Drifter;Graphs: A dynamically sliced model for Additives

Valentin Maestracci

LDP Team
Aix-Marseille University

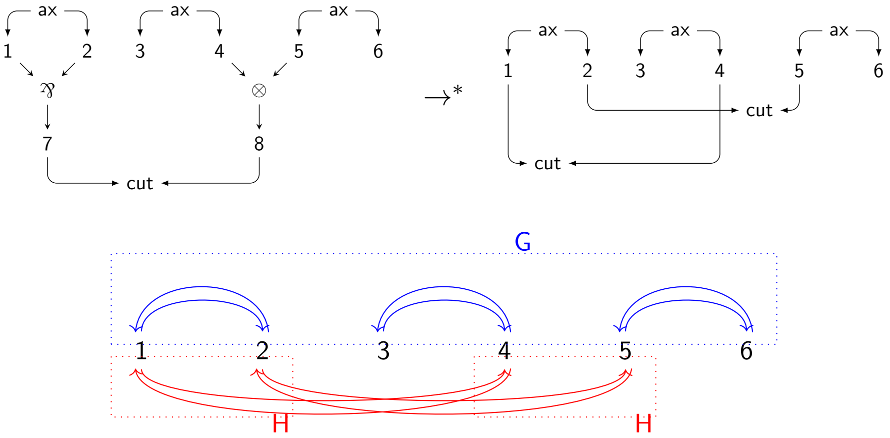
July 18, 2025

Outline

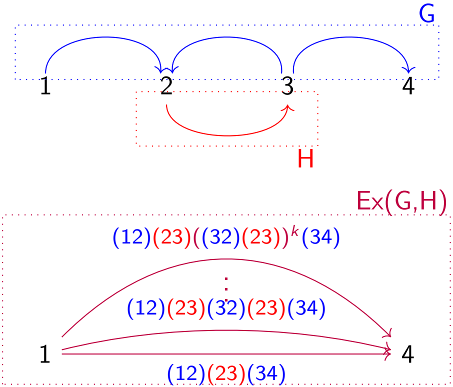
1. Interaction graphs: a model of computation for GoI
2. A new presentation of Execution
3. Drifter;Graphs: a dynamically sliced model
4. Conclusion

Interaction graphs: a model of computation for GoI

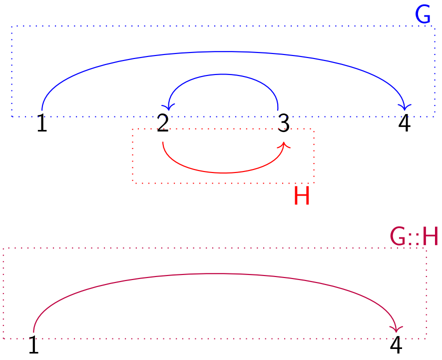
From Linear Logic to Interaction Graph



Execution of Interaction Graph



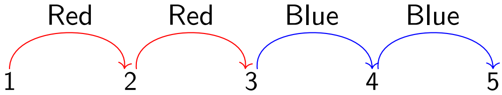
Disappearance of internal cycles



Notice how we can see the above as 1 program with colors instead of 2 programs.

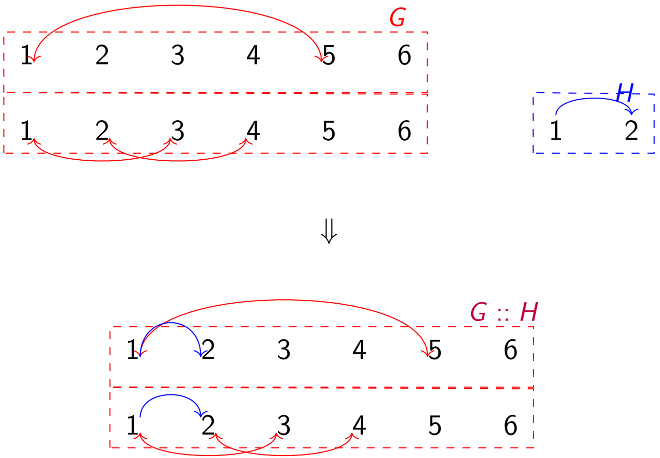
Supercalifraligisquick explanation

Imagine a small step semantics to comose the arrows in the middle:

