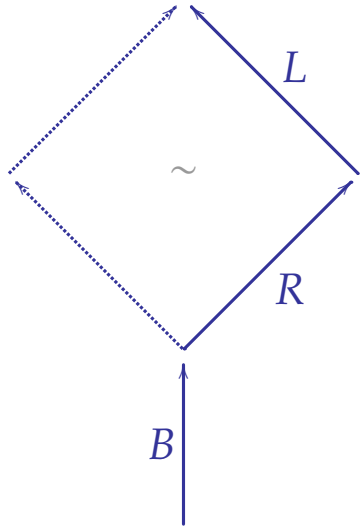


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

True concurrency

Event structure $\mapsto$ Transition system

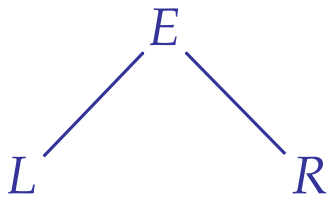


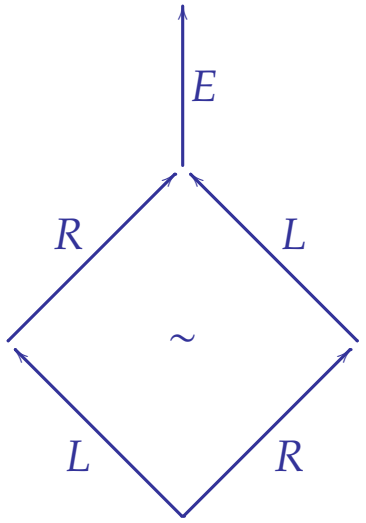


B	=	Buy a pair of shoes
L	=	Put left shoe on
R	=	Put right shoe on

Synchronization

Event structure $\mapsto$ Transition system

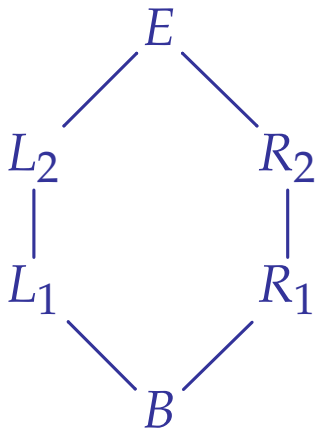


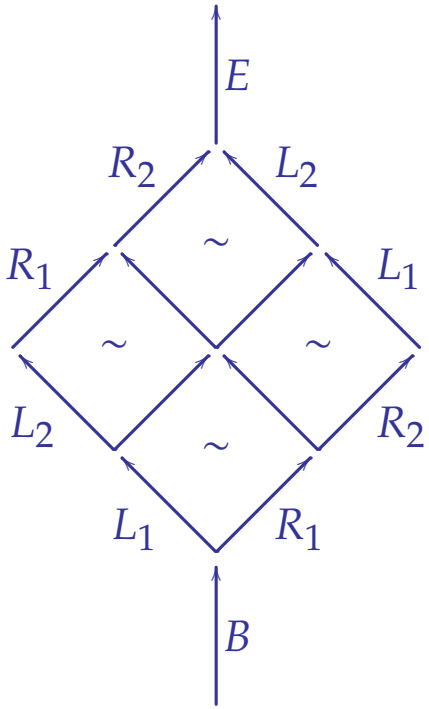


L	=	Lace left shoe.
R	=	Lace right shoe.
E	=	Exit the store.

