Transactional Nets

concurrent construction of proof nets as model for transactional systems

Roberto Maieli

Università degli Studi "Roma Tre"
maieli@uniroma3.it

joint work with Jean-Marc Andreoli
(XRCE, Grenoble)

Seminario GLGC – 20 Marzo 2009

our problem ...

we want to give a logical (proof-theoretical) model of **ACID-Transactions**.

- a "single" (possibly non elementary) logical operation on (DBMS) data is called a **transaction**;
- **ACID** (Atomicity, Consistency, Isolation, Durability) is a set of properties that guarantee that database transactions are processed reliably;
- *Atomicity* refers to the ability of the DBMS to guarantee that either all of the tasks of a transaction are performed or none of them are.
- *Consistency* ensures that the database remains in a consistent state before the start of the transaction and after the transaction is over (whether successful or not).
- *Isolation* refers to the requirement that other operations cannot access or see the data in an intermediate state during a transaction.
- *Durability* (non-reversibility) refers to the guarantee that once the user has been notified of success, the transaction will persist, and not be undone.

our approach ...

our general programme is to set a correspondence:

transactional concepts $\Leftrightarrow$ proof-theoretical concepts

Computation	**Proof theory**	
paradigm	*functional*	*logic*
state	proof: with cuts without proper axioms	proof: cut-free with proper axioms
transformation	proof reduction: i.e. cut-elimination	proof construction: i.e. proper axioms elimination
final state	cut-free proof	proof without proper axioms
type	formula	formula/sequent

our technology

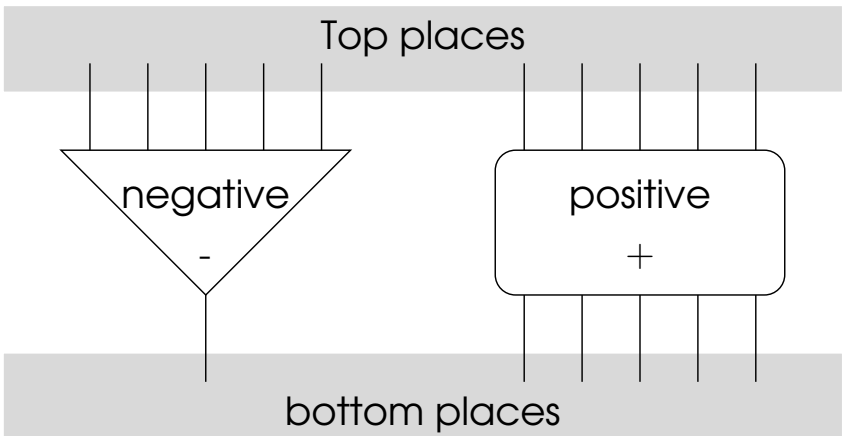
- the construction of MALL Proof Nets (Modules, indeed) by a set of concurrent agents (links);
- this construction (expansion) must be performed in an incremental and local (possibly efficient) way;
- agents are elementary components (positive or negative links) whose role is to guarantee that the overall structure they build is correct (still a proof-module);
- checking correctness of a PN is a task that may require to explore (travel and freeze) a large portion of the structure, resulting in conflicts between agents and forcing them to take turn (scheduling).
- we try to identify "precisely" (a lower boubd) the portion of structure must be checked (and locked) by a candidate (to the expansion) agent in such a way to reduce possible conflicts with other concurrent candidates.

our model of concurrent transactions

- **positive agents** behave like schedulers in transactional models: they role is to ensure that transactions they enact could be serialised, i.e., isolated and executed atomically:
  - the usual way the positive agents achieve this task is by identifying as precisely as possible (using locks) the potential conflicts between transactions and by actually serializing only the conflicting ones;
  - this is known as the *isolation property* of *acid transactions*
- **negative agents** may force transactions to be performed (or aborted) simultaneously.

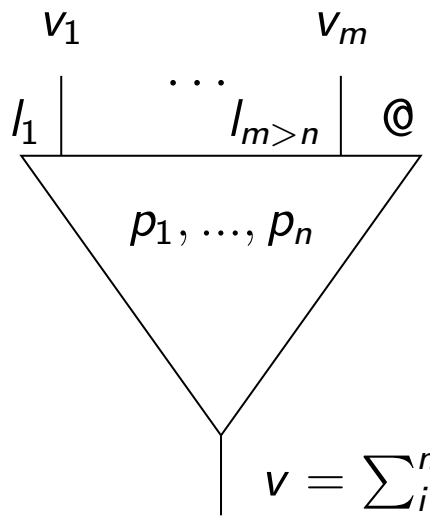
links

- assume an infinite set of *places* (loci, addresses) $\mathcal{L}$;
- a *link* consists in two disjoint sets of loci, top and bottom, together with a polarity $+/-$:
  - a positive link must have *at least one* bottom locus;
  - a negative link must have *exactly one* bottom locus;



if the set of top places is not empty, then a link is said *transitional*.

weighted negative link



$\forall i, 1 \leq i \leq m$,

v_i is a monomial in $Bool(p_1, ..., p_n)$

@ a special place with weight v

$p_1, ..., p_n$, with $n \geq 0$, eigen weights, i.e., Boolean variables

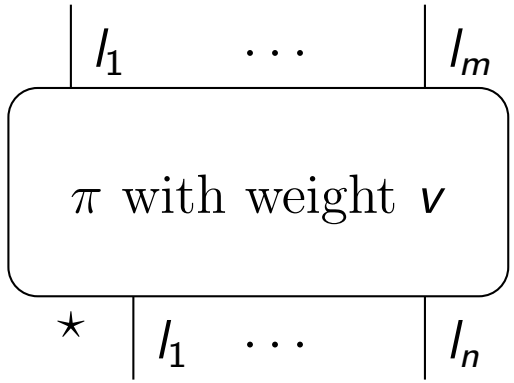
it corresponds to a generalized (via distr.) negative link of MALL

$$\&_{J \in \mathcal{J}} (\parr_{j \in J} P_j^+)$$

with the following conditions:

1. if $n = 0$ then $v_1 = ... = v_m = v = 1$;
2. for each evaluation $\varphi(p_1, ..., p_n)$ there exists, at least, an active top place l_i (i.e., $\varphi(v_i) \neq 0$);
3. for each p_i, $1 \leq i \leq n$, $\sum_j v_j(\epsilon_{p_i})$ must be a monomial, where:
   - $v_j(\epsilon_{p_i})$ denotes a weight v_j containing an occurrence ϵ_{p_i};
   - ϵ_{p_i} denotes an occurrence of p_i or $\bar{p}_i$

weighted positive link



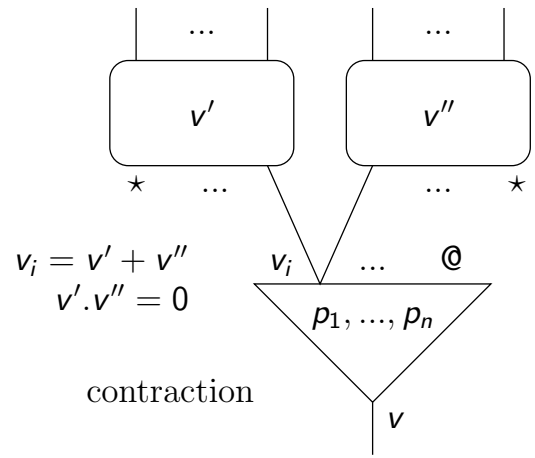
- all palaces (bottom, top and the special $\star$) have the same non-zero weight v (a monomial of the Boolean algebra over the set of eigen weights.
- it corresponds to a generalized (via distr.) positive link of MALL:

$$\oplus_{I \in \mathcal{N}} (\otimes_{i \in I} N_i^-)$$

MALL (focussing) Transactional Structure (TS)

A TS is a set R of links satisfying:

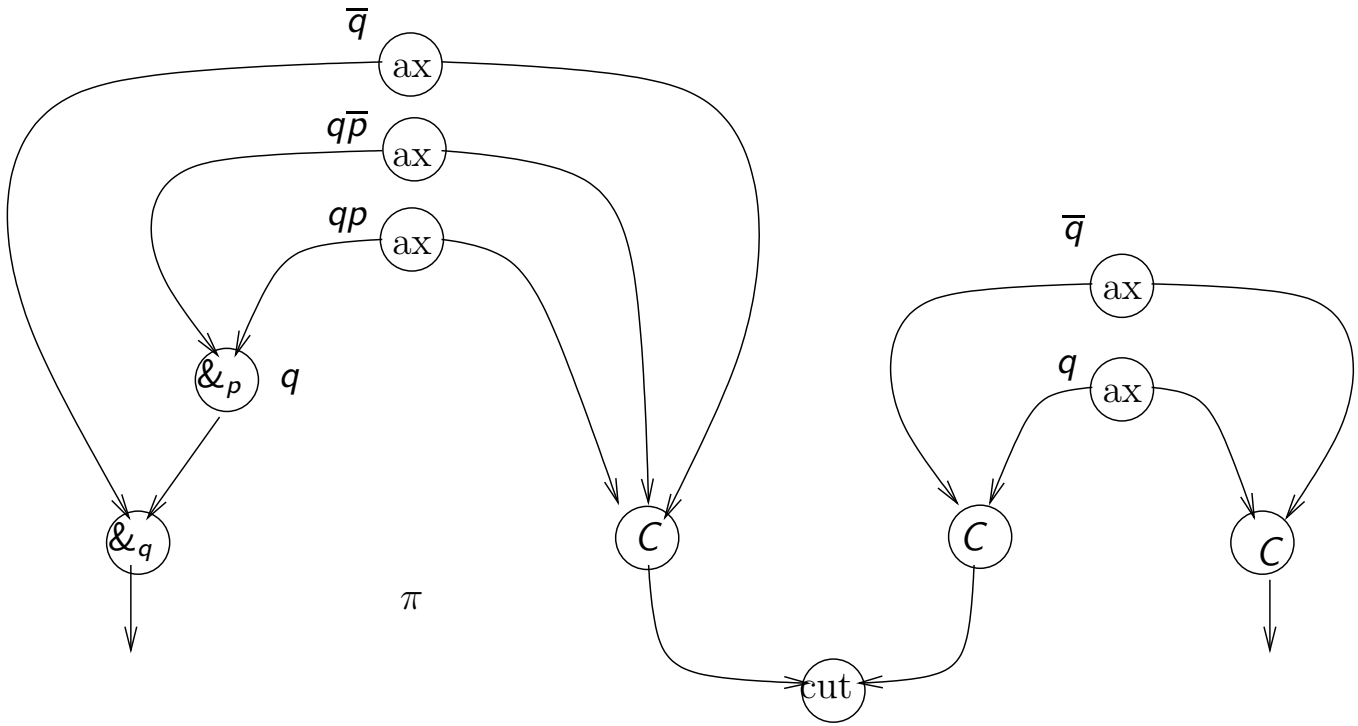
1. the sets of top and bottom places of a link are disjoint;
2. the sets of top (resp., bottom) places of any two links with opposite polarities are disjoint;
3. two adjacent links (with a common place) have opposite polarities and same weight, except in the case of a contraction:



4. if the weight v of a (positive) link y^+ depends on an eigen weight p (i.e., p or $\bar{p}$ occurs in v) of a negative link x^-, then $v \leq \sum_i v_i$, where v_i is the weight of any top places of x^- depending on p too;
5. a bottom border place (it is not top place of any link) has weight 1.

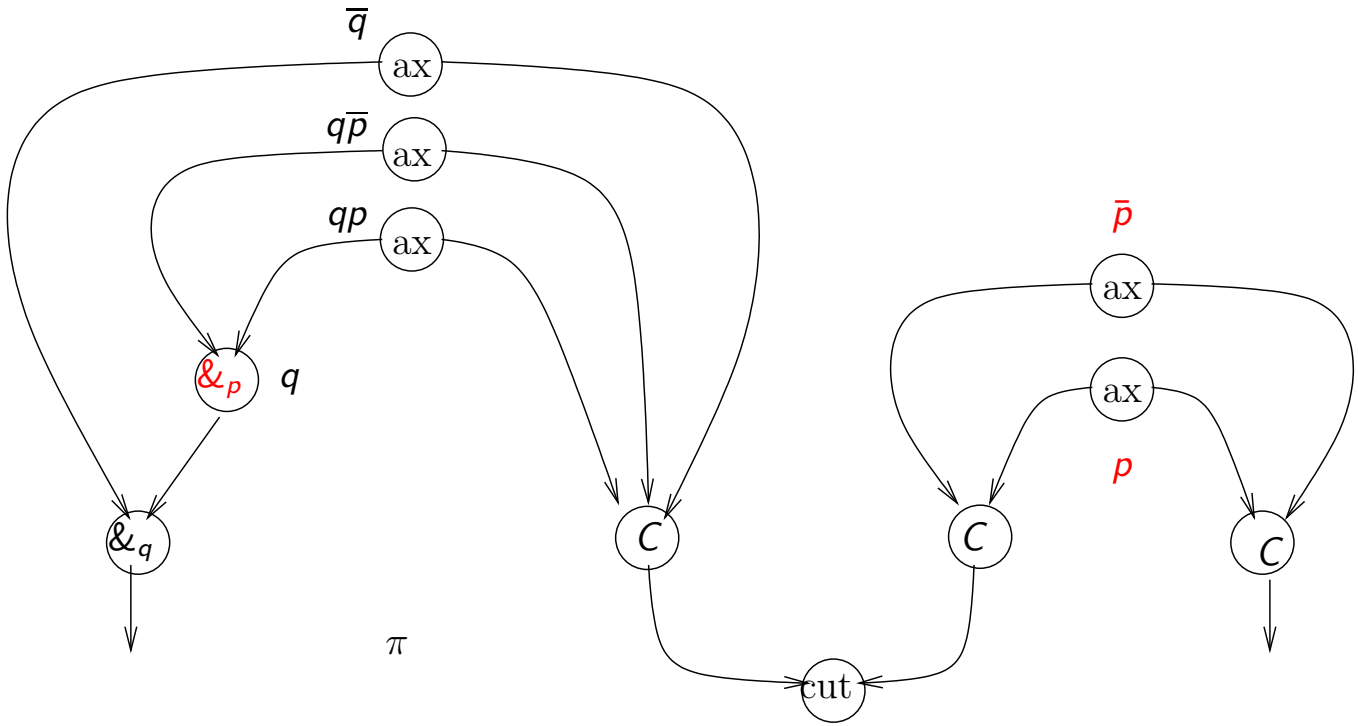
Remark on the monomiality condition (4) of TS

This is a MALL Proof Structure (with weights):



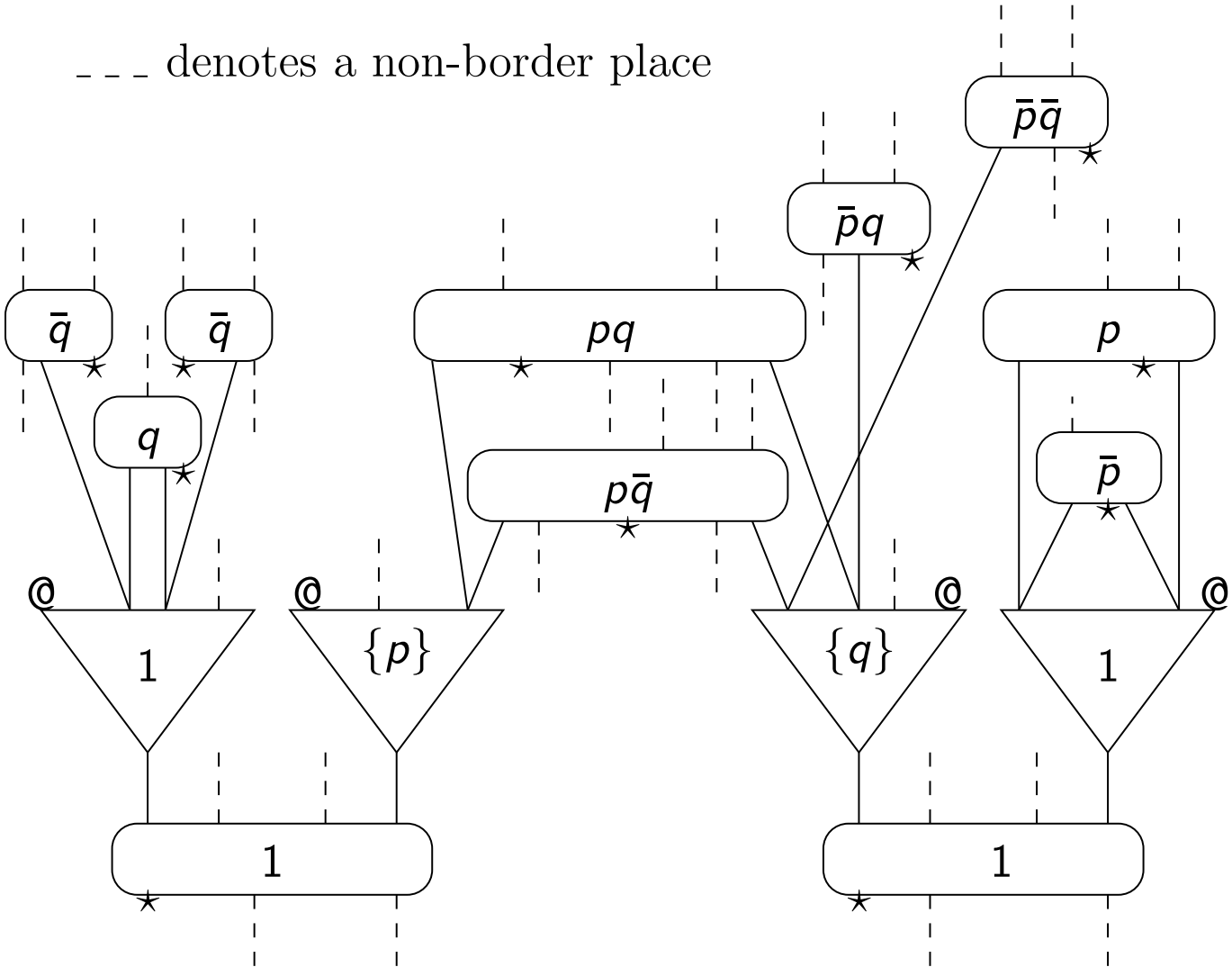
Remark on the monomiality condition (4) of TS

while this is not a proof structure:



it violates the *monomiality condition*:
if a node L depends on p then $w(L) \leq w(\&_p)$
(but, there exists a node whose weight is $\overline{p} \not\leq q$!)

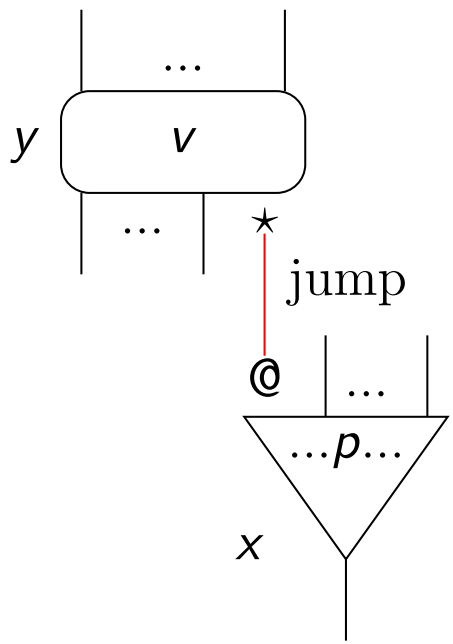
Example (1): a TS



Transactional Nets (1/3): slice and switching

We are interested in correct TS (i.e. TN).

