GPO: A Path Ordering for Graphs

Nachum Dershowitz and Jean-Pierre Jouannaud

WST, 18 July 2018

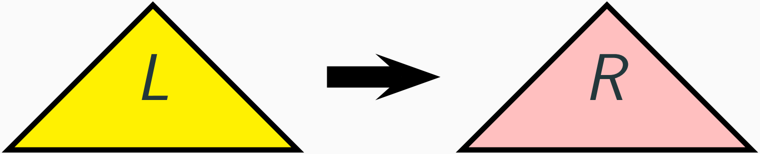
Tel Aviv University and École Polytechnique

Motivation

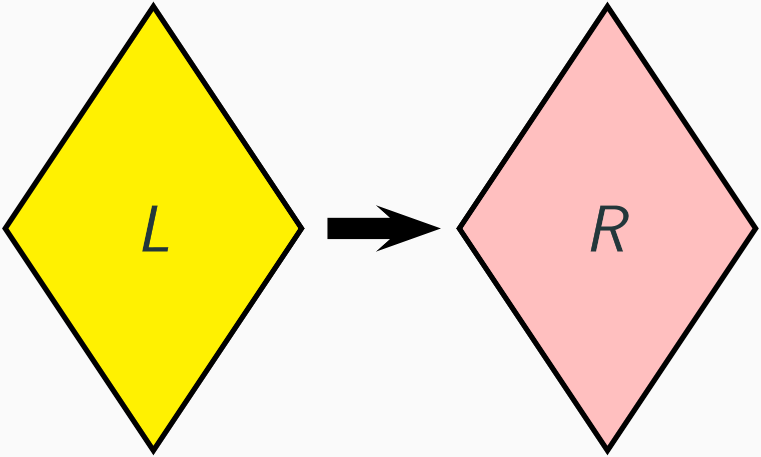
Goals

- Design a total order for comparing polynomials over graphs (algebraic operads)
- *Design a framework for rewriting graphs that generalizes term-rewriting*
- Design a recursive ordering for proving termination

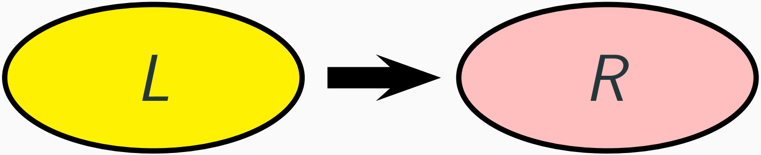
Tree Rewriting



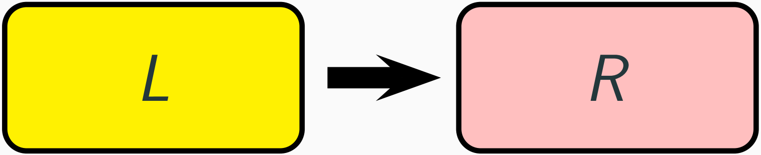
Dag Rewriting



Graph Rewriting



Drag Rewriting



Drags

Drags

Σ functions: 🔴 🔵 🟡

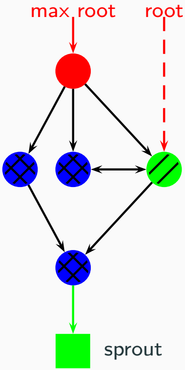
Ξ variables: 🟩 □

A *drag* consists of

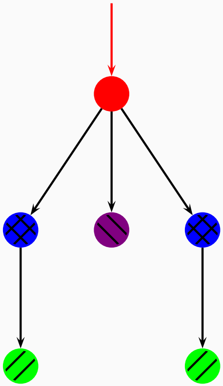
- *vertices* V
- *roots* $R \in V^*$
- *sprouts* $S \subseteq V$
- *labels* $L : S \to \Xi$

