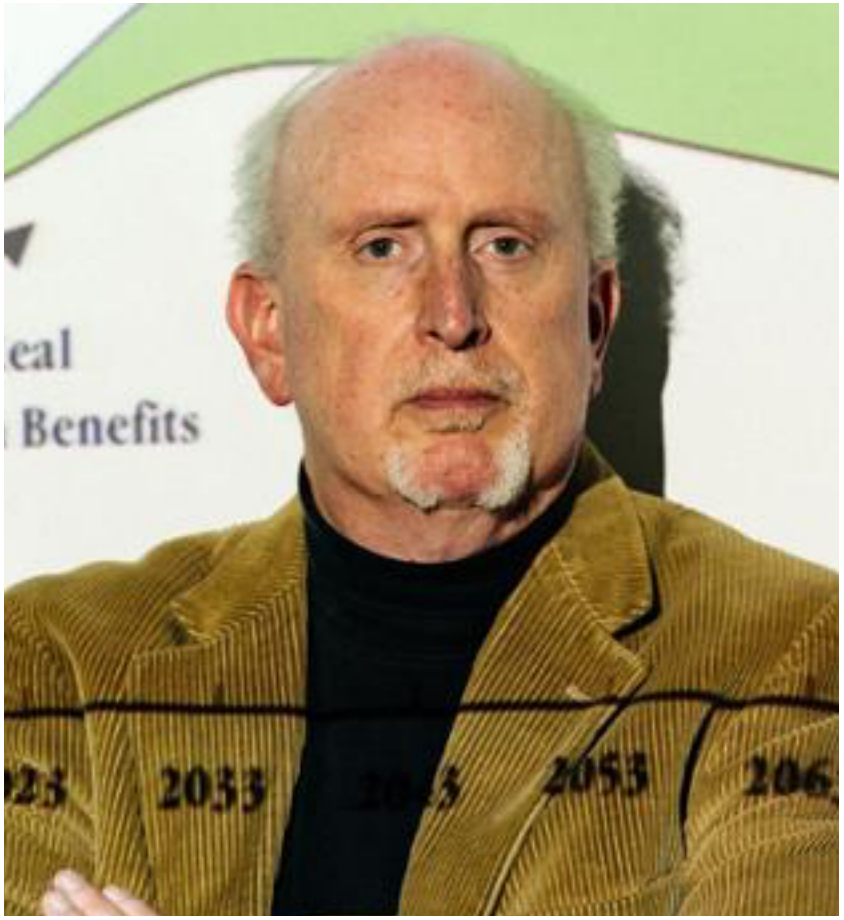
Church-Rosser



Alonzo Church
1903–1995
President of the American Philosophical Association, Pacific Division
1973–1974
University of California at Los Angeles, Los Angeles, California

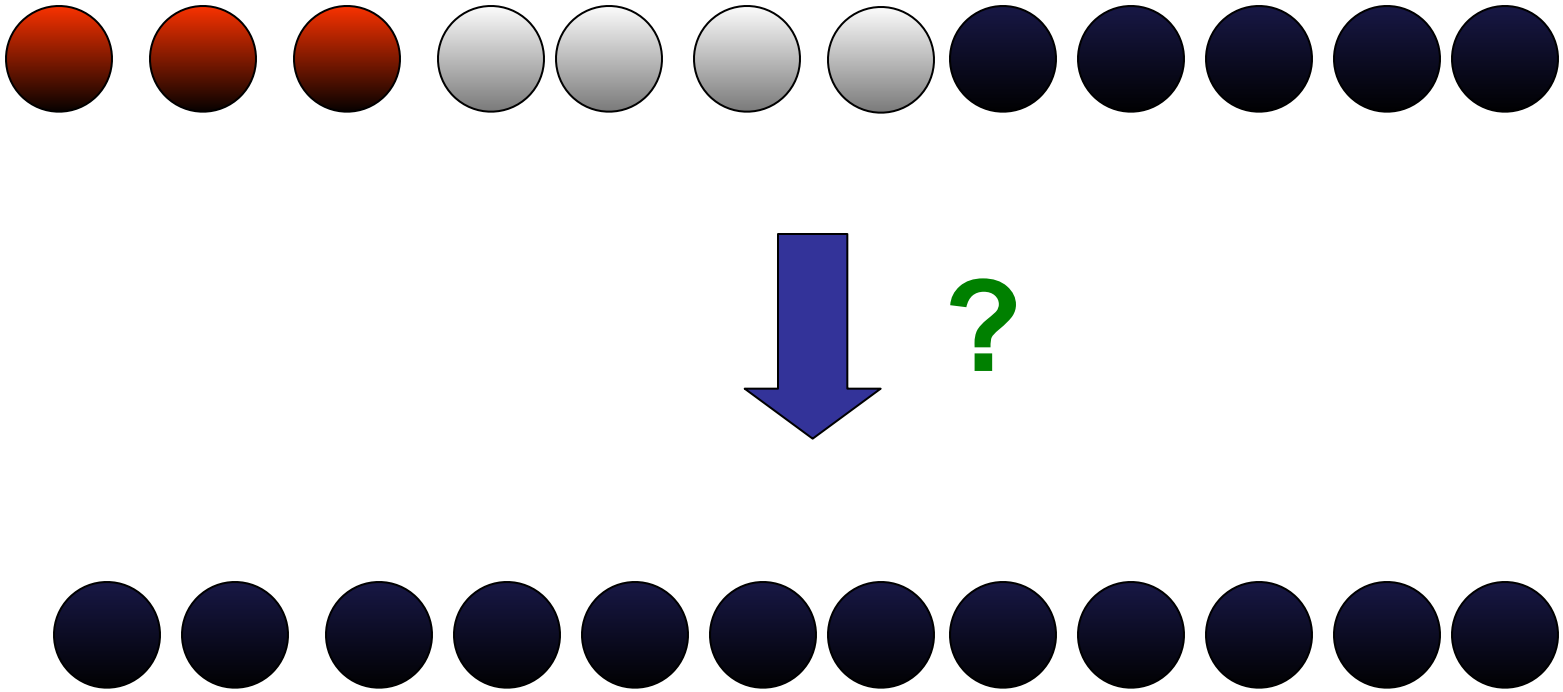
Photo courtesy of Anthony Ralston, Department of Computer Science
State University of New York at Buffalo



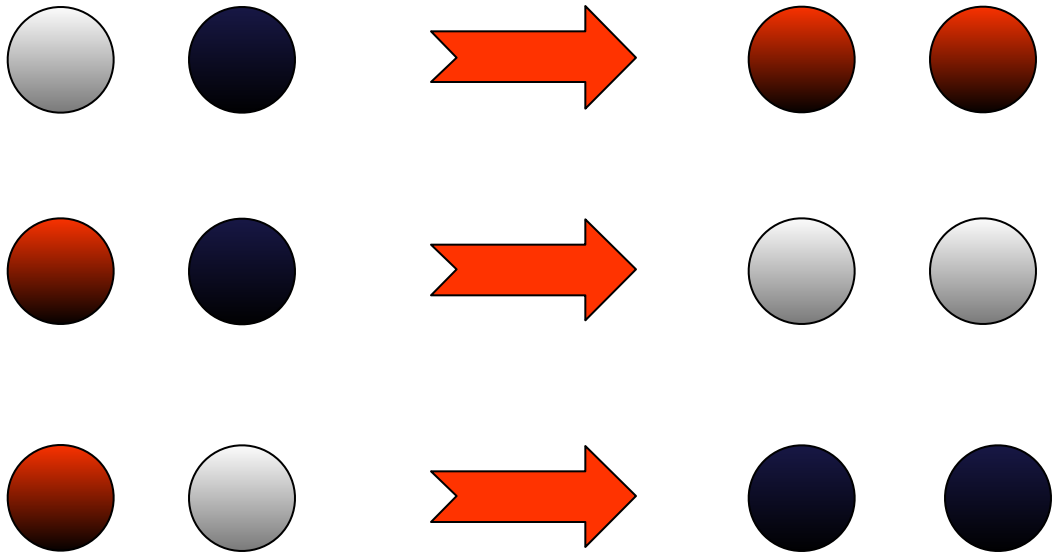
J. Barkley Rosser, 81, a professor emeritus of mathematics and computer science at the University of Wisconsin who had served in government, died Sept. 5, 1989. Along with Alan Turing and John von Neumann, his 1930s work in mathematical logic laid the foundation for modern computer science theory. Dr. Rosser did mathematics work for the space program and served on the National Security Agency's scientific advisory board