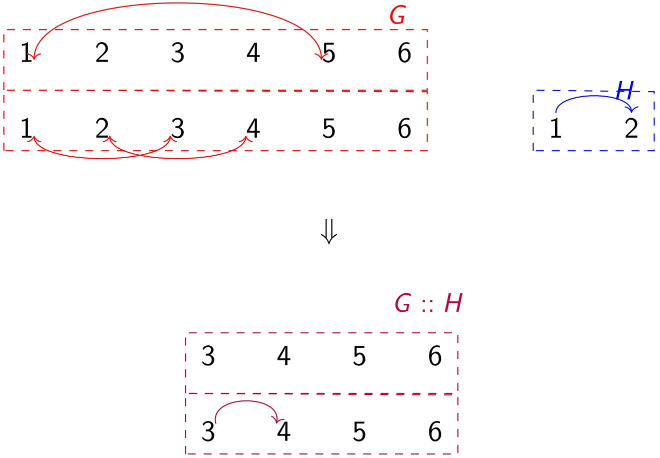


What color should the arrow be?

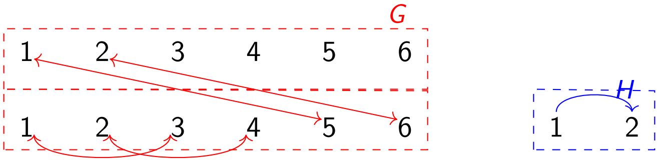
What about additives?



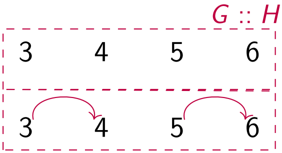
What about additives?



What about additives for LL?

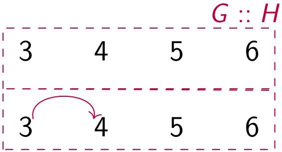


$\Downarrow$



Useless slices

There is no need to remember the slice above!



Let's compute slices dynamically instead!

A new presentation of Execution

A reformulation of execution

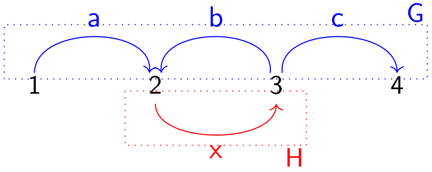
Traditionally, execution = maximal alternated paths.

A reformulation of execution

Traditionally, execution = maximal alternated paths.

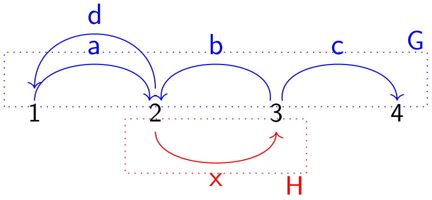
Diagrams

Diagrams are "linear" graphs with edges above every arrow.
$\Rightarrow$ Used as a "unifying notion" with Transcendantal Syntax
$\Rightarrow$ Good diagrams = edges in the execution.



$$\bullet \xrightarrow{a} \bullet \xrightarrow{x} \bullet \xrightarrow{b} \bullet \xrightarrow{x} \bullet \xrightarrow{c} \bullet$$

Diagrams can be...



Not saturated :

$$\bullet \xrightarrow{a} \bullet \xrightarrow{x} \bullet \xrightarrow{b} \bullet \xrightarrow{x} \bullet \xrightarrow{\quad b \quad} \boxed{3 \in \text{Cut}}$$

Not alternated :

$$\bullet \xrightarrow{a} \bullet \xrightarrow{x} \bullet \xrightarrow{b} \boxed{\bullet} \xrightarrow{d} \bullet \xrightarrow{c} \bullet$$

Not correct :

$$\bullet \xrightarrow{a} \bullet \xrightarrow{x} \boxed{3 \neq 1} \xrightarrow{a} \bullet \xrightarrow{x} \bullet \xrightarrow{c} \bullet$$

What is the point?