  $L : V \setminus S \to \Sigma$
- *successors* $X : V \to V^*$

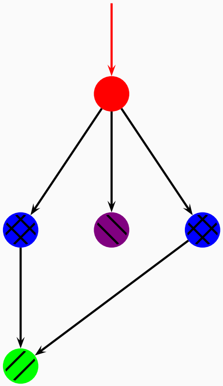
Drag



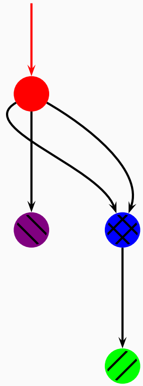
Example 1: Tree



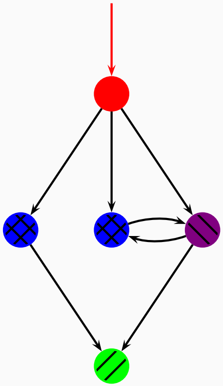
Example 2: Sharing



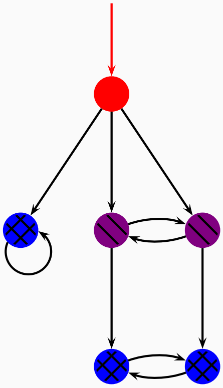
Example 4: More Sharing



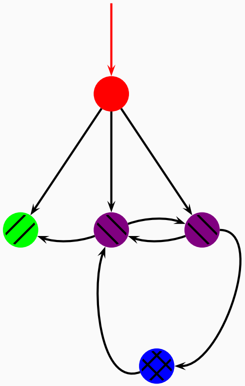
Example 5: Horizontal Sharing



Example 6: Cycles

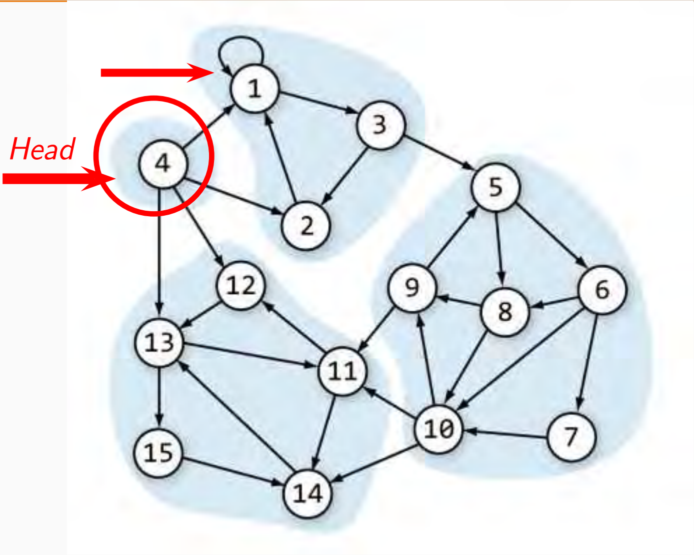


Example 7: Big Cycle

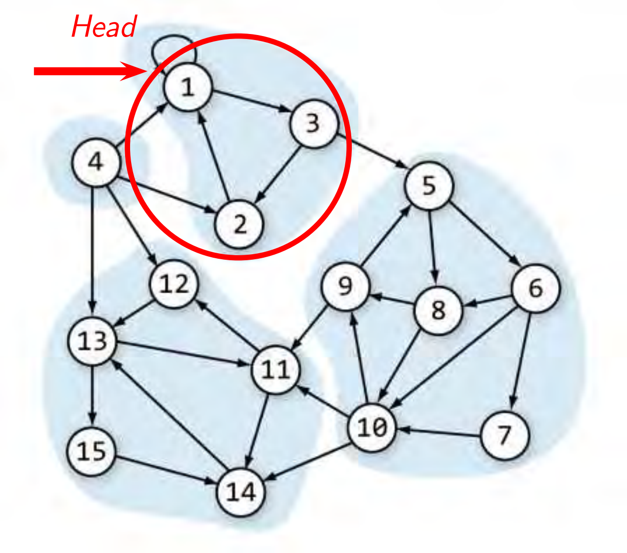


Composition

Components



Components



Subdrag

A *subdrag* is a reachability-closed subset of components – plus their inter-connections.

Switchboard

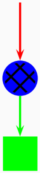
- A *switchboard* links two drags
- Maps sprouts of one to roots of other (and vice-versa)
- Sprouts with same name point to roots of the same (or isomorphic) subdrags