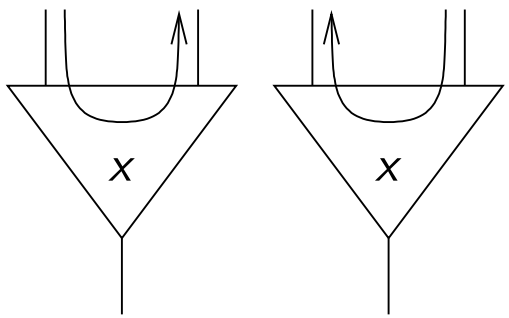
- given a TS R and fixed a valuation φ, a **slice** $\varphi(R)$ is the graph obtained from π by keeping only those links whose weight is 1;
- a **switching** S for R is the slice $\varphi(R)$ in which we add jumps:



from a positive y we jump to a negative x if $v(y)$ depends on an eigen weight p of x

Transactional Nets (2/3): trip and singularity

- a **trip** α in a switching S for R is a non-empty binary relation on $|S|$ (the set of link of S) which is finite, connected and s.t. any $x \in |S|$ has at most one successor (resp., one predecessor), if it exists. Then:
- a **singularity** for a trip



a negative middle point x of a trip α
is a **singularity** for α iff
α enters x downwards and
exists x upwards

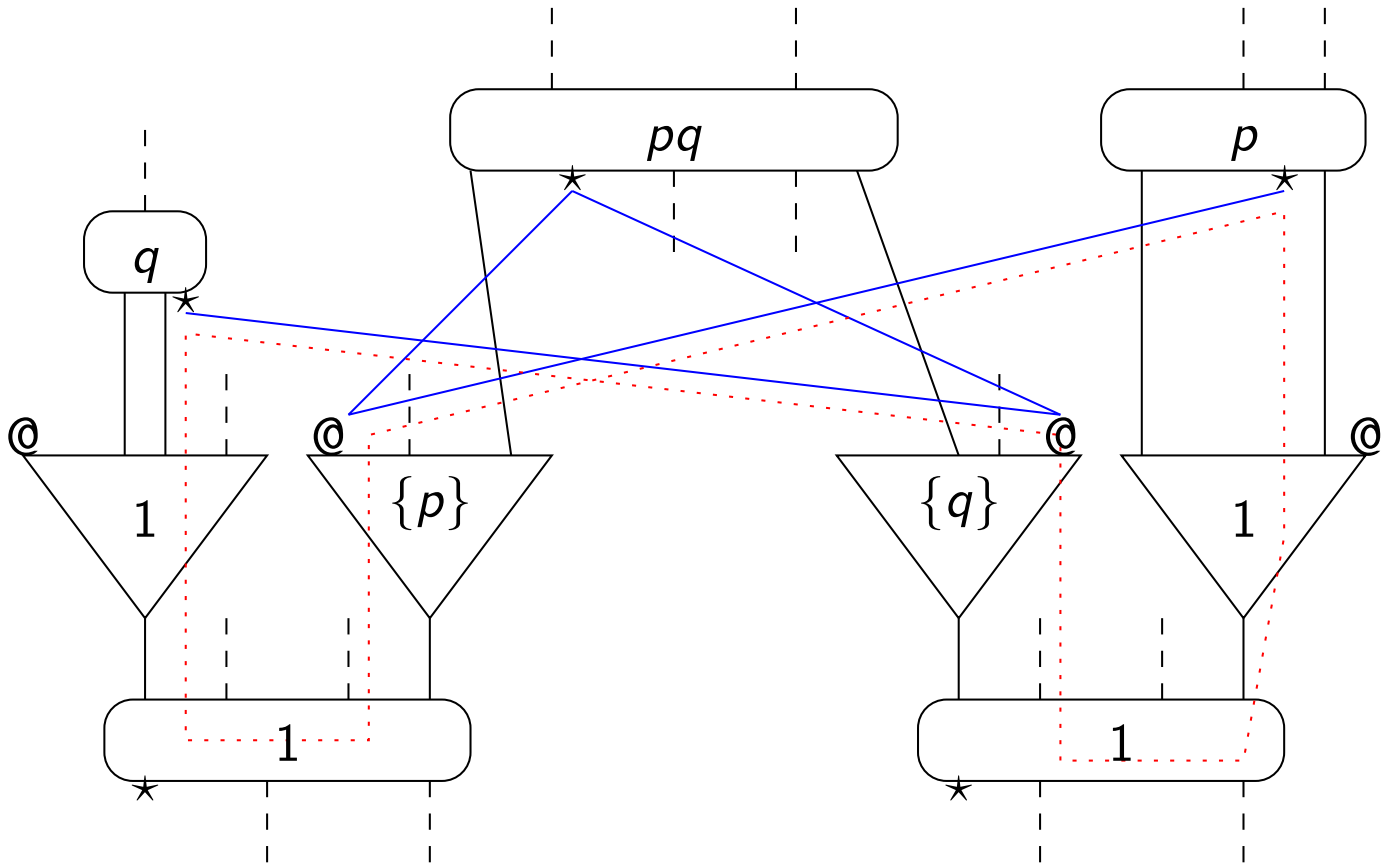
Transactional Nets (3/3): correctness

CC01 : a TS is correct, it is a **Transactional Net**, iff any proper loop (with at least two links) in any switching contains at least a singularity.

- A TN sequetialises into a Focussing Sequent Proof.

Example (2): a TS that is not a TN

The TS in the Example (1) is not correct: there exists a loop in the switched slice S (with $\varphi(p) = \varphi(q) = 1$) that is singularity free.



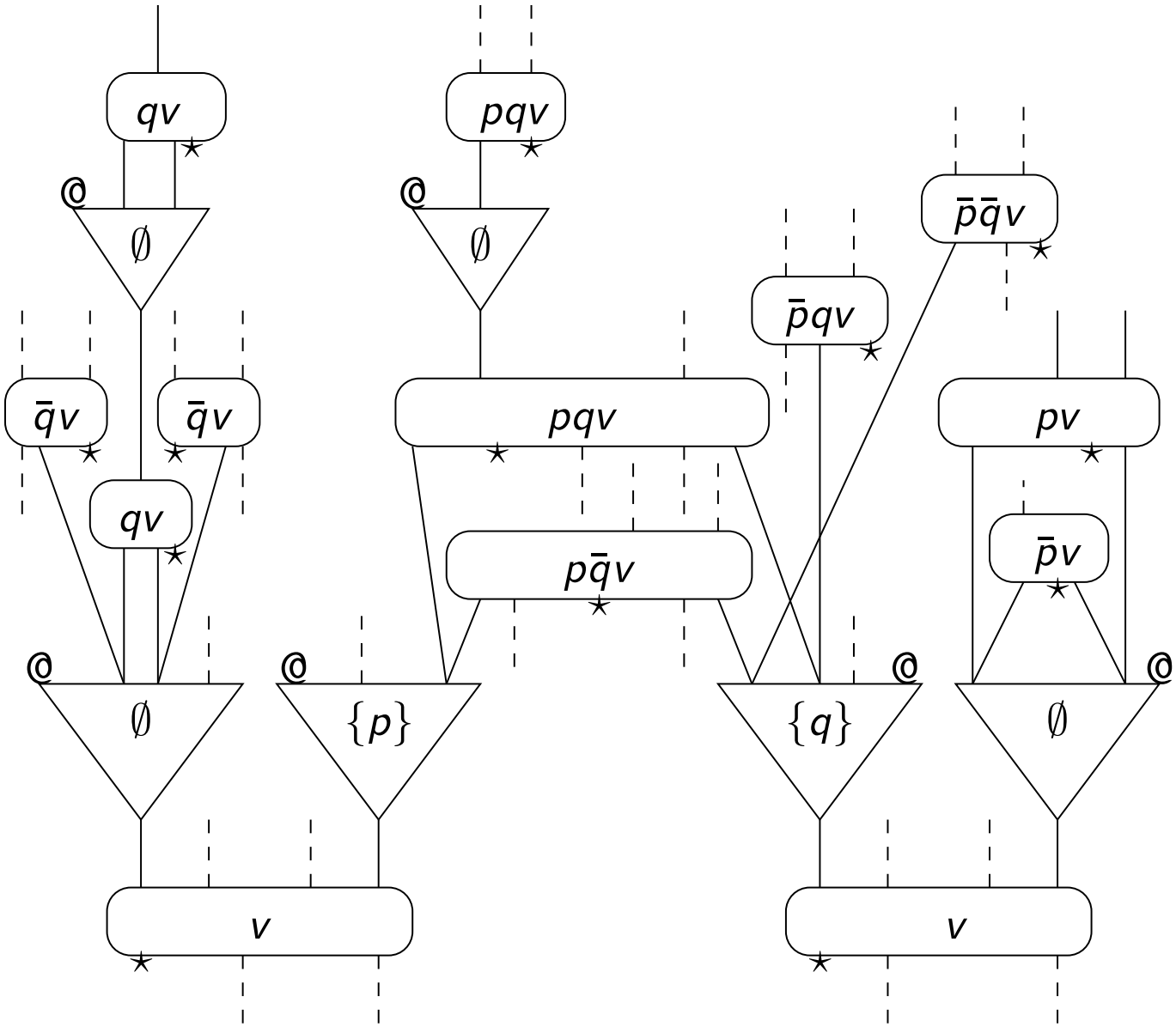
Properties of TN: top switching

- a $+$link y is called **top $+$link** when none of its top places is below a positive link (i.e. there is no path starting from a top place of y and stopping upwards at a bottom place of an other $+$link).
- a **top switching** is a switching where we get rid of any non-top jump (a jump is top when starts from a top $+$link);