Chamelion Problem



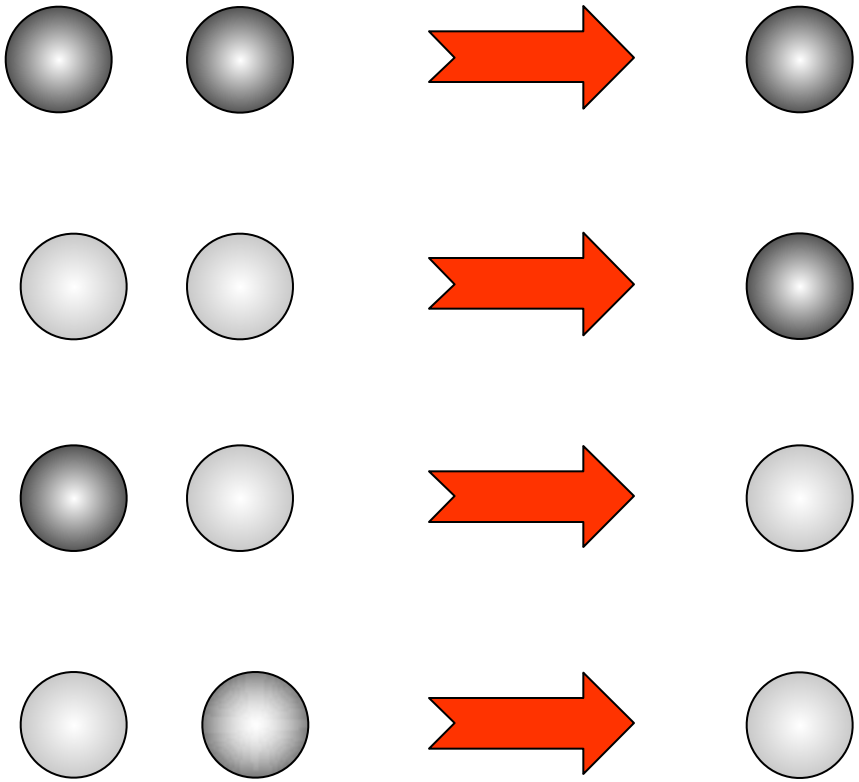
Grecian Urn



Urn Problem



Urn System



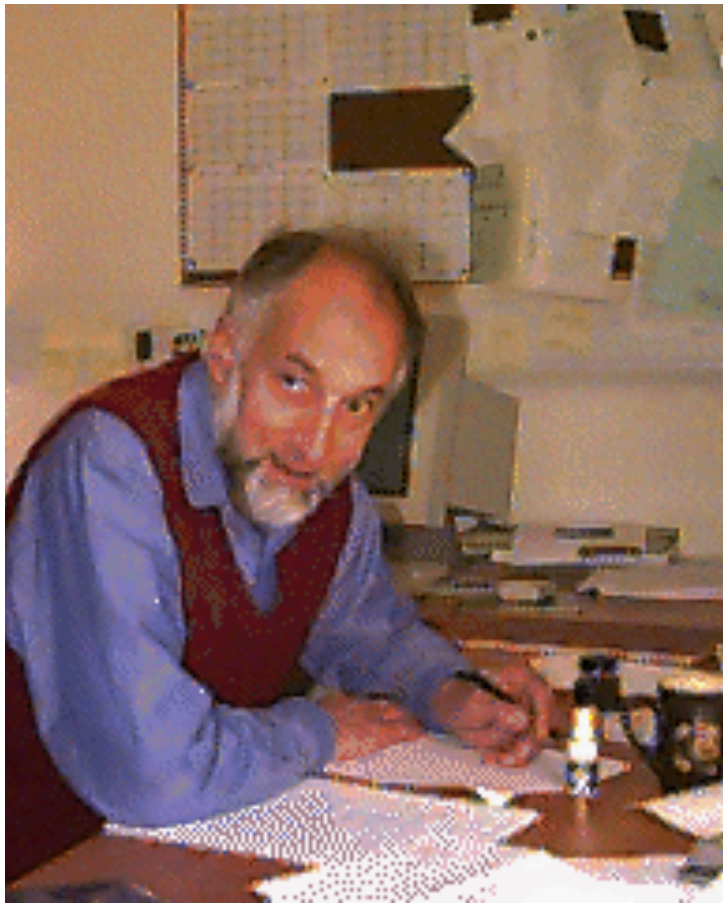
Rewriting

- Lambda calculi
- Recursive functions
- Functional languages
- Logic languages
- Symbolic computation

Functional Programming Languages

- Lisp, Scheme
- APL, FP, M
- ML, OCaml
- SASL, Miranda
- Haskell, Curry
- F#

ML



- Robin Milner (1934-2010)
- Logic for Computable Functions

  Stanford & Edinburgh
  1972-1995
- Meta-Language
  - Theorem proving
  - Type system
  - Higher-order functions

ML Examples

- fun length nil = 0
  | length (x::s) = 1 + length(s);

- fun append(nil, ys) = ys
  | append (x::xs, ys) = x :: append (xs, ys);

- fun reverse nil = nil
  | reverse (x::xs) =
  append ((reverse xs), [x]);

Abstract Rewriting

An abstract rewriting system is composed of

- elements T
- binary relation $\rightarrow \subseteq T \times T$
  - there may be several relations $\rightarrow$, $\rightarrow$, ...
  - **labelled transition system**

Properties

- Termination (always halts)
  - Normalization (sometimes halts)
    - Has at least one normal form
- Confluence (can always recover)
  - Unique normalization
    - Has at most one normal form
- Order of evaluation may matter!

Normal Form

- Descendent — reachable
- Normal form — cannot be rewritten
- An object's nf is any descendent nf
- Ambiguous — has more than one nf
- Weakly normalizing — at least one nf
- Immortal — has a non-halting path
- Strongly normalizing — mortal (not immortal)

Why Nontermination?

- Full computability, including interpreters
  - A nontrivial PL with only halting programs cannot interpret itself
    - c(x) := I(x,x)+1
- Streams
    - ones := 1 :: ones
    - map f (x::s) := f(x) :: map f s
    - int := 1 :: map +1 int

Interpreter

$$eval(\mathbf{zap0}, \langle x,\ y\rangle) \rightarrow \langle 0,\ y\rangle \qquad eval(\mathbf{zap1}, \langle x,\ y\rangle) \rightarrow \langle x,\ 0\rangle$$

$$eval(\mathbf{inc0}, \langle x,\ y\rangle) \rightarrow \langle s(x),\ y\rangle \qquad eval(\mathbf{inc1}, \langle x,\ y\rangle) \rightarrow \langle x,\ s(y)\rangle$$

$$eval(\mathbf{dec0}, \langle 0,\ y\rangle) \rightarrow \langle 0,\ y\rangle \qquad eval(\mathbf{dec1}, \langle x,\ 0\rangle) \rightarrow \langle x,\ 0\rangle$$

$$eval(\mathbf{dec0}, \langle s(x),\ y\rangle) \rightarrow \langle x,\ y\rangle \qquad eval(\mathbf{dec1}, \langle x,\ s(y)\rangle) \rightarrow \langle x,\ y\rangle$$

$$eval(\mathbf{ifpos0}\ p, \langle 0,\ y\rangle) \rightarrow \langle 0,\ y\rangle \qquad eval(\mathbf{ifpos1}\ p, \langle x,\ 0\rangle) \rightarrow \langle x,\ 0\rangle$$

$$eval(\mathbf{ifpos0}\ p, \langle s(x),\ y\rangle) \rightarrow eval(p, \langle s(x),\ y\rangle)$$