Switchboard Example

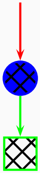
🟩 $\mapsto$ 1

☒ $\mapsto$ 2

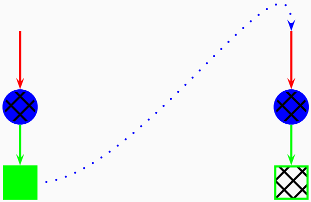
Directional Composition



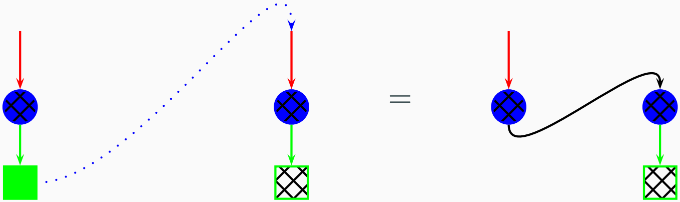
$\otimes_{\blacksquare} \mapsto 1$



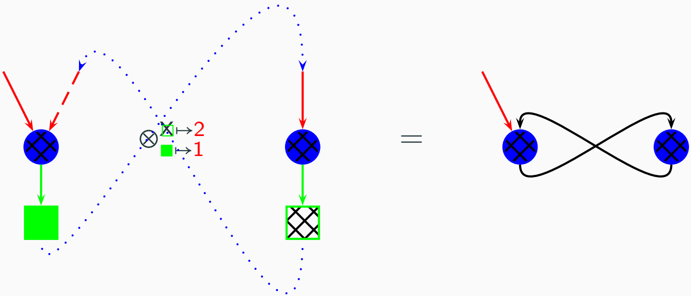
Directional Composition



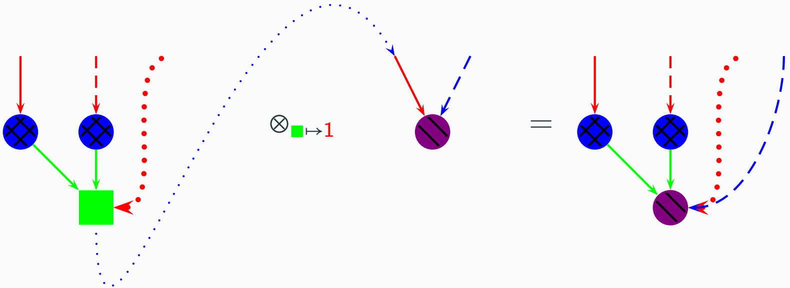
Directional Composition



Cyclic Composition



Root Transfer



Rewriting Switchboard

A *rewriting switchboard* ξ connects context/substitution D to drag *pattern* L

We say $D\xi$ *extends* L

- Roots of L are connected
- Sprouts of L are connected
- Extension variables appear once

Properties

Composition is

- *commutative* (up to permutation of roots)
- *associative* (if names don't conflict)
- enjoys *identities* (switchboards whose vertices are *both* sprouts and roots)

Decomposition

Subdrag

The *subdrag* $D_{|V'}$ of D generated by subset V' of its vertices:

- vertices reachable from V'
- reachable sprouts
- labels and successors restricted to remaining vertices
- remaining roots plus “induced” roots

Antecedent A and directed switchboard ζ s.t.

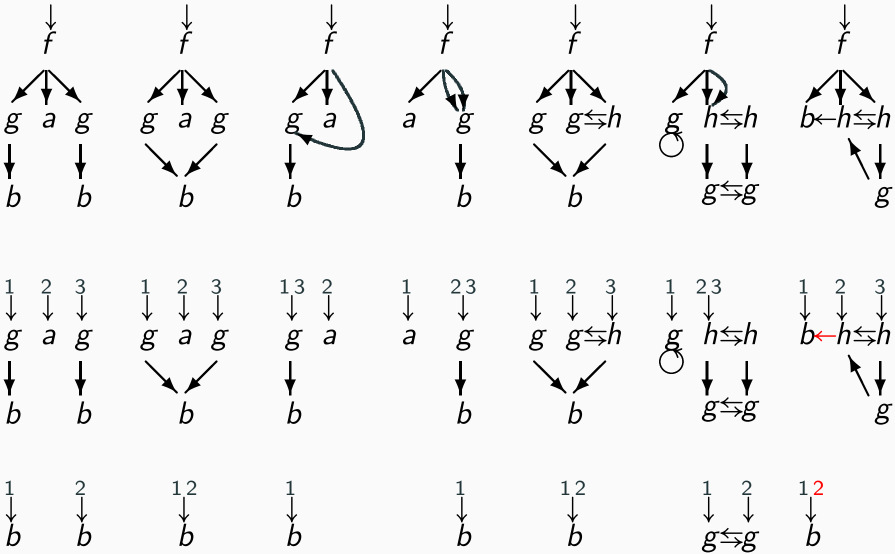
$$D = A \, \zeta \, D_{|V'}$$

Head and Tail

- The *head* $\widehat{D}$ includes vertices from which a maximal root is reachable
- The *tail* ∇D includes all the rest (only *one* tail)
- There's a directed switchboard ζ s.t. $D = \widehat{D} \; \zeta \; \nabla D$