The extended shoe shop experience

Event structure $\mapsto$ Transition system

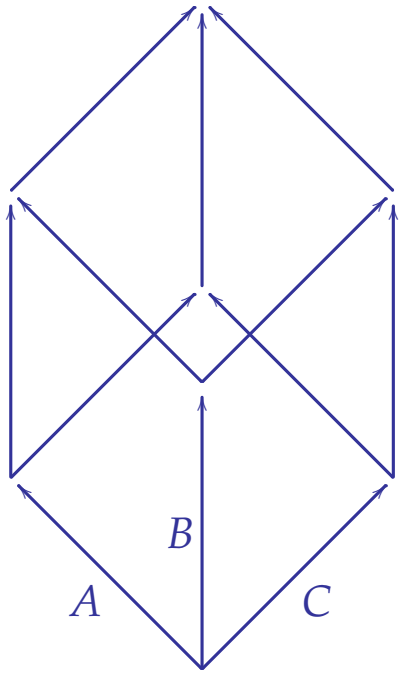




The n-dimensional structure

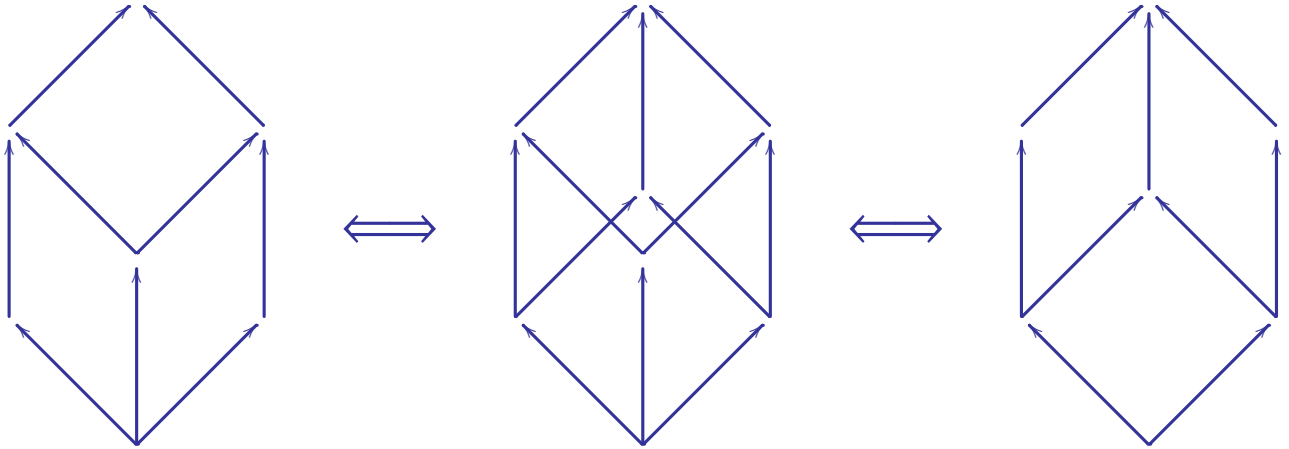
Event structure $\mapsto$ Transition system

$A \quad B \quad C$



The 3-dimensional cube property

Property. every transition system generated by an order satisfies:



Configurations

Definition. A **configuration** is a set of events

$$X \quad \subseteq \quad E$$

downward-closed:

$$\forall e \in X, \qquad e' \leq e \quad \Longrightarrow \; e' \in X$$

and whose events are pairwise compatible:

$$\forall e_1, e_2 \in X, \qquad e_1 \uparrow e_2$$

where $e_1 \uparrow e_2$ means that the two events are compatible: $\neg(e_1 \# e_2)$.

Asynchronous graphs

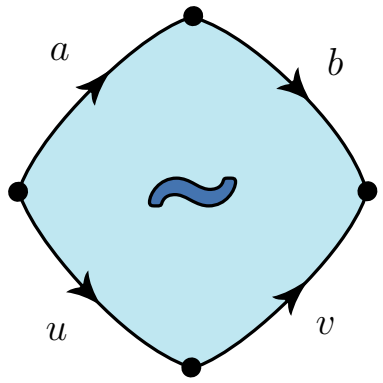
Definition. A graph

$$\mathbf{G} \quad = \quad (V, E, \partial_0, \partial_1)$$

defined by its source and target functions

$$\partial_0 \, , \, \partial_1 \quad : \quad E \quad \longrightarrow \quad V$$

together with a set of 2-dimensional tiles of the form



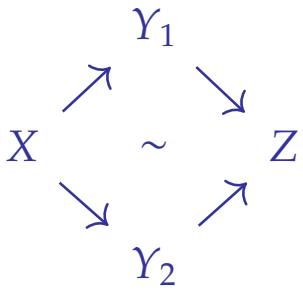
The configuration space of an event structure

Every event structure $(E, \leq, \#)$ generates an asynchronous graph

- ▷ whose vertices are the finite configurations $(E, \leq, \#)$,
- ▷ whose edges $X \to Y$ are the triples