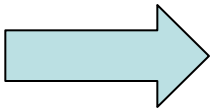
$$eval(\mathbf{ifpos1}\ p, \langle x,\ s(y)\rangle) \rightarrow eval(p, \langle x,\ s(y)\rangle)$$

$$\mathbf{whilepos0}\ p \rightarrow (\mathbf{ifpos0}\ p;\ \mathbf{whilepos0}\ p)$$

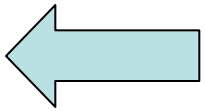
$$\mathbf{whilepos1}\ p \rightarrow (\mathbf{ifpos1}\ p;\ \mathbf{whilepos1}\ p)$$

$$eval((p;\ q), u) \rightarrow eval(q, eval(p, u))$$

Notation



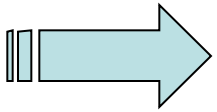
Binary relation



Inverse

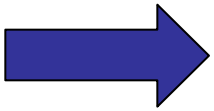


Symmetric closure



Reflexive-Transitive closure (*)

Notation

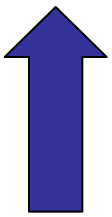


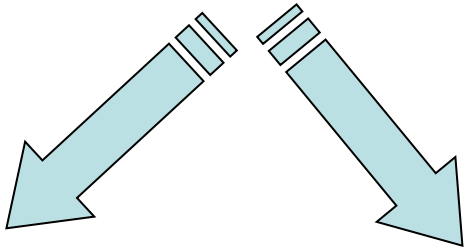
Given $\longrightarrow$



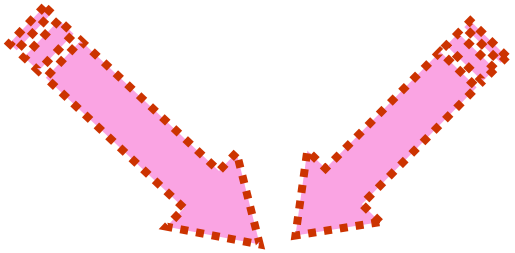
Exists $\dashrightarrow$

Notation

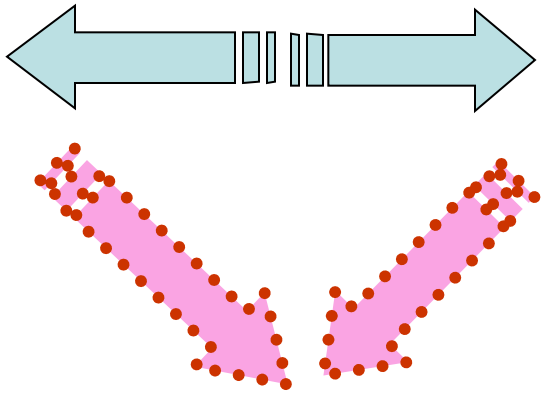




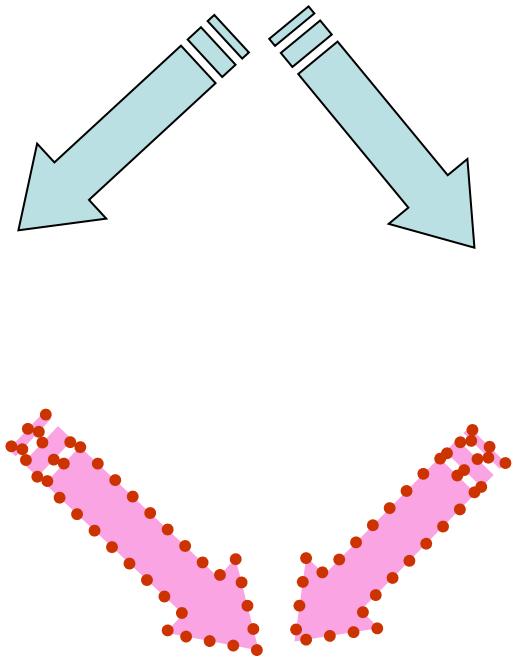




Church-Rosser

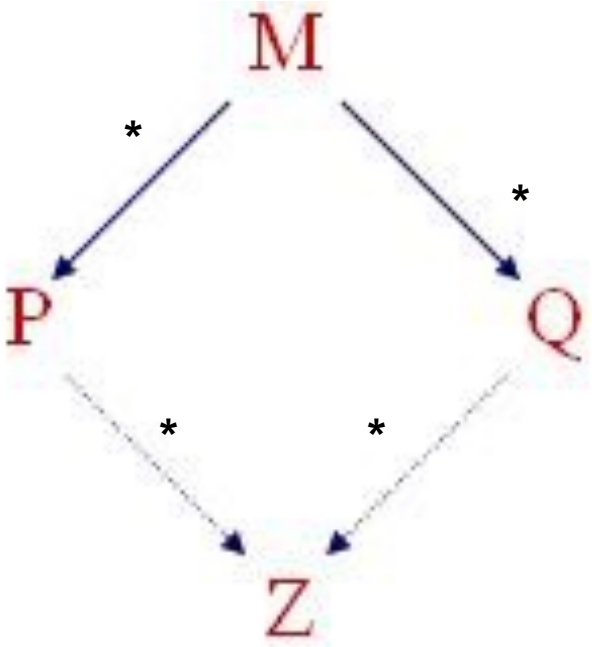


Confluence



Confluence

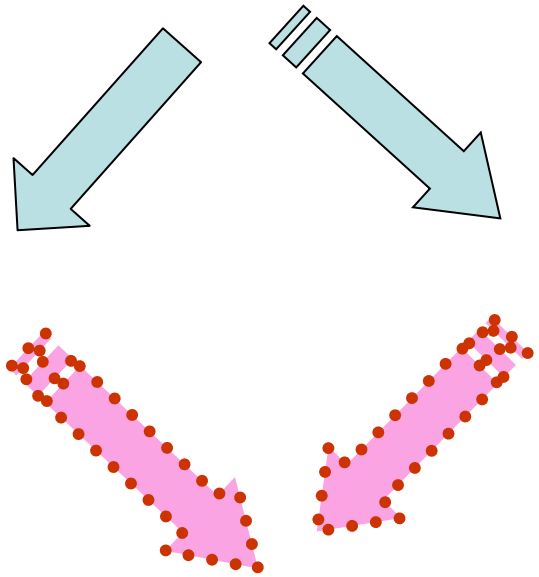
for all M,P,Q



there exists Z s.t.