Two drags are isomorphic iff heads and tails are isomorphic

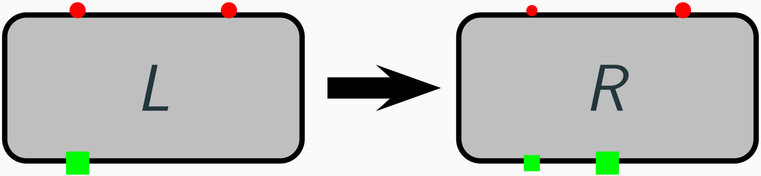
Examples



f f f f f f f
g a g
b b
g a g
b
g a
b
a g
b
g g⇆h
b
g h⇆h
g⇆g
b←h⇆h
g
1 2 3
g a g
b b
1 2 3
g a g
b
1 3 2
g a
b
1 2 3
a g
b
1 2 3
g g⇆h
b
1 2 3
g h⇆h
g⇆g
1 2 3
b←h⇆h
g
1 2
b b
1 2
b
1
b
1
b
1 2
b
1 2
g⇆g
1 2
b

Rewriting

Rules



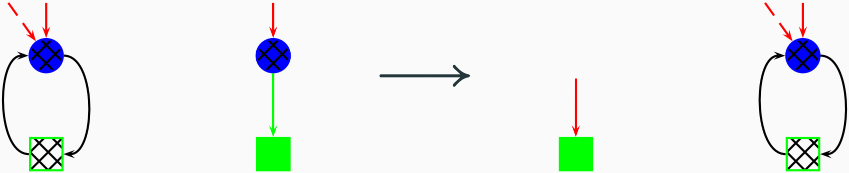
- matching roots on both sides
- no extra variables on right

$D \longrightarrow D'$ if there's an extension $W\xi$ of L s.t.

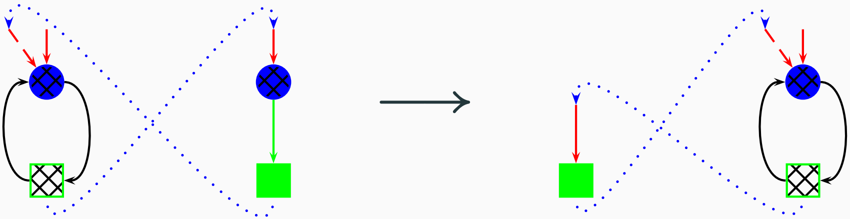
- $D = W\ \xi\ L$
- $D' = W\ \xi\ R$

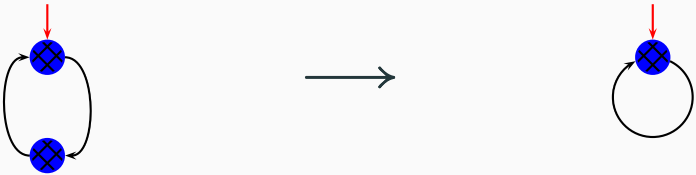
Rewriting tail does not affect head!