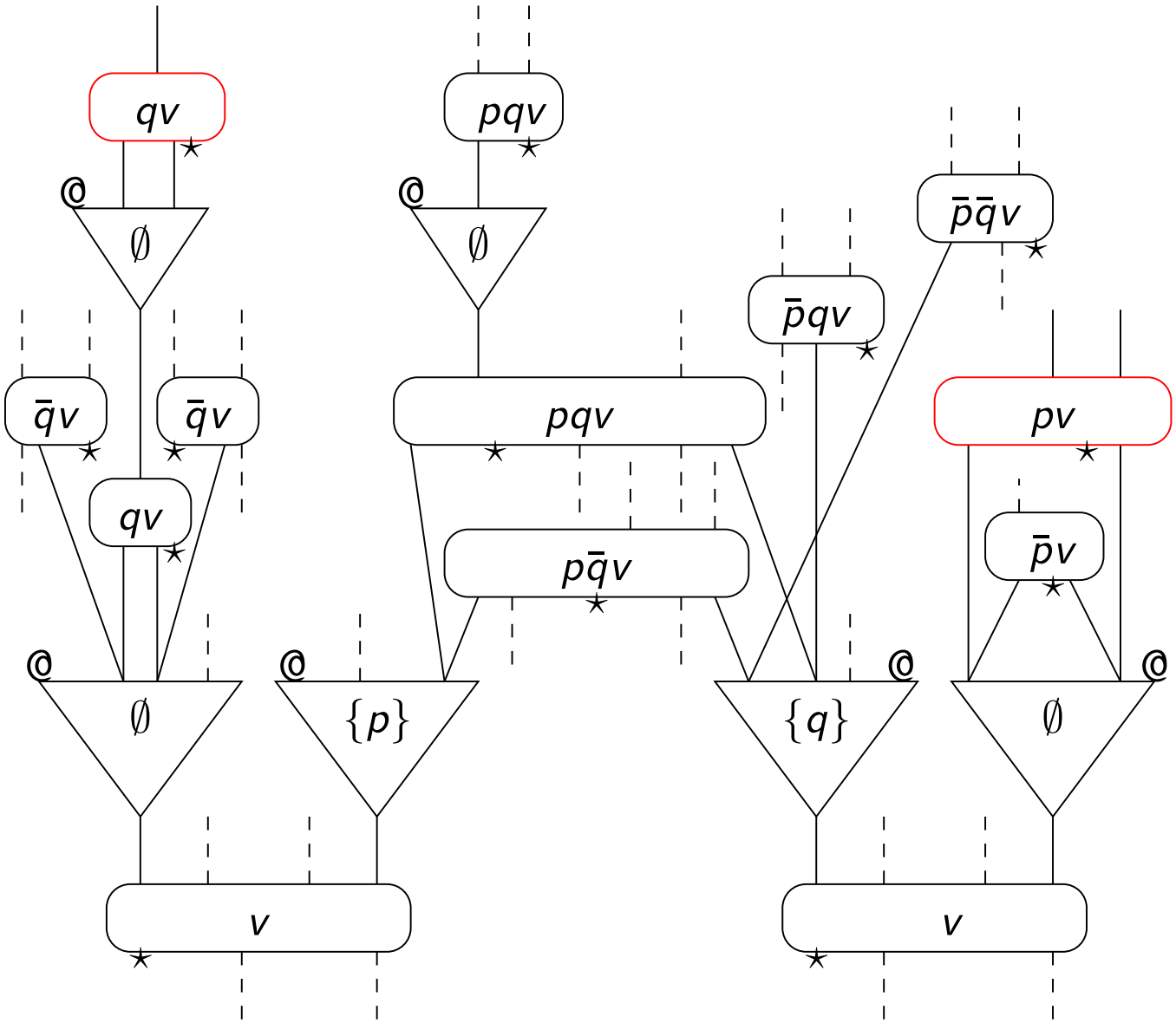
CC02 : a TS is a transactional net, iff any proper loop of any top switching contains at least a singularity.

- $(CC01) \Leftrightarrow (CC02)$

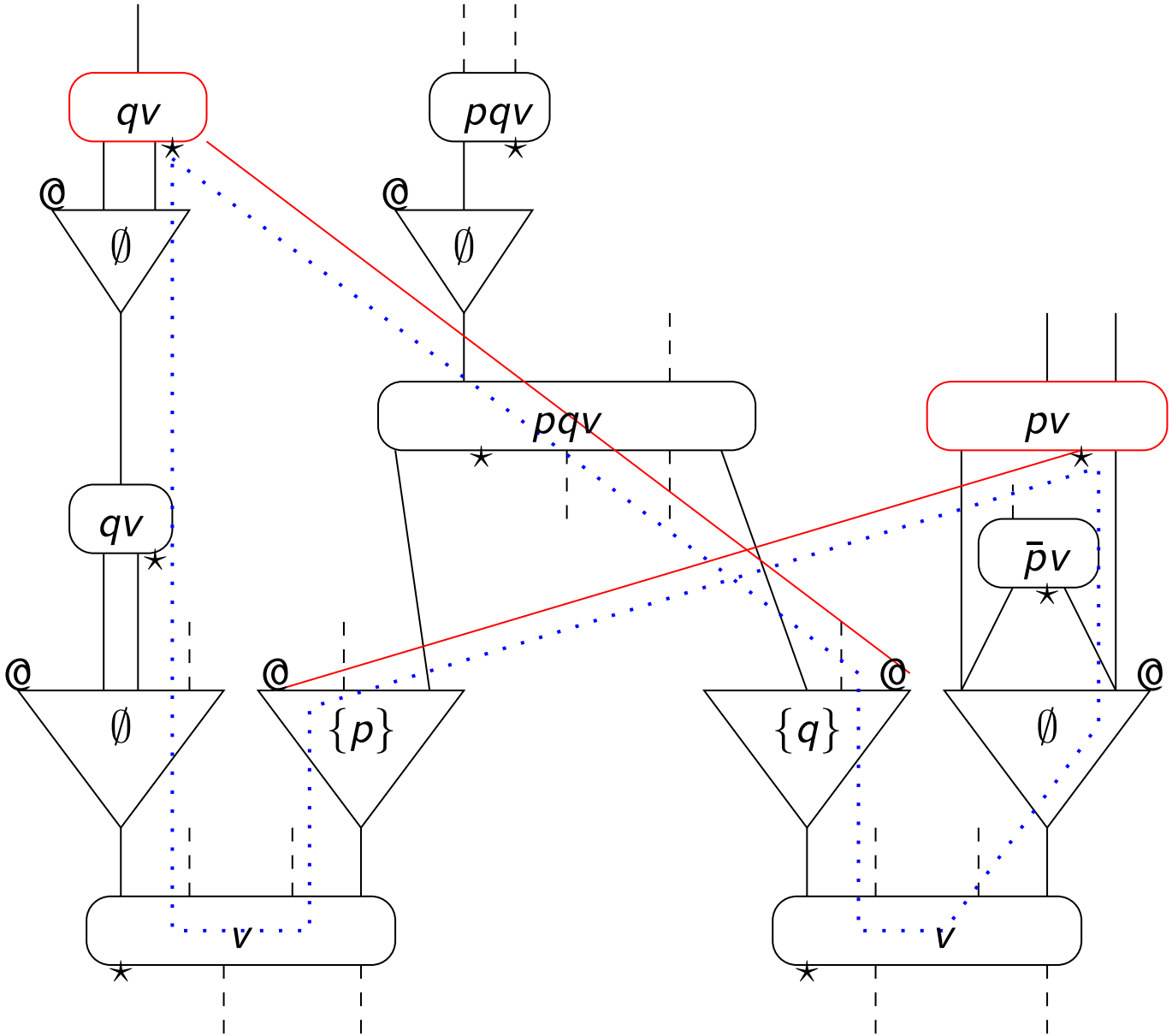
Example (3): a TN



Example (3): top +links (in red)