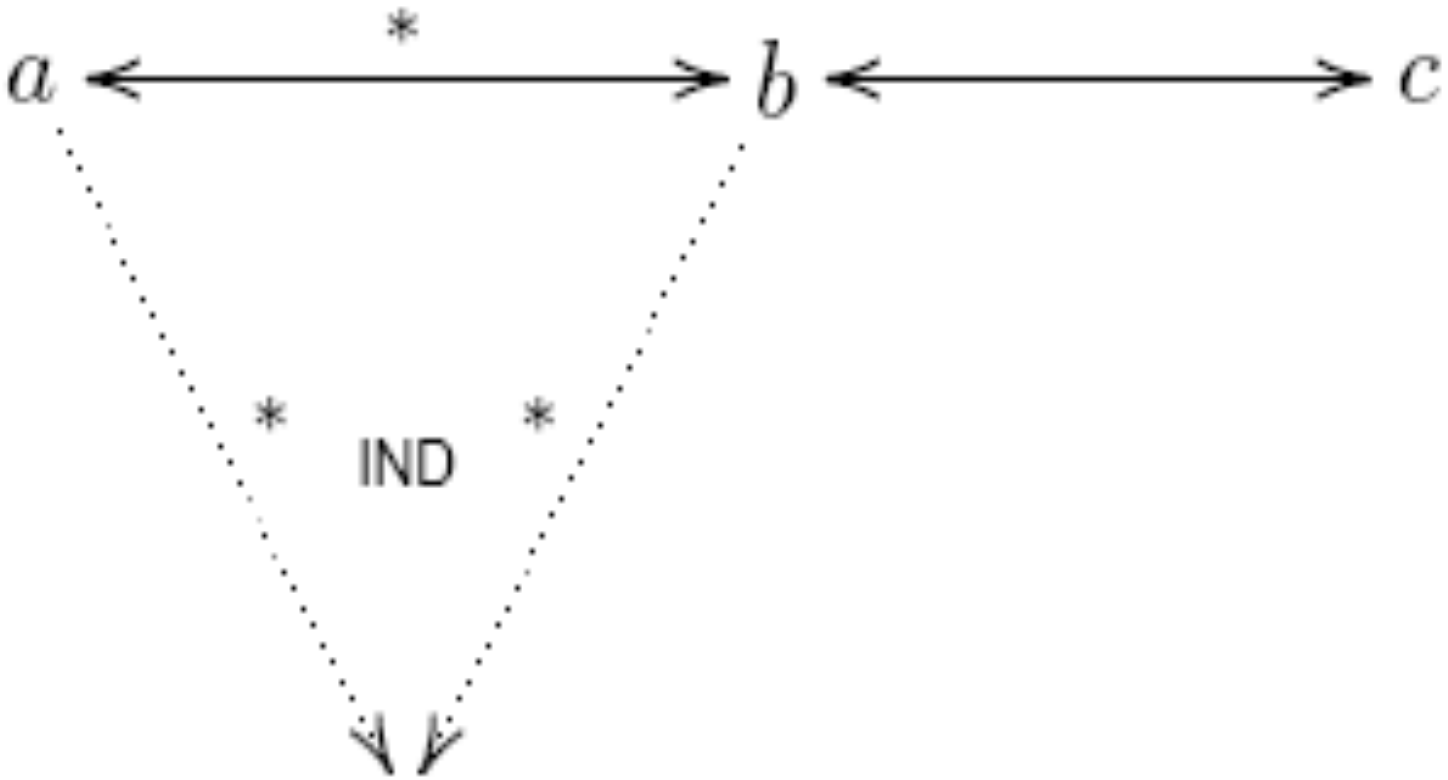
Confluence = Church-Rosser

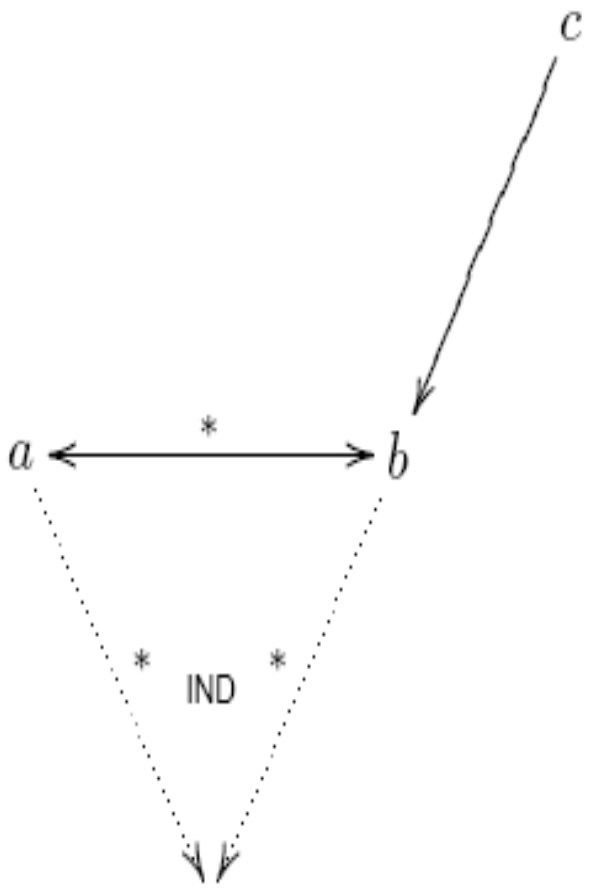
Semi-Confluence

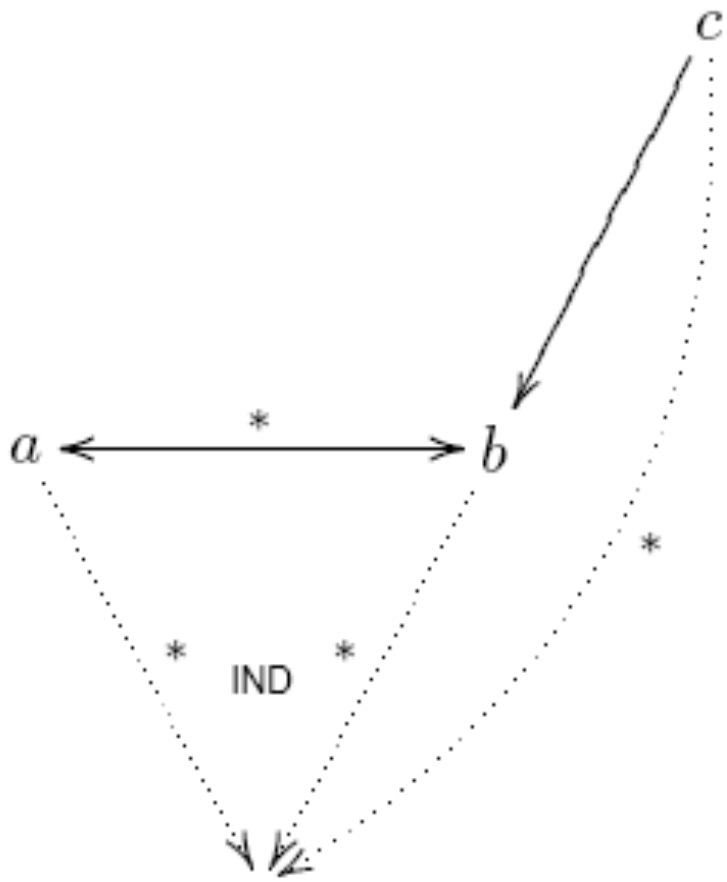


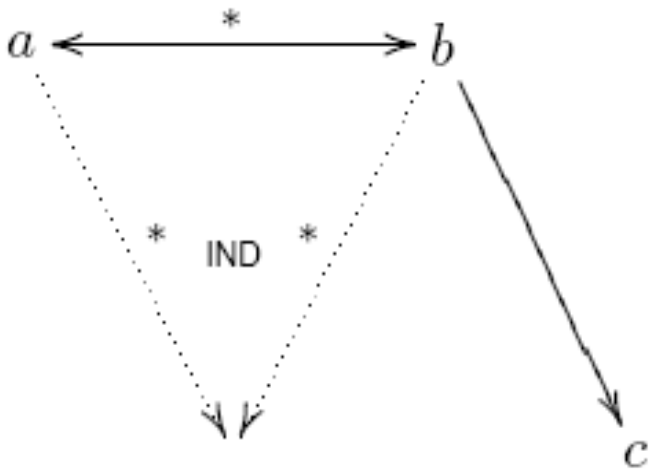
Semi-Confluence = Confluence