Example: Shrinking

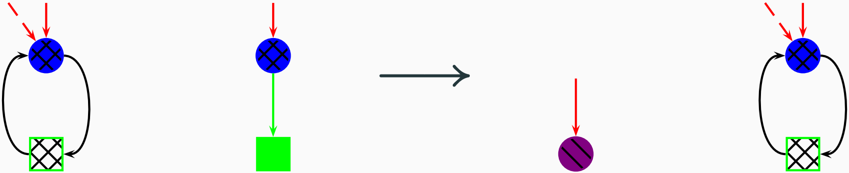


Example: Shrinking

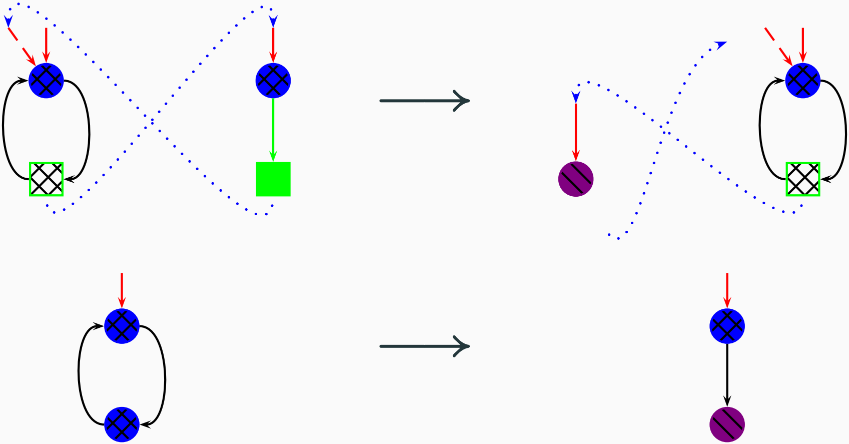




Example: Breaking

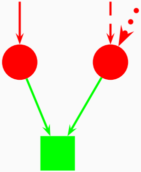


Example: Breaking

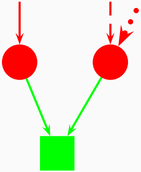


Cyclic Extensions

An extension of D is *cyclic* if all its (non-sprout) vertices are reachable from (roots of) D in the composition

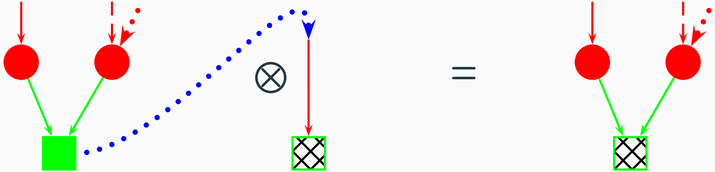


$$\otimes \quad \varnothing \quad =$$



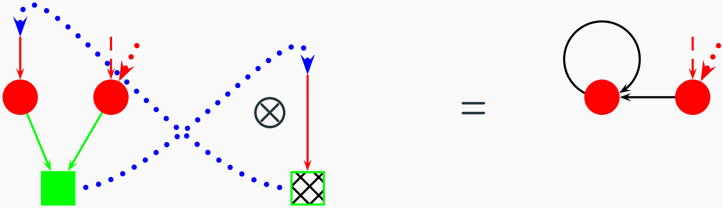
Cyclic Extensions

An extension of D is *cyclic* if all its (non-sprout) vertices are reachable from (roots of) D in the composition



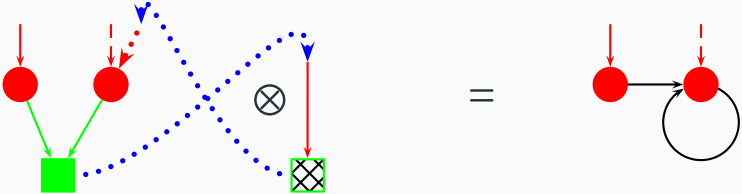
Cyclic Extensions

An extension of D is *cyclic* if all its (non-sprout) vertices are reachable from (roots of) D in the composition



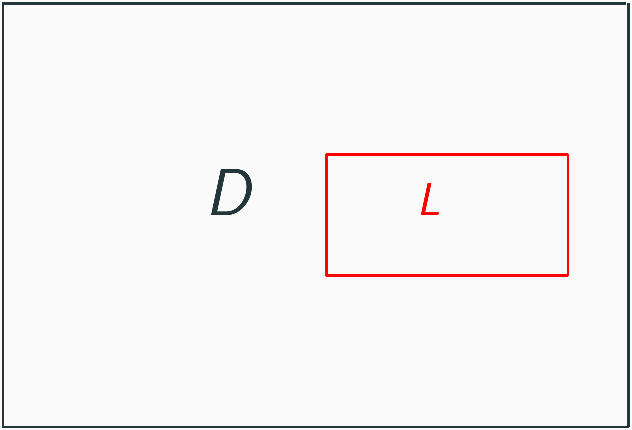
Cyclic Extensions

An extension of D is *cyclic* if all its (non-sprout) vertices are reachable from (roots of) D in the composition