Example (3): a top switching with a singularity free loop



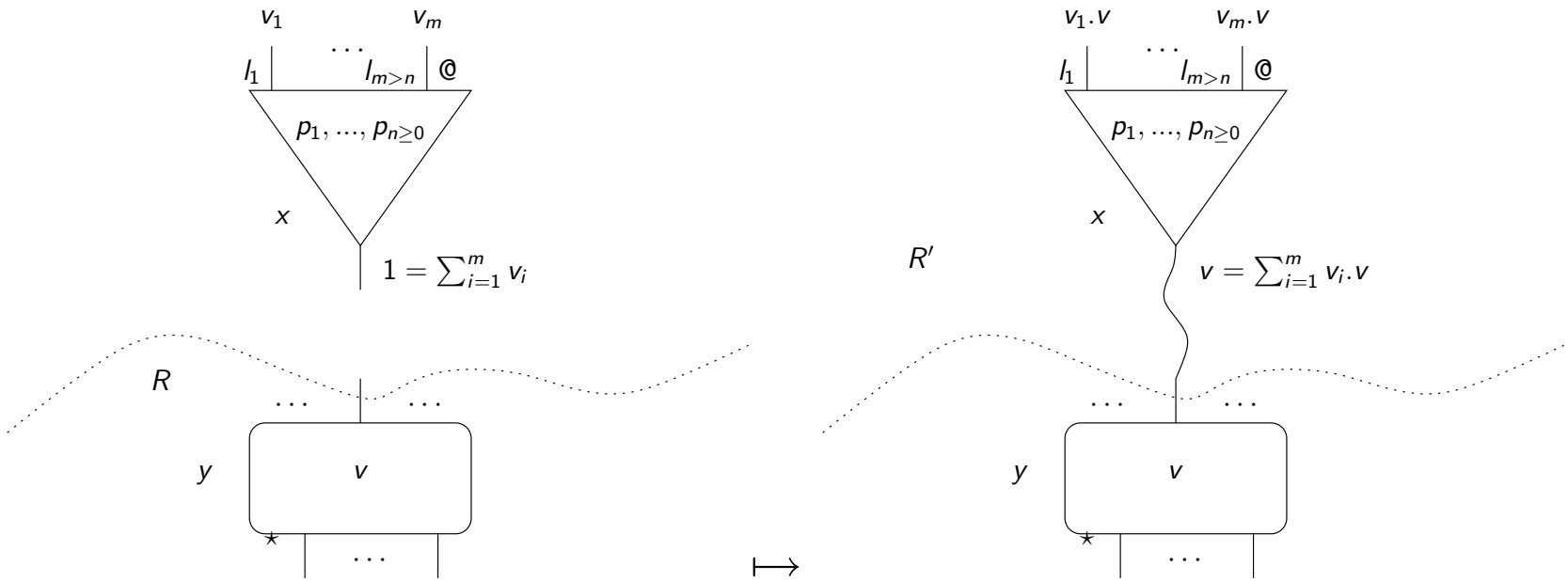
Construction of TN: transaction steps

assume the top border of a TN is not empty

- a positive/negative expansion step consists into adding a positive/negative link at the top of TN built so far.
- the construction proceeds bottom-up and it is driven by "available" top places of links of the TN;
- an expansion step is called transaction if it preserves the property of being a TN (TS + correctness).

negative transaction

assume a TN R and a negative link x whose bottom place matches a positive top border place t of R, then we can expand R by glueing x with R at t (we write $x * R$).

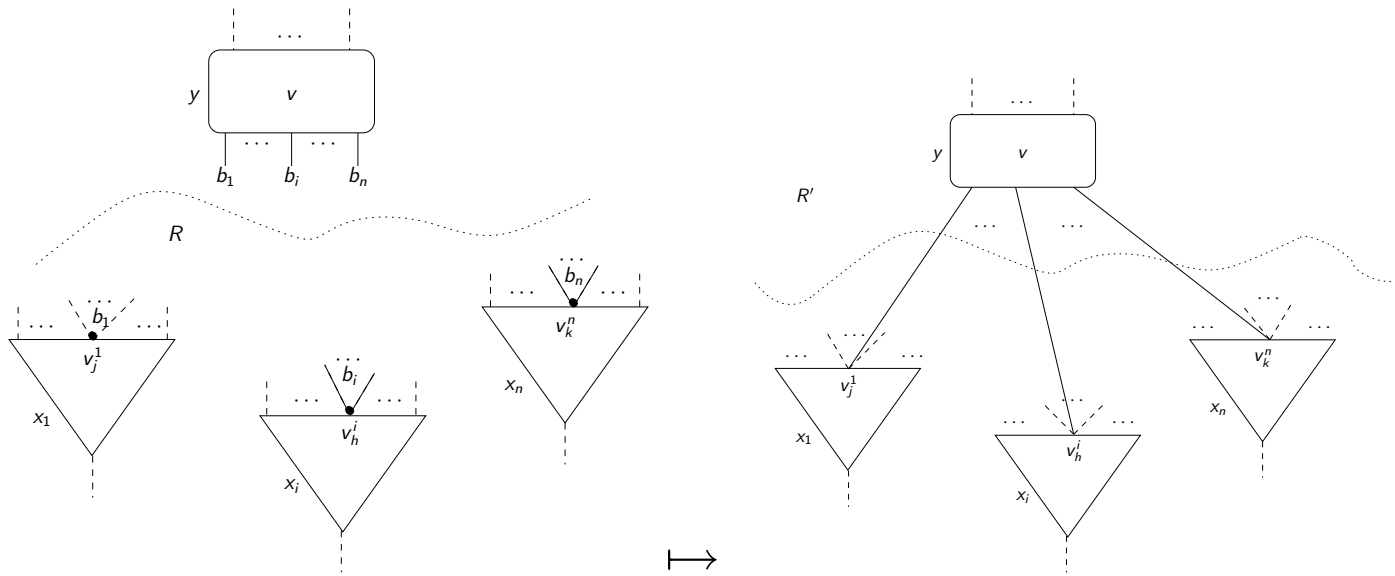


A negative expansion (-**transaction**) step preserves correctness of a TN.

We can perform concurrently clusters of negative transactions.

positive transaction

assume a TN R and a positive link y s.t. each bottom place matches a different top (possibly border) place t of a negative link x of R, then we expand R by glueing x with R at t (we write $x * R = R'$).



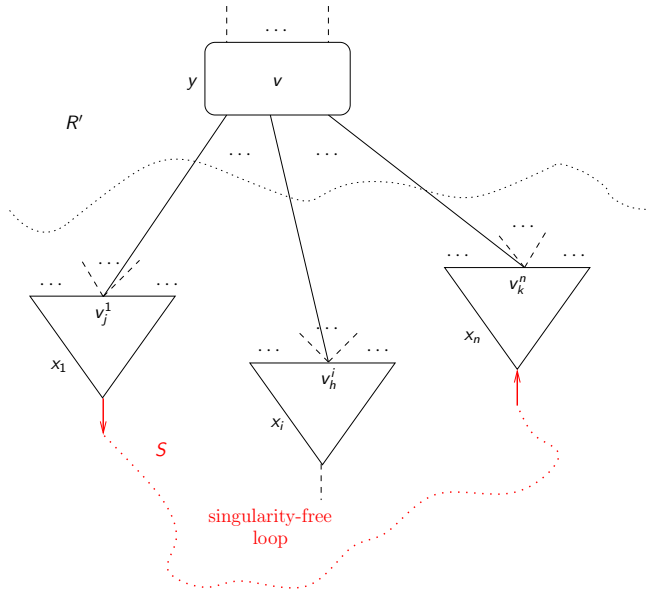
PROPOSITION: A positive y-expansion is correct (a **+transaction**) if :

1. R' is a TS:
2. for each top switching S for R' containing y^+ there is no singularity-free trip starting downwards at some $x \in N$ and stopping upwards at some $z \in N$, where N is the set of negative links at the top places (including @) of which y is connected (also by jumps).

then, $y * R'$ is a TN.

positive transaction

assume a TN R and a positive link y s.t. each bottom place matches a different top (possibly border) place t of a negative link x of R, then we expand R by glueing x with R at t (we write $x * R = R'$).



PROPOSITION: A positive y-expansion is correct (a **+transaction**) if :

1. R' is a TS:
2. for each top switching S for R' containing y^+ there is no singularity-free trip starting downwards at some $x \in N$ and stopping upwards at some $z \in N$, where N is the set of negative links at the top places (including @) of which y is connected (also by jumps).

then $y * R'$ is a TN

positive ACID transactions

- checking the expanded R' is a TS can be performed almost locally;
- checking correctness (singularity-free trips) is a task which may involve visiting (reading/writing) a large portion of R'.

  Since the TN construction is performed collaboratively and concurrently by a cluster of positive (resp. negative) agent/links, for true concurrency we need to:

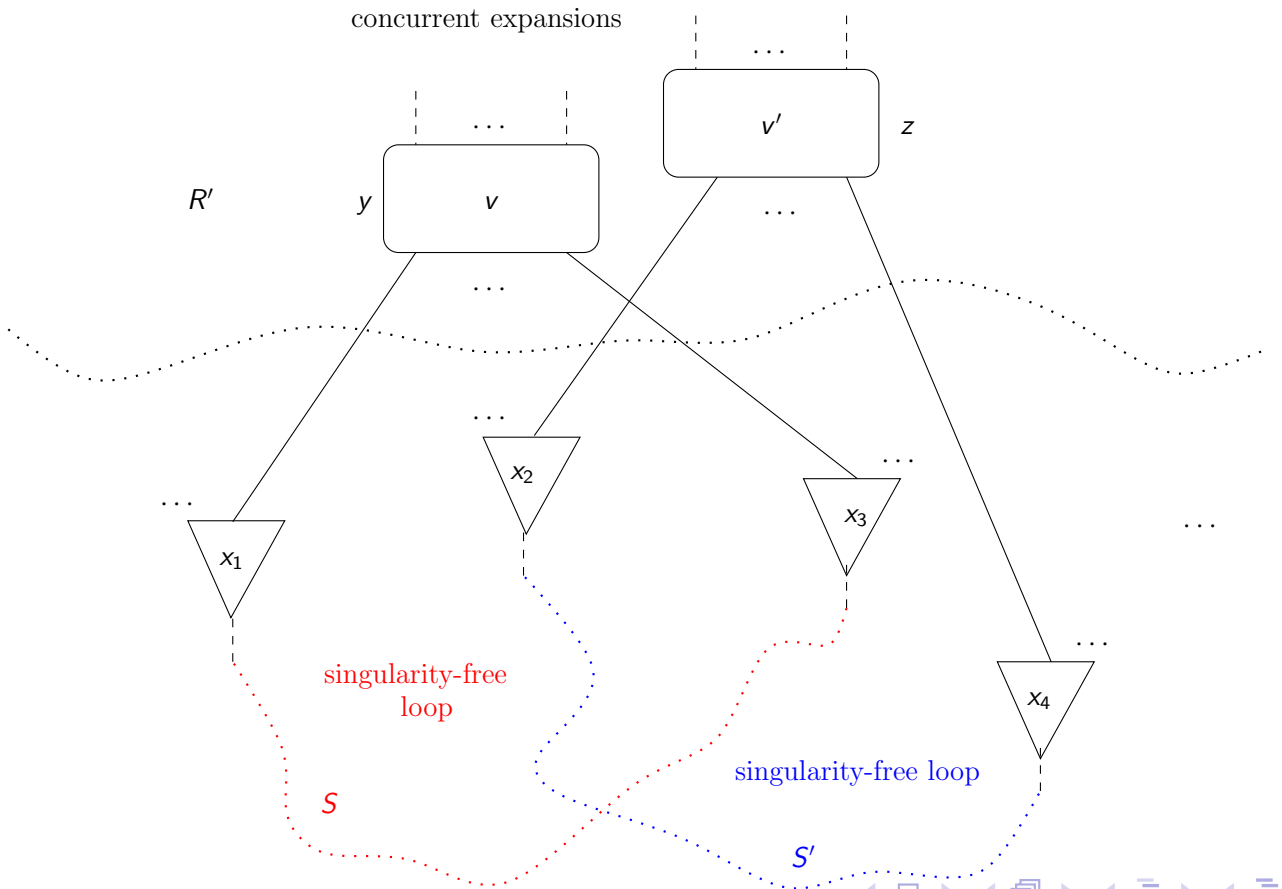
  1. restrict the travelling region (reducing possible conflicts among agents);
  2. protect (lock) the gathered information against attempts of other concurrent $+$agents;
  3. increment/update, in case of success, the locked information for transition (a good bound for this task is necessary).