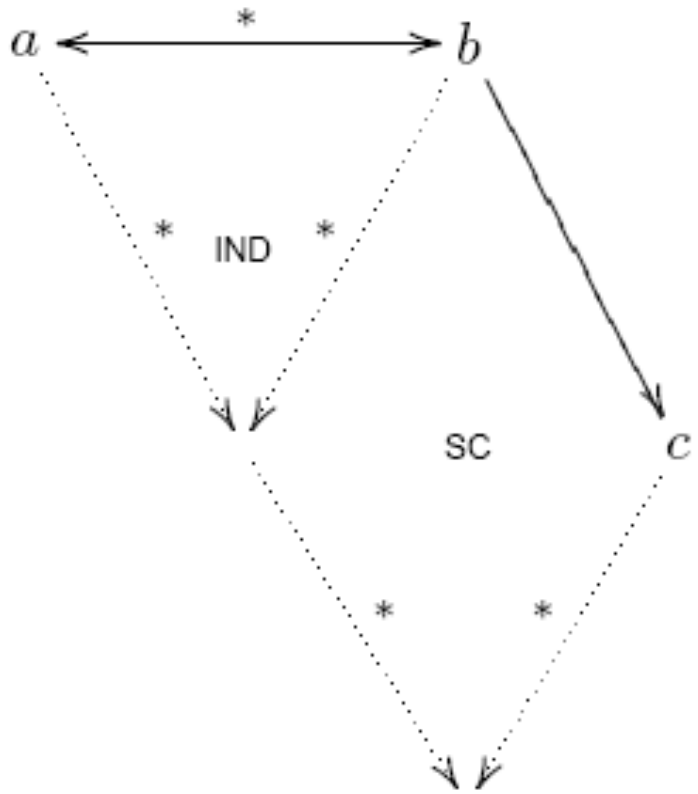
1C ⇨ CR



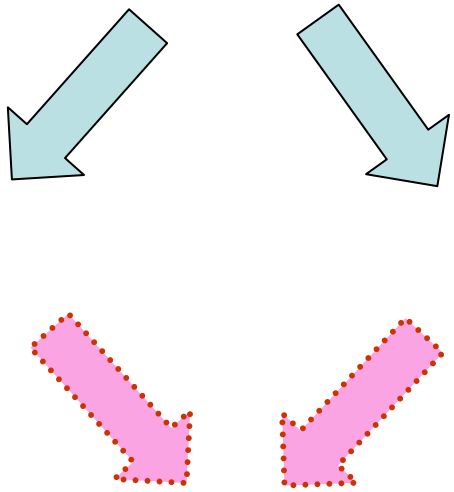

c
a
*
b
*
*
IND
*


c
a
*
b
*
*
IND
*

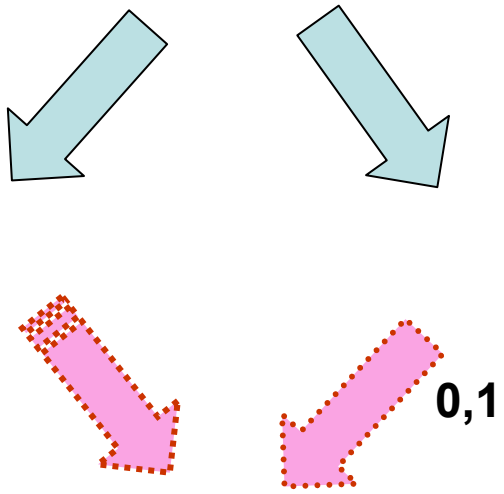

a
*
b
*
IND
*
c


a
*
b
*
IND
*
SC
c
*
*

Very Strong Confluence