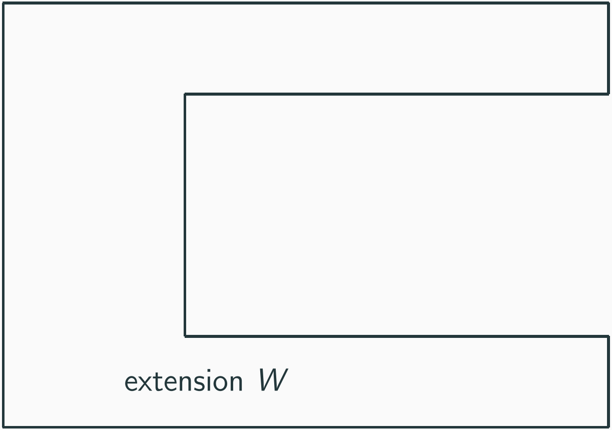


Matching

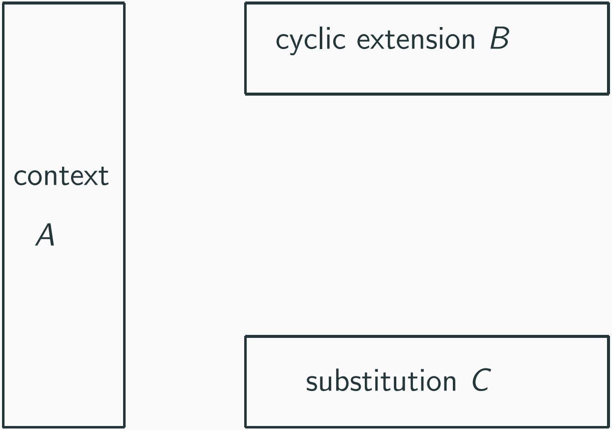
Extension Decomposition: $D = A\zeta(B\theta C)$



Extension Decomposition: $D = A\zeta(B\theta C)$



Extension Decomposition: $D = A\zeta(B\theta C)$

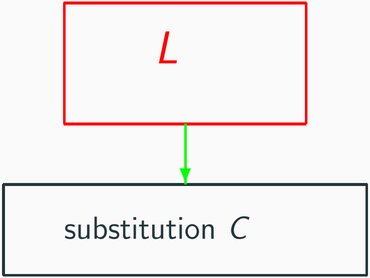


context A

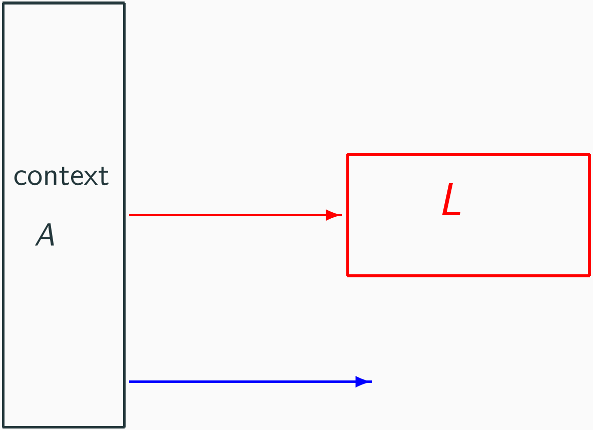
cyclic extension B

substitution C

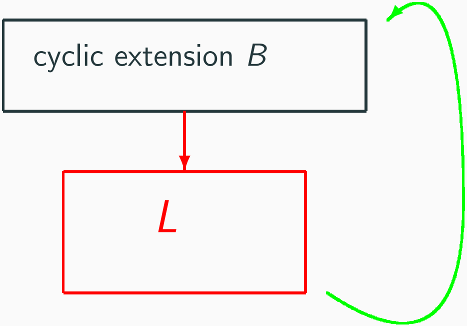
Extension Decomposition: $D = A\zeta(B\theta C)$



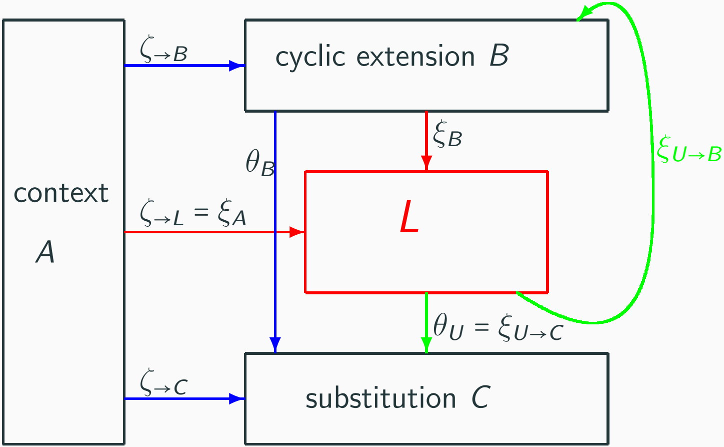
Extension Decomposition: $D = A\zeta(B\theta C)$