Now, defining a new model = defining it's "good" diagrams.

$\Rightarrow$ All theorems (associativity of execution etc...) are just abstract theorems on good diagrams.

Drifter;Graphs: a dynamically sliced model

Assume a set of "wordlines" $\mathscr{W}_I$, which are just indexes for slices.

Tracker

A tracker is a partial function $\mathbf{C} \rightharpoonup \mathscr{W}_I$ from colors to worldlines.
It tracks for each "color" (ie, graph identity) in which slice it is.

Trackers can be ordered $T \leq_S T'$, when $\forall u \in S$, $T(u)$ is undefined or defined and $= T'(u)$.
This induce a coherence relation.

Example

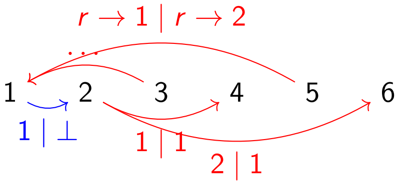
$p \to 1, b \to 1 \frown b \to 1, r \to 2$.

Finally, Drifter;Graphs

Defintion: Drifter;Graph

A graph with a function $T : E \times \{s, t\} \to (\mathbf{C} \rightharpoonup \mathscr{W}_I)$

- $G_i := T(i, s)$, checks that we are in the right worldline to take an edge (precondition).
- $D_i := T(i, t)$, allows to drift from the current worldline to another (postcondition).

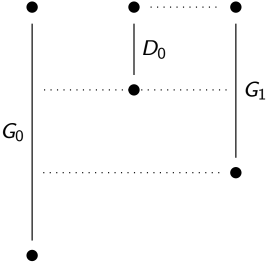


Correct diagrams: when are arrows compatible?

A pair of arrows:

is *timeline coherent* (or *compatible*) when:

- $D_0 \subset G_1$, ie D_0 passes the test of G_1
- $G_0 \subset_{\bar{D}_0} G_1$, ie the tests that are not overwritten by D_0 in G_0 are compatible with those in G_1 (if not then one could not pass both)



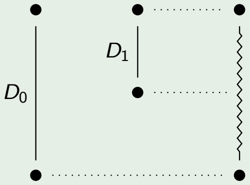
Overriding

Overriding

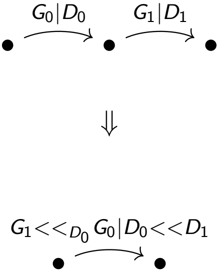
Given trackers from compatible arrows G_0, D_1, G_1, we define overriding as:

$$G_1 <<_{D_1} G_0 : u \to \begin{cases} G_1(u) & \text{if } u \in \overline{\text{dom}(G_1)} \cap \overline{\text{dom}(D_1)} \\ G_0(u) & \text{otherwise} \end{cases}$$

Drifting is $D_0 << D_1$



Composition of arrows

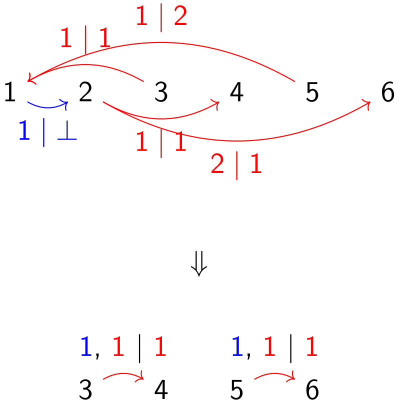


Composition is associative

Given trackers G_0, G_1, G_2 and D_0, D_1, we have:

$$G_2 <<_{D_0 << D_1} (G_1 <<_{D_0} G_0) = (G_2 <<_{D_1} G_1) <<_{D_0} G_0$$

Contraction with Drifter;Graphs



No more mention of 2!

Conclusion

Future Work?

- Still a bit of superfluous data: $1 \mid 1$
- Try to extend it to solve the "problem of the additives" (hard)
- Similar to flows: partial composition, doesn't work well categorically.
- From the abstract notion of composition, can we define other models with "weird" compositions? Can we generalize?

Questions?