  Point 1-3 correspond to the **isolation property** of ACID transactions.

positive ACID transactions and weights

Assume a TN R and (at least) two positive agents/links, x and y, that try concurrently to expand R. Then

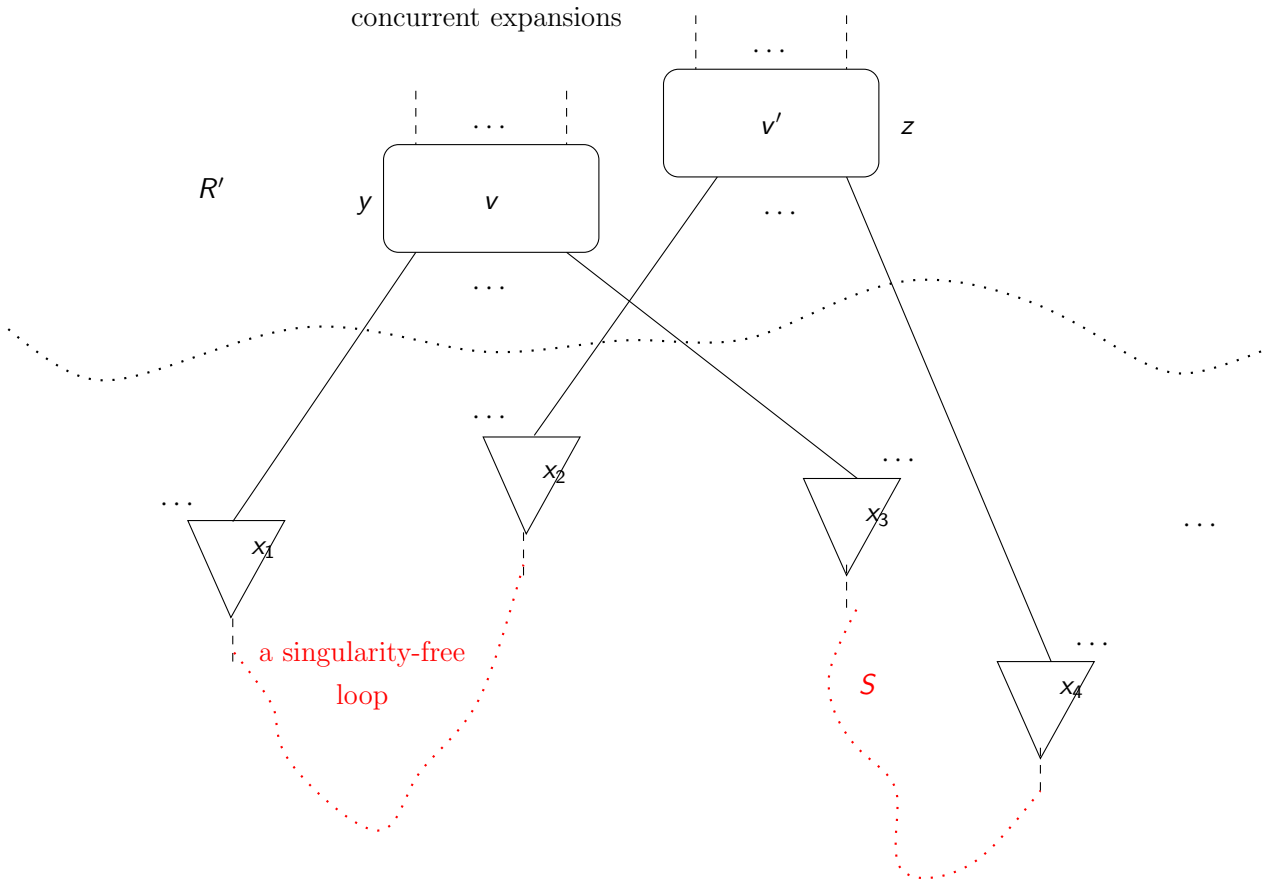
- $v(y).v'(z) = 0$, then the two (possible) corresponding transactions:
  – do not need to be isolated (if they do not share bottom places);
  – do need to be isolated (and eventually performed) as a *pair* of transactions.



positive ACID transactions and weights

Assume a TN R and (at least) two positive agents/links, x and y, that try concurrently to expand R. Then

- $v(y).v'(z) \neq 0$, then they may interact, so they need to be isolated:



ACID transaction: expansion under domination

- assume x, y negative links in a top switching S for R; a *root* of S is any (positive) link of S that has no link below it. Then:

  $x \leq y$ (x **dominates** y) if any singularity-free trip starting at a root and stopping upwards at y visits x upwards.

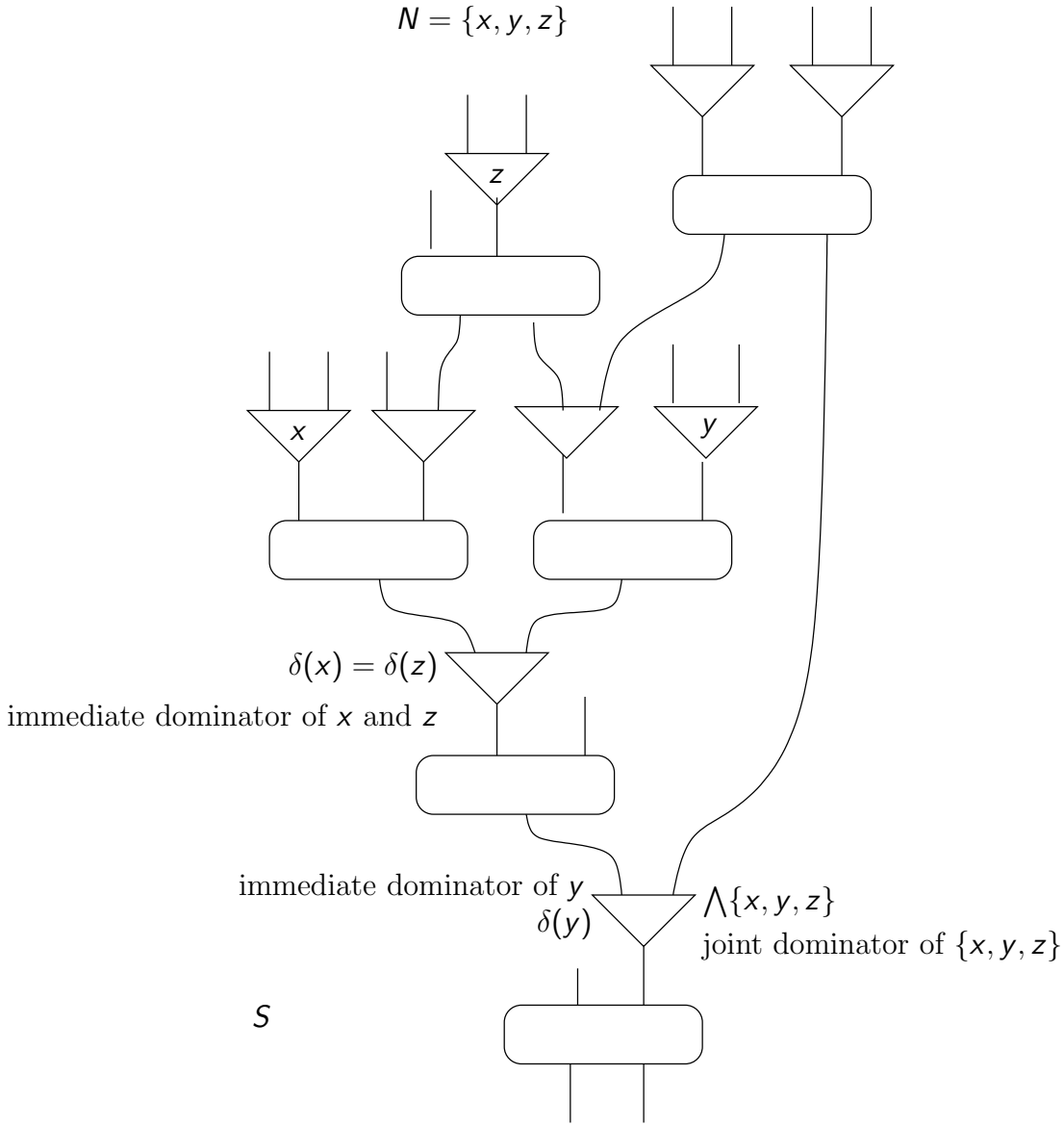
- PROPOSITION: the relation $\leq$ on negative links of S is a **forest order**; it is reflexive, antisymmetric, transitive and it satisfies the following property on negative links:

$$\forall x, y, z \;\; \textit{if} \;\; (x \leq z \;\wedge\; y \leq z) \;\; \textit{then} \;\; (x \leq y \;\vee\; y \leq x).$$

- $\bigwedge(N)$ (the **joint dominator of** N) is the *greatest lower bound* (g.l.b., when it exists), by $\leq$, of a set of negative links N.

- if the set of the predecessor by $<$ of a negative link x is not empty, then it has a greatest element by $\leq$ called the **immediate dominator**, $\delta(x)$, of a negative link x.

Example (4): immediate and joint dominator for $N = \{x, y, z\}$



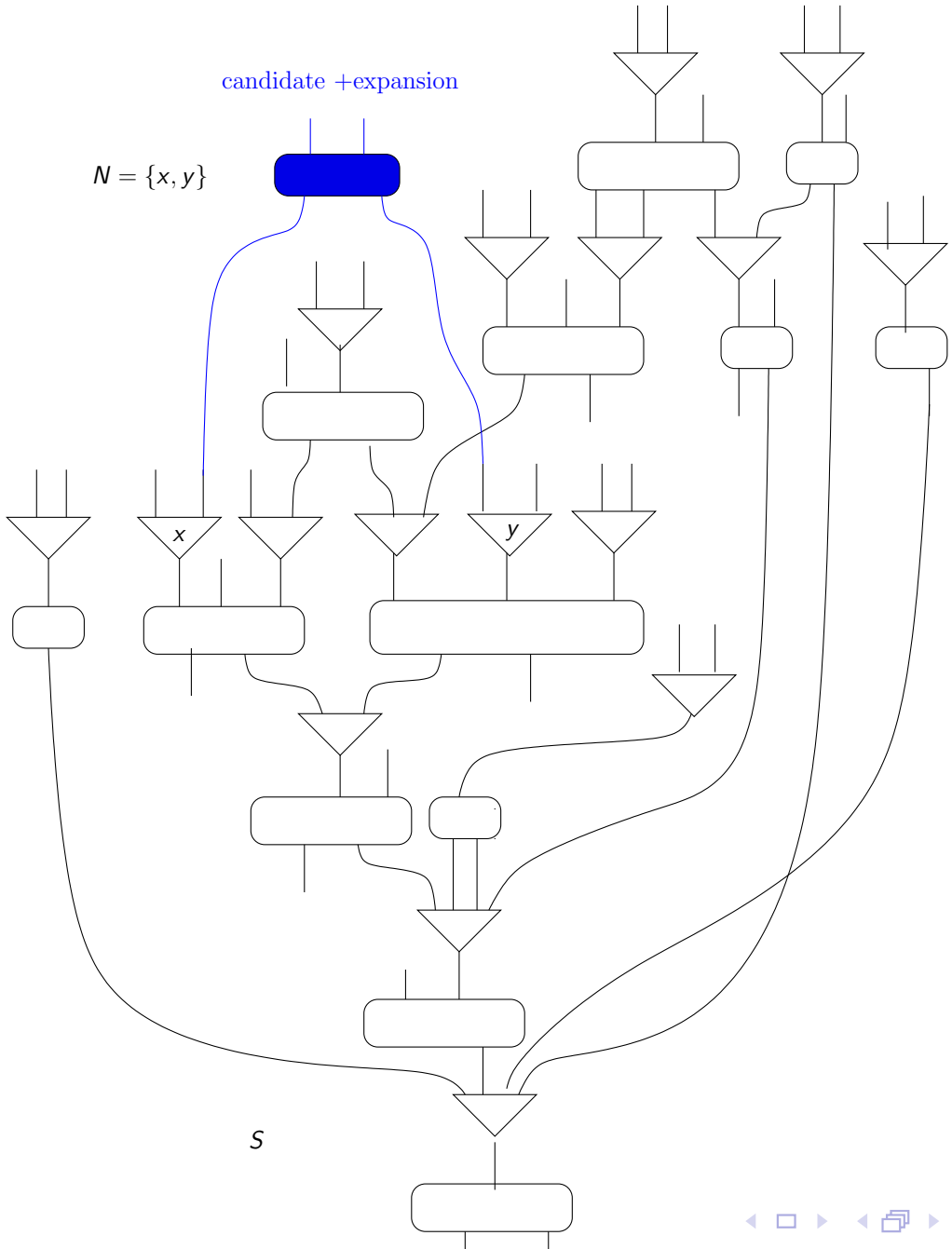
the isolation property

- THEOREM (LOWER BOUND): x, y two negative links and α a singularity-free trip of a top switching S for R starting downwards at x and stopping upwards at y; then, any negative link z visited by α is **strictly dominated** by the joint dominator $\bigwedge\{x, y\}$ (if defined):

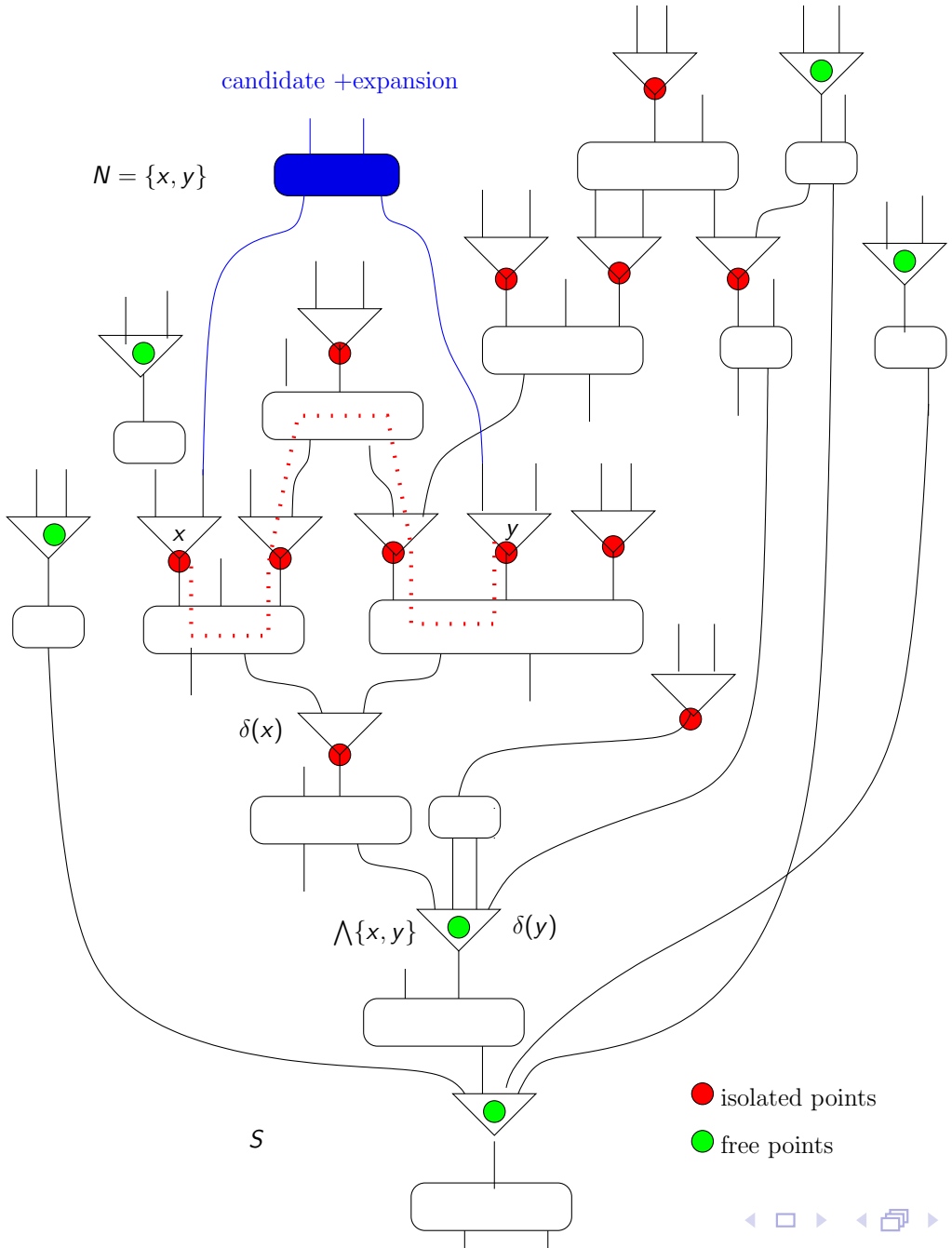
$$\forall z^- \in |\alpha|,\ \bigwedge\{x, y\} < z.$$

- only certain -links must be isolated (locked); the other ones are available for other possible transactions.

Example (5): isolation property



Example (5): isolation property



other (under investigation) questions

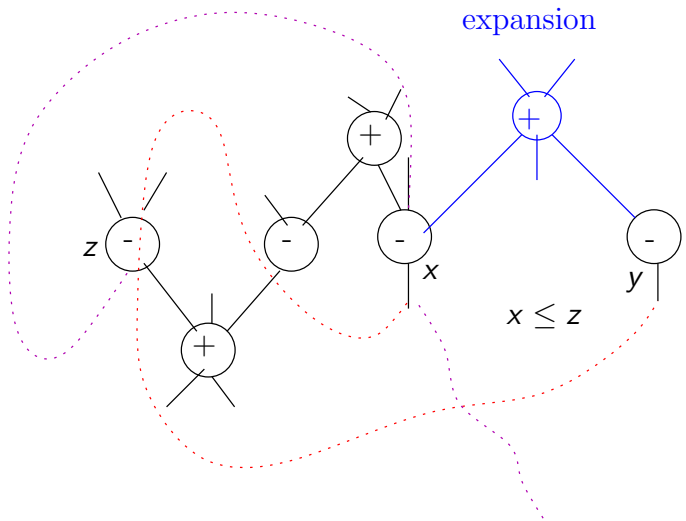
1. additional information should be enable to restrict the region to be explored for detecting singularity-free trips; a better (an upper, not only a lower-)bound to the region we explore/isolate during an expansion attempt.
2. immediate dominators (of negative links in a top switching), if exist, are enough for computing joint dominator (the isolation lower-bound); this information should be updated in case of successfull transaction.

looking for an upperbound for the isolation

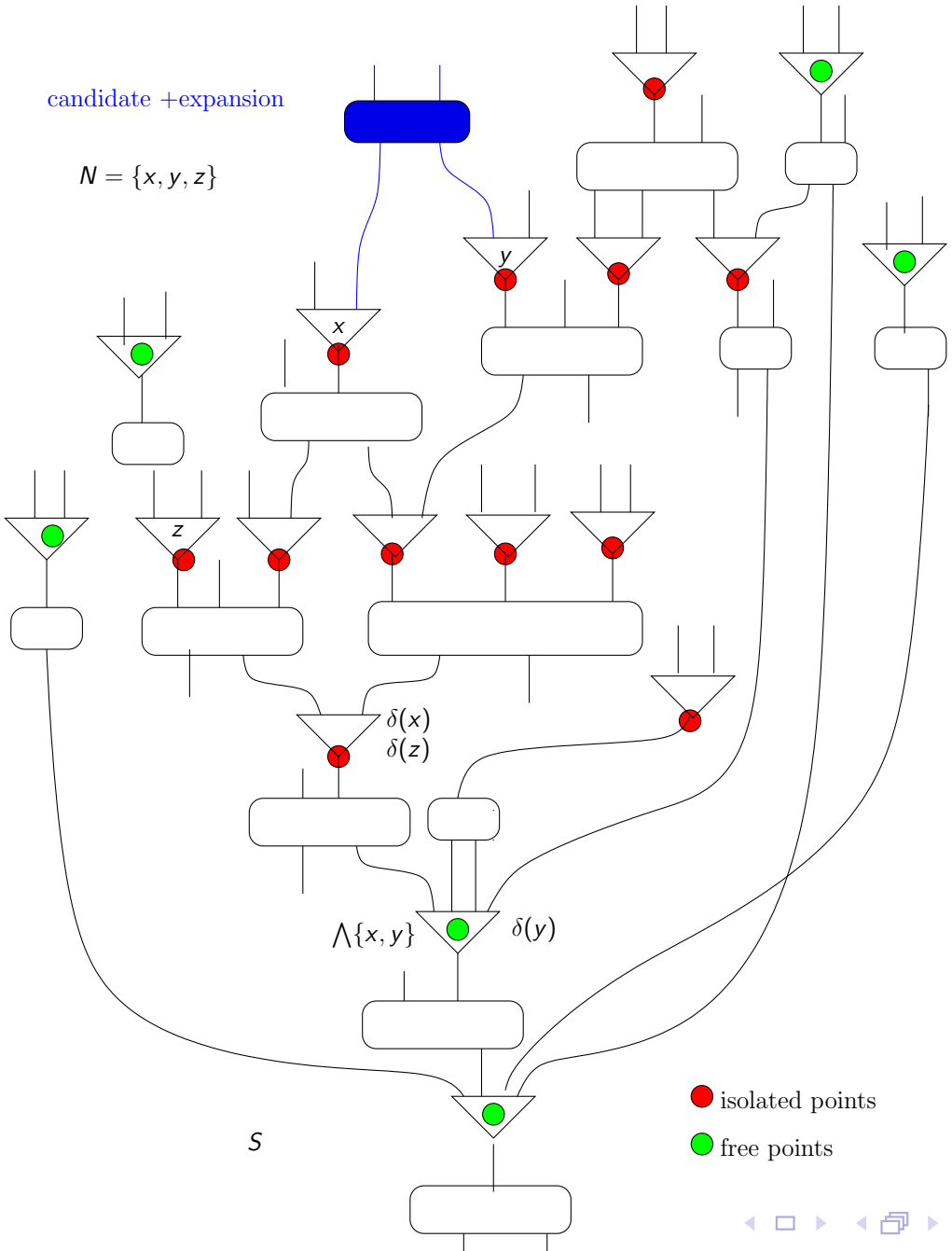
can we better restrict the region to be explored during $+$transaction?

CONJECTURE (UPPER-BOUND) : x, y two negative links and α a singularity-free trip of a top switching S for R starting downwards at x and stopping upwards at y; then, any negative link z visited by α is **strictly dominated** by the joint dominator $\bigwedge\{x, y\}$ (if defined):

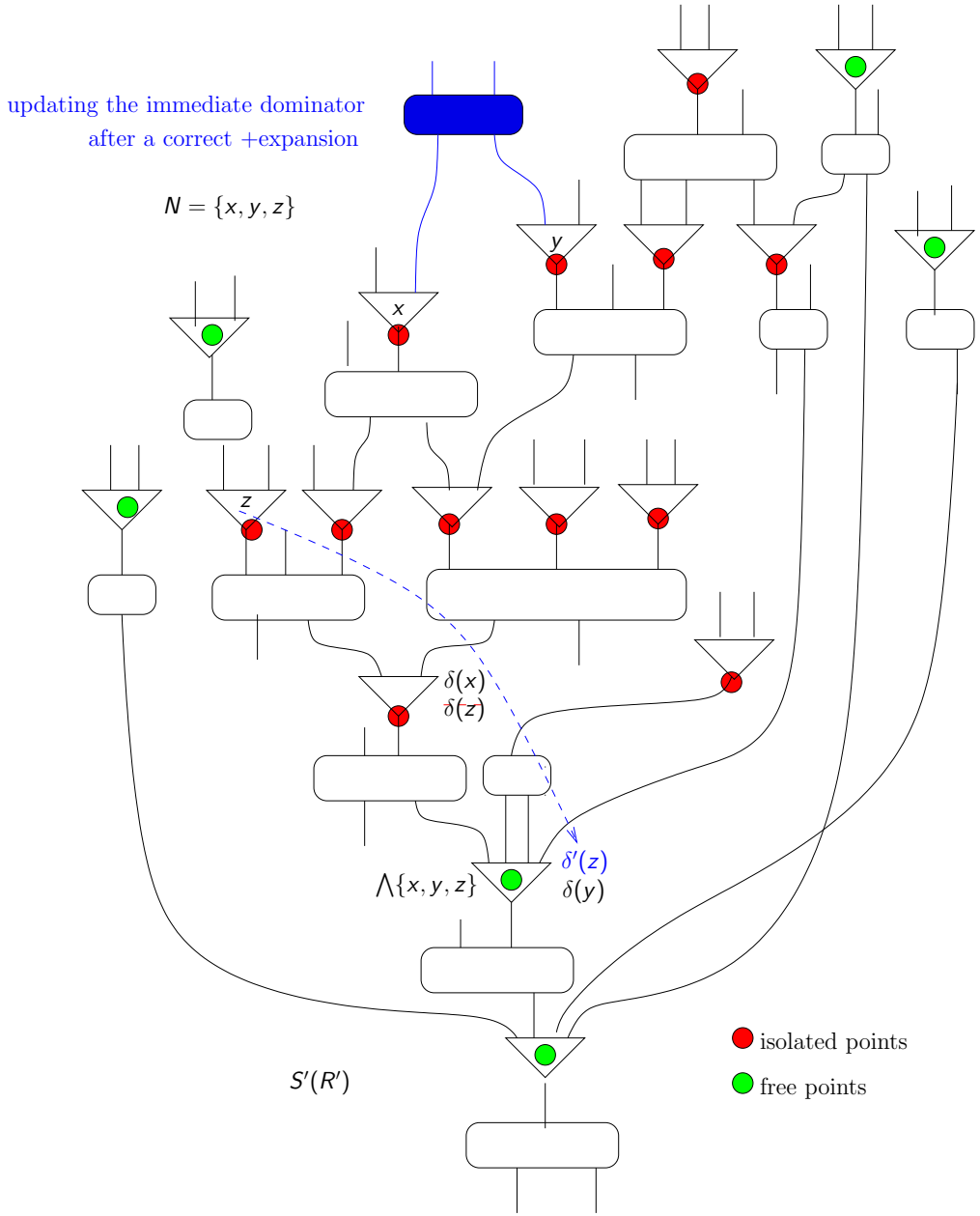
$$\forall z^- \in |\alpha|,\ \bigwedge\{x, y\} < z < x, y$$



updating the immediate dominator



updating the immediate dominator



updating the immediate dominator

Assume R' is obtained after a positive transaction performed on a set of negative links N of R, then:

- either $\delta(x) \leq' x$ and $\delta'(x) = \delta(x)$
- or $\delta(x) \not\leq' x$ and $\bigwedge(N) \leq \delta'(x) < \delta(x)$

where δ' and $\leq'$ are taken w.r.t. R'.

other under investigation questions

a better understanding of the additive behaviour of ”pairwise“ depending transactions.