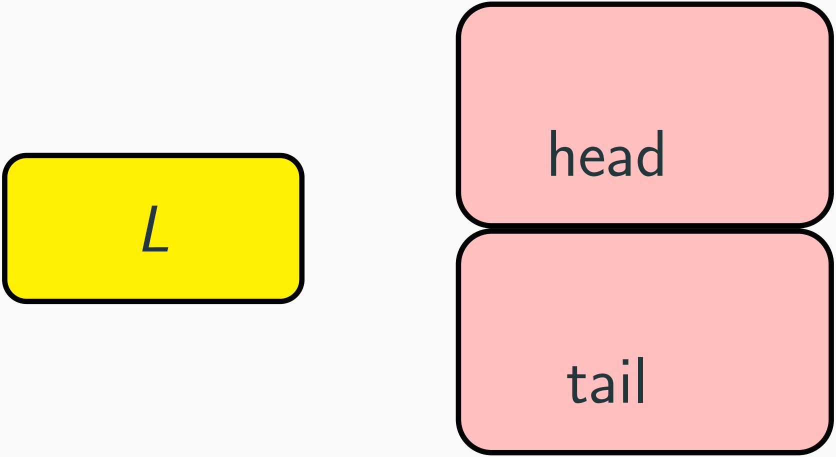
Extension Decomposition: $D = A\zeta(B\theta C)$



Extension Decomposition: $D = A\zeta(B\theta C)$



Drag Matching

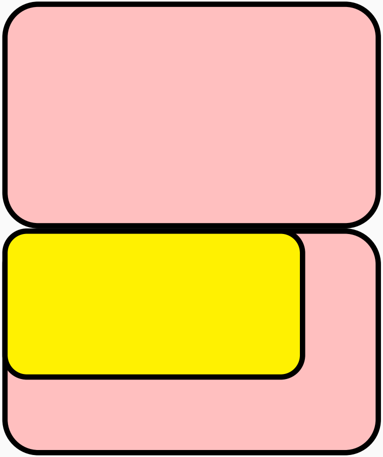


L

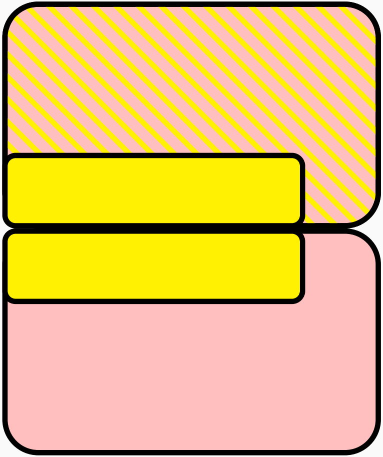
head

tail

Drag Matching: Tail



Drag Matching: Circular Extension



Termination

Graph Rewrite Ordering

A *rewrite ordering* $>$ is

- *well-founded*
- *compatible* with isomorphism
- *monotonic* wrt substitutions and contexts

Termination

Terminates iff there's an ordering $\succ$ such that

$$L \, \xi \, B \succ R \, \xi \, B$$

for all rules $L \to R$ and *cyclic* extensions ξB of L

Graph Path Order (GPO)

Given *head order* $\dot{\succeq}$, $s \succ t$ in GPO if one of

- $\nabla s \succeq t$
- $\widehat{s} \mathrel{\dot{\succ}} \widehat{t}$ and $s \succ \nabla t$
- $\widehat{s} \doteq \widehat{t}$, $\nabla s \succ \nabla t$, $s \succ \nabla t$

GPO Properties

$s \triangleright t$ means t is a tail of s

- $\triangleright$ is well-founded (excluding empty drags)
- Empty is minimal: $D \geq \varnothing$
- Subdrags are smaller: $\triangleright \subseteq \succ$
- $D \succ E \Rightarrow \mathcal{V}ar(D) \supseteq \mathcal{V}ar(E)$
- $\succ$ is transitive

GPO is Good

GPO is. . .