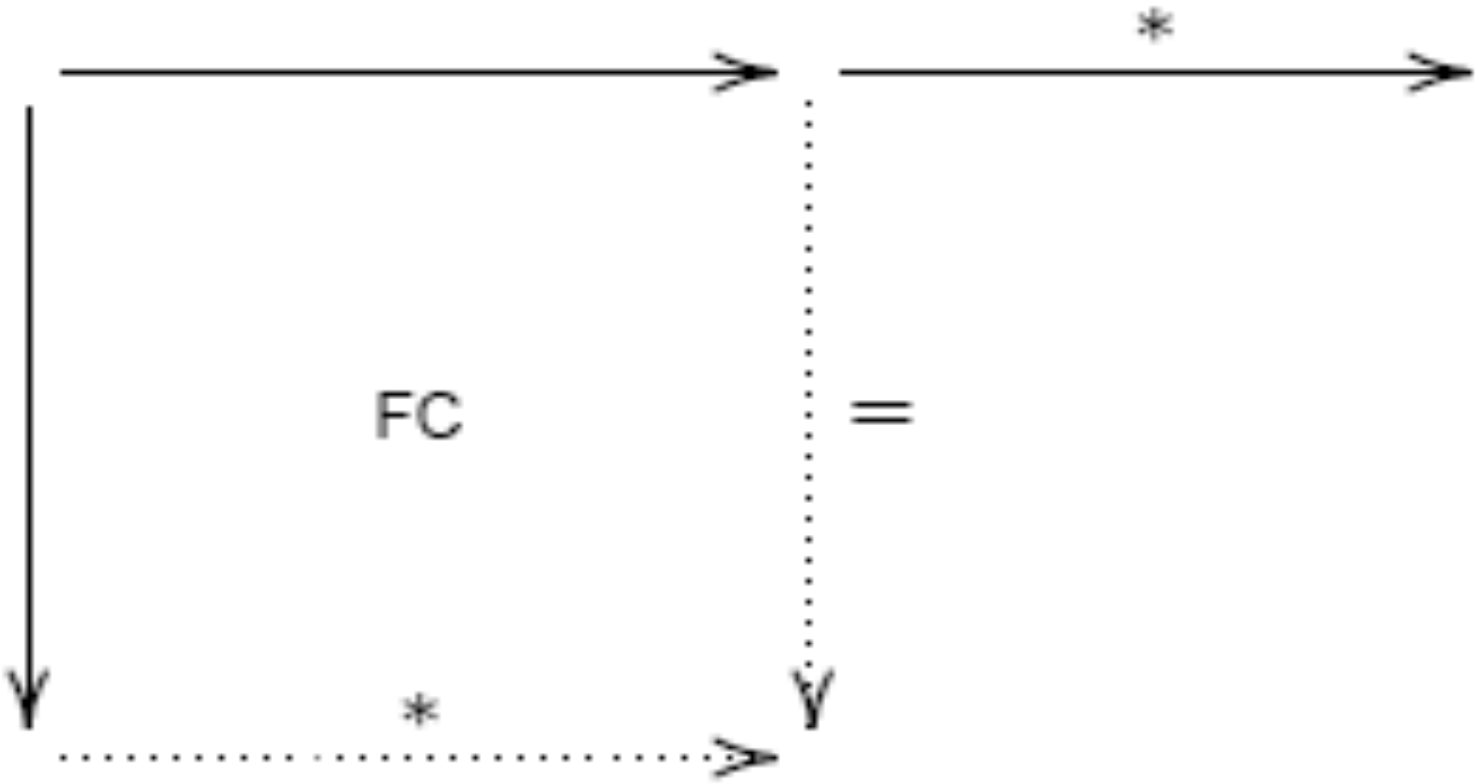


Strong Confluence

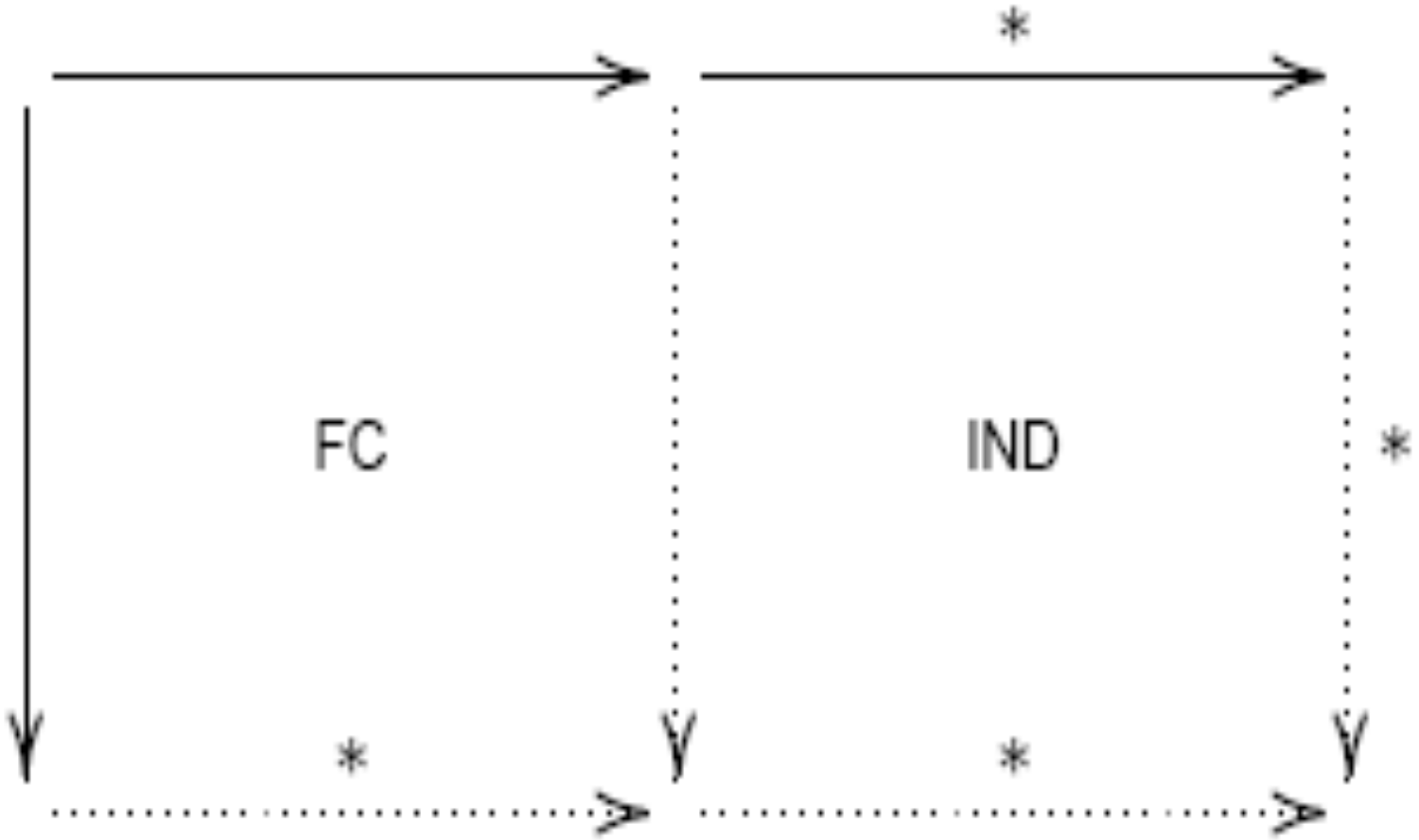


Strong Confluence = Semi-Confluence

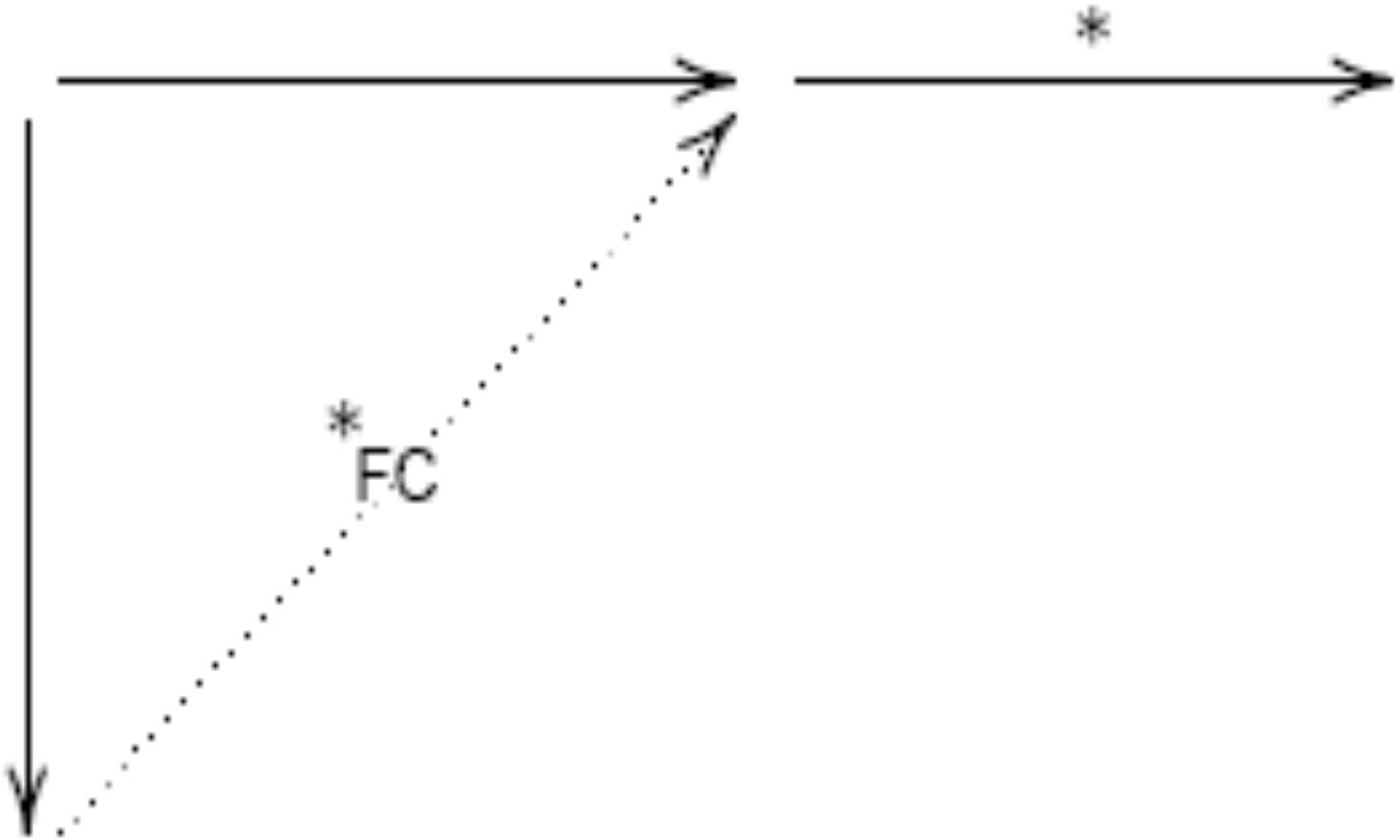
Strong Confluence



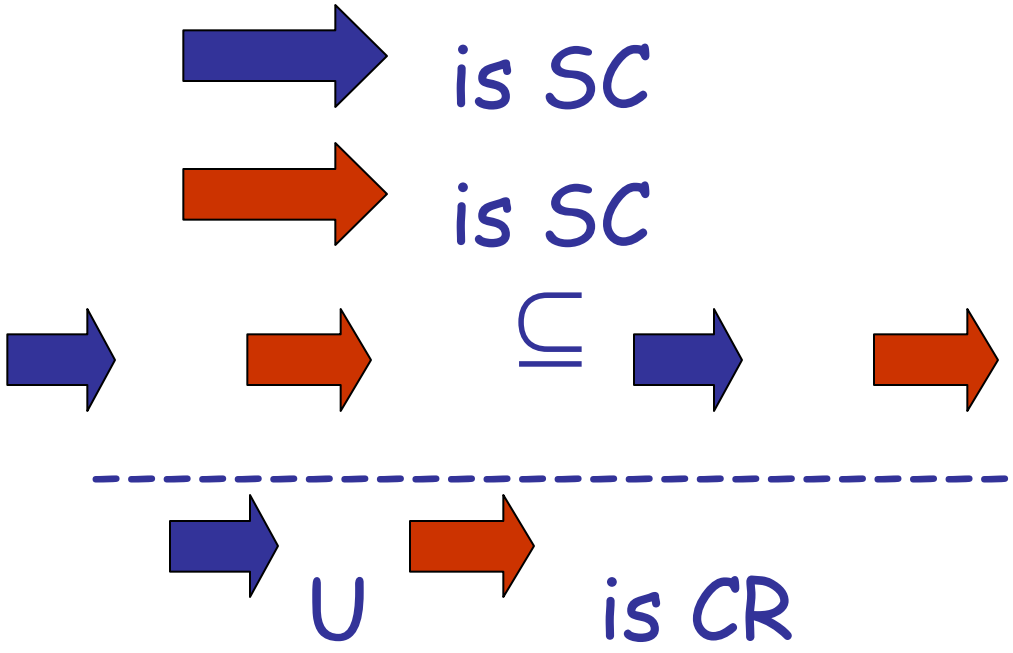
Strong Confluence



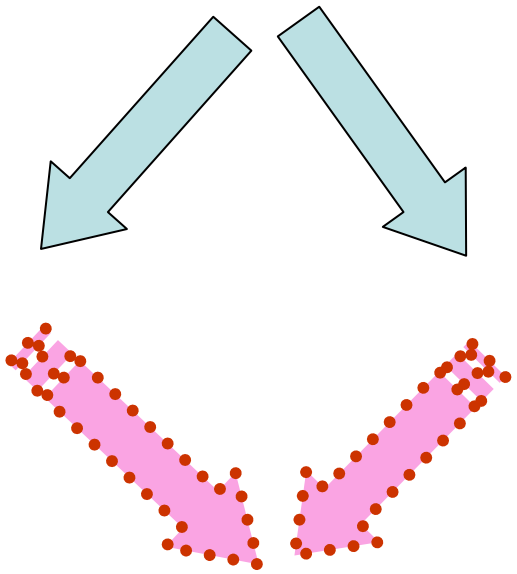
Strong Confluence



Hindley-Rosen Lemma

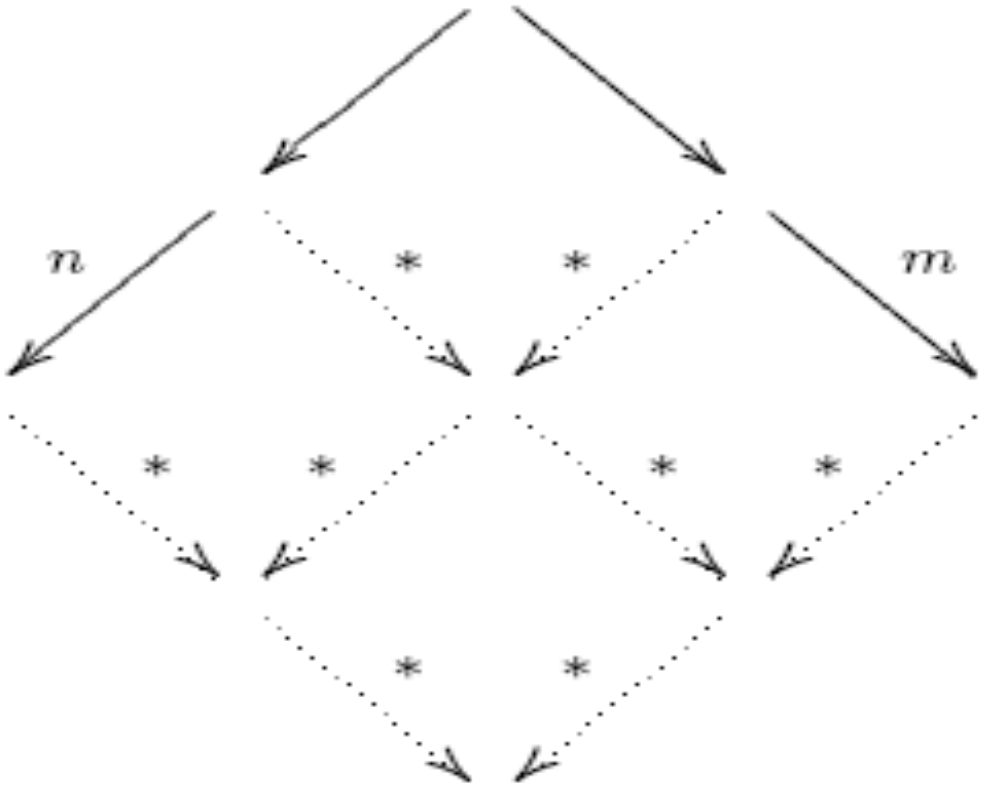


Local Confluence



Question

$LC \Rightarrow CR$

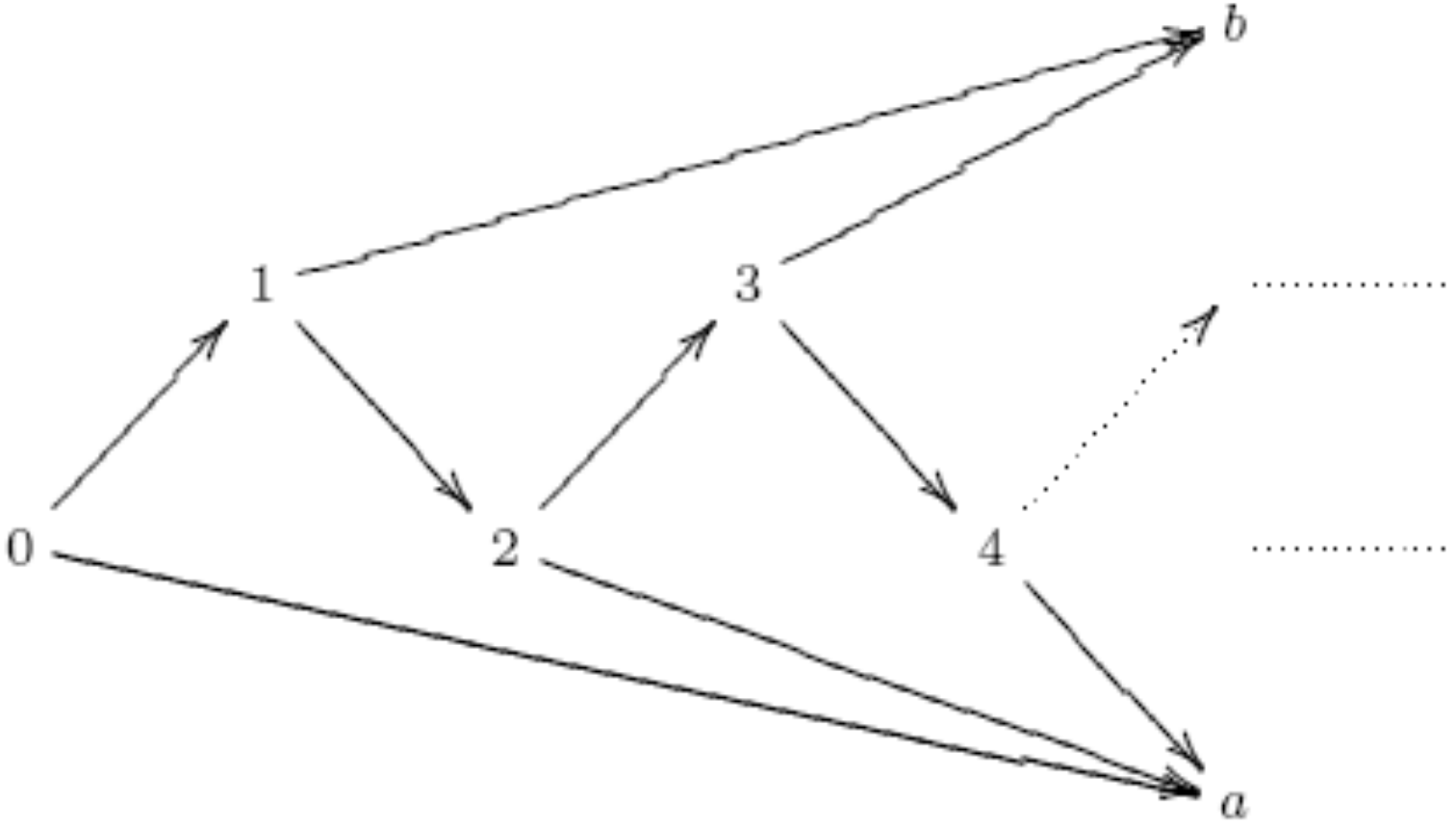

n
m



Counterexample [Curry]



Counterexample



Newman's Lemma

If

terminating (SN),

then

local confluence (LC)

implies

(global) confluence (CR)

Newmann's Lemma

$$LC + SN \Rightarrow CR$$

Complete Induction

$$\frac{\forall x.\ \{[\forall\ y.\ x > y \Rightarrow P(y)] \Rightarrow P(x)\}}{\forall x.\ P(x)}$$

Emmy Noether



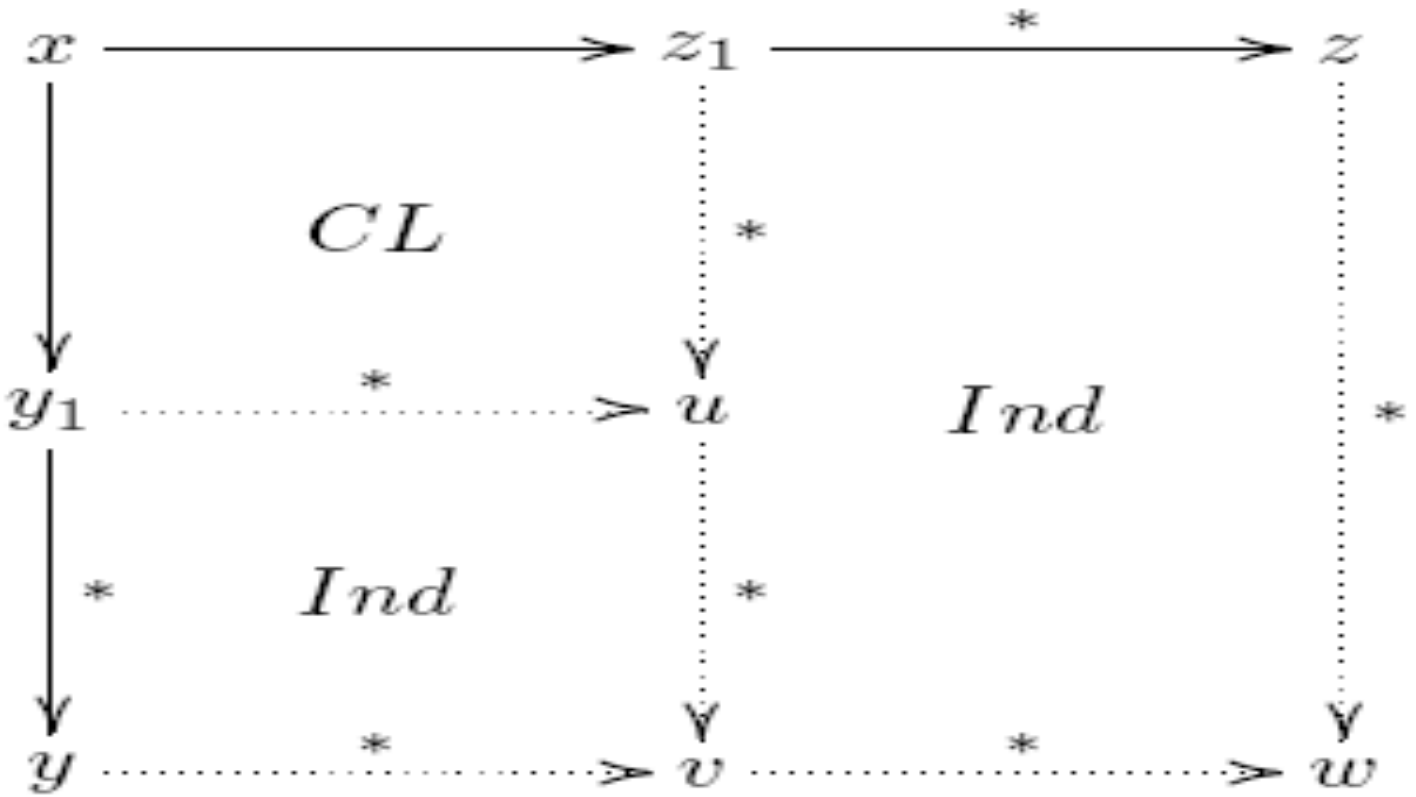
Well-founded Induction

$$\frac{\forall x.\ \{[\forall\ y.\ x > y \Rightarrow P(y)] \Rightarrow P(x)\}}{\forall x.\ P(x)}$$

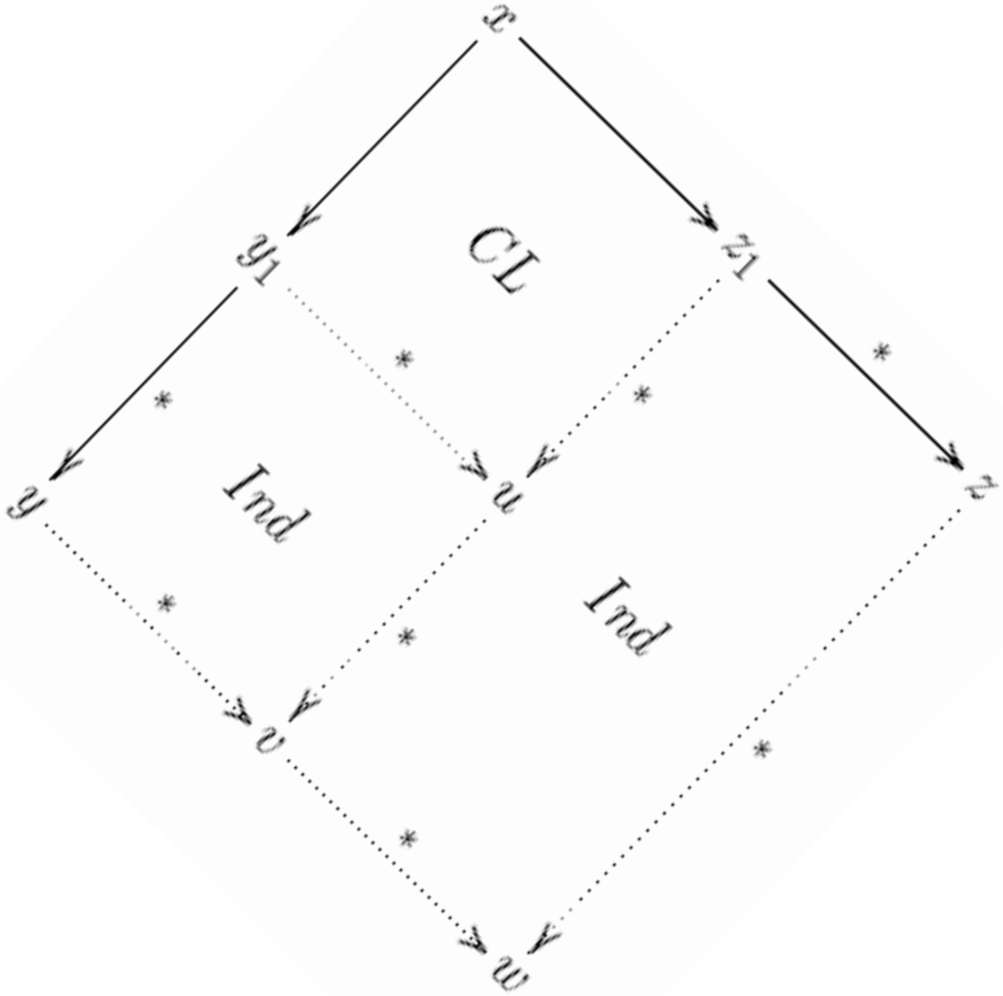
Noetherian Induction

$$\frac{[\forall\, y.\ x \rightarrow y \Rightarrow P(y)] \Rightarrow P(x)}{\forall x.\ P(x)}$$

Proof



Diamond Proof



Orthogonality