$$(\quad X \quad , \quad e \quad , \quad Y = X + \{e\} \quad)$$

where e is an event in Y not appearing in X,

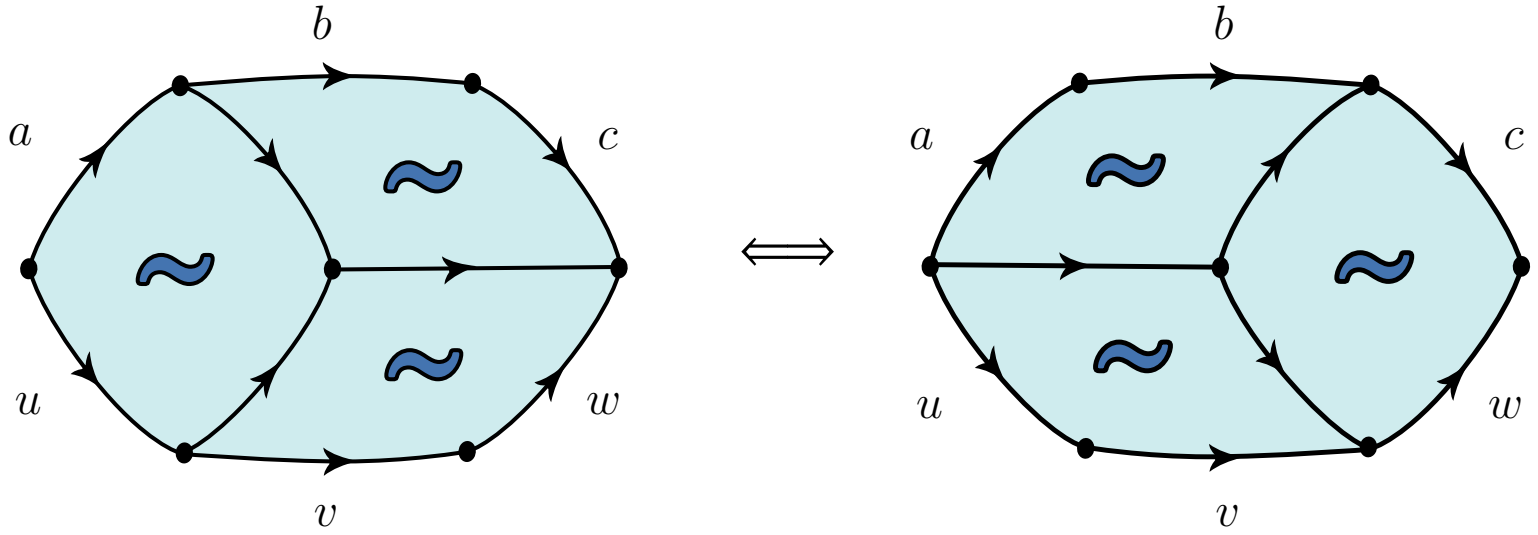
- ▷ whose tiles  are the quadruples

$$(\quad X \quad , \quad e_1 \quad , \quad e_2 \quad , \quad Z = X + \{e_1, e_2\} \quad)$$

where e_1 and e_2 are two **different** events in Z not appearing in X.

The cube property

The resulting asynchronous graph satisfies the cube property:



Conversely...

Fifth part

Asynchronous graphs

A 2-dimensional account of rewriting

Asynchronous graphs

Definition. An asynchronous graph $\mathbf{G}$ is a graph

$$E \quad \underset{\text{target}}{\overset{\text{source}}{\rightrightarrows}} \quad V$$

equipped with a set of permutation tiles

$$f \diamond g \quad : \quad P \longrightarrow\!\!\!\!\!\rightarrow Q$$

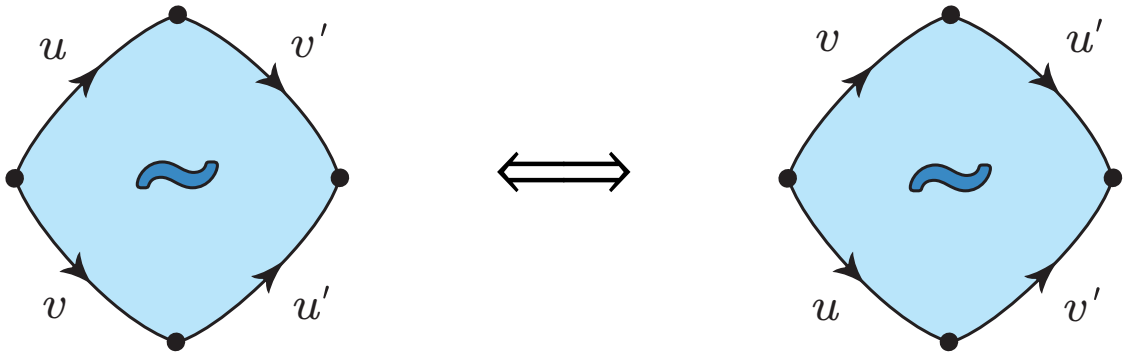
between coinitial and cofinal paths of length 2, satisfying the following properties.

Reversibility

Every tile is reversible:

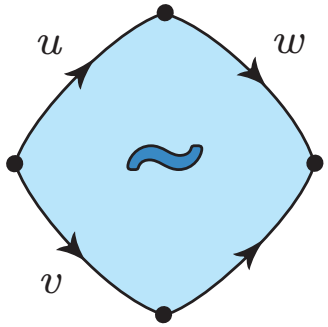
$$f \diamond g \quad \Rightarrow \quad g \diamond f$$

This property can be represented in this way:



Structure axiom 1

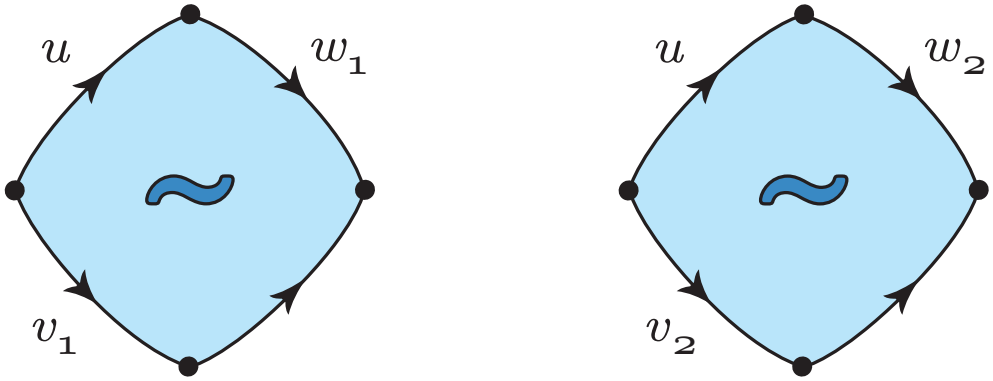
In every reversible tile



the two redexes u and v are required to be different.

Structure axiom 2

Suppose that one has two reversible tiles of the form



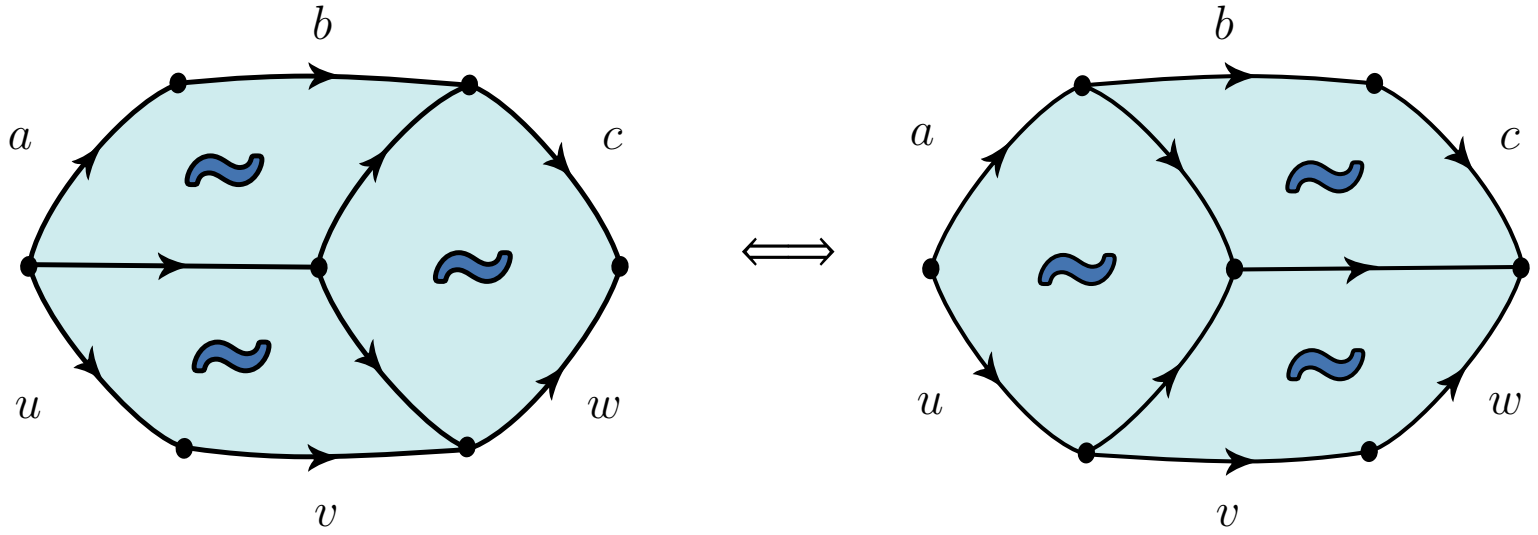
in the asynchronous graph **G**.

In that case, one requires that

$$v_1 = v_2 \quad \Longleftrightarrow \quad w_1 = w_2.$$

The cube axiom

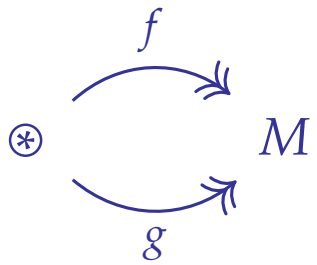
The asynchronous graph $\mathbf{G}$ satisfies the forward cube property:



Characterization theorem

Theorem. An asynchronous graph $\mathbf{G}$ is associated to an event structure precisely when:

- the graph $\mathbf{G}$ contains a distinguished position $\circledast$
- every position is accessible from $\circledast$
- every two cofinal paths



are equivalent modulo a series of tile permutations.