- a rewrite ordering
- well-founded
- total (modulo isomorphism) when the head order is

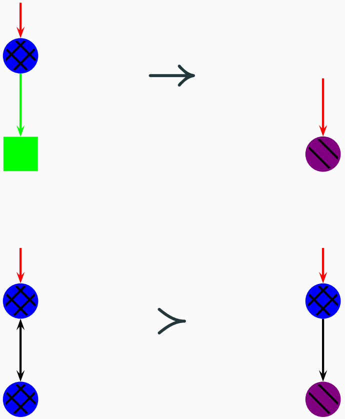
Head Order

Compare V, X

1. compare multiset V of vertices – vis-à-vis precedence $\geq$

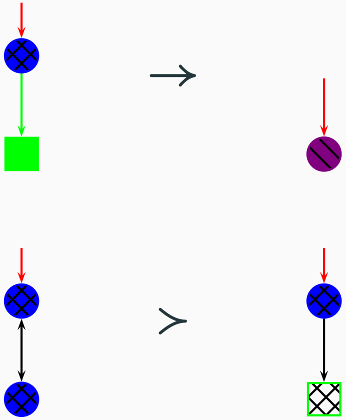
2. compare multiset X of edges – as (lexicographic) pairs of symbols

Example



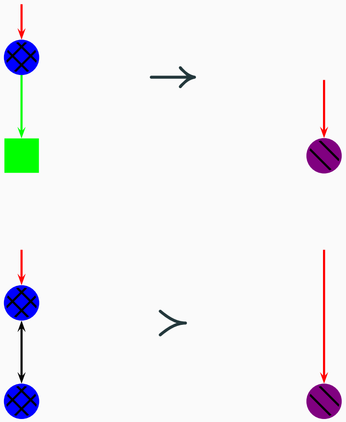
Compare heads. . .

Example



Compare heads. . . : Subdrag

Example

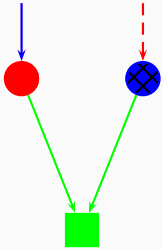


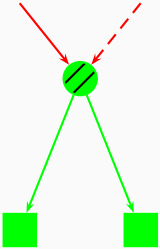
Right tail. . . : Precedence 🔵 > 🟣 does the trick

GPO Decidability

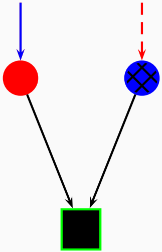
Decidable whether rules terminate under GPO – provided $\forall$ fragment of head-order constraints is decidable

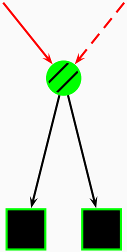
Comparing Extensions



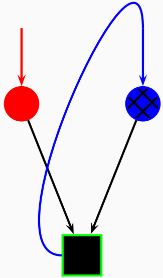


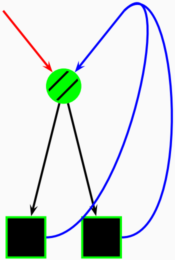
Comparing Extensions



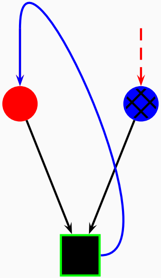


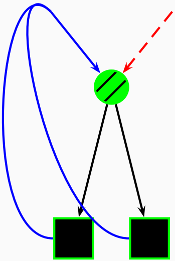
Comparing Extensions



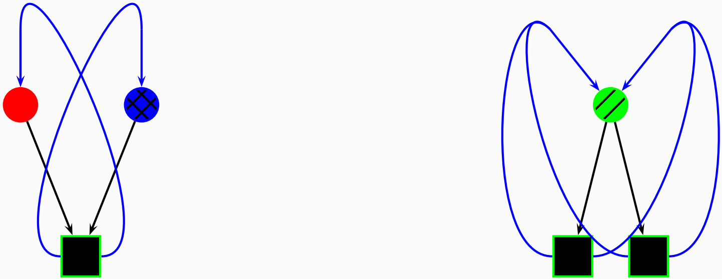


Comparing Extensions

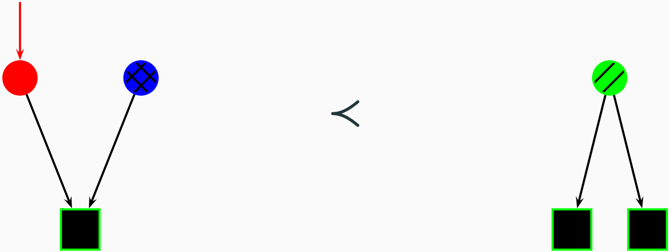




Comparing Extensions



Comparing Extensions



🔴 , 🔵 < 🟢

■ < ■ ∪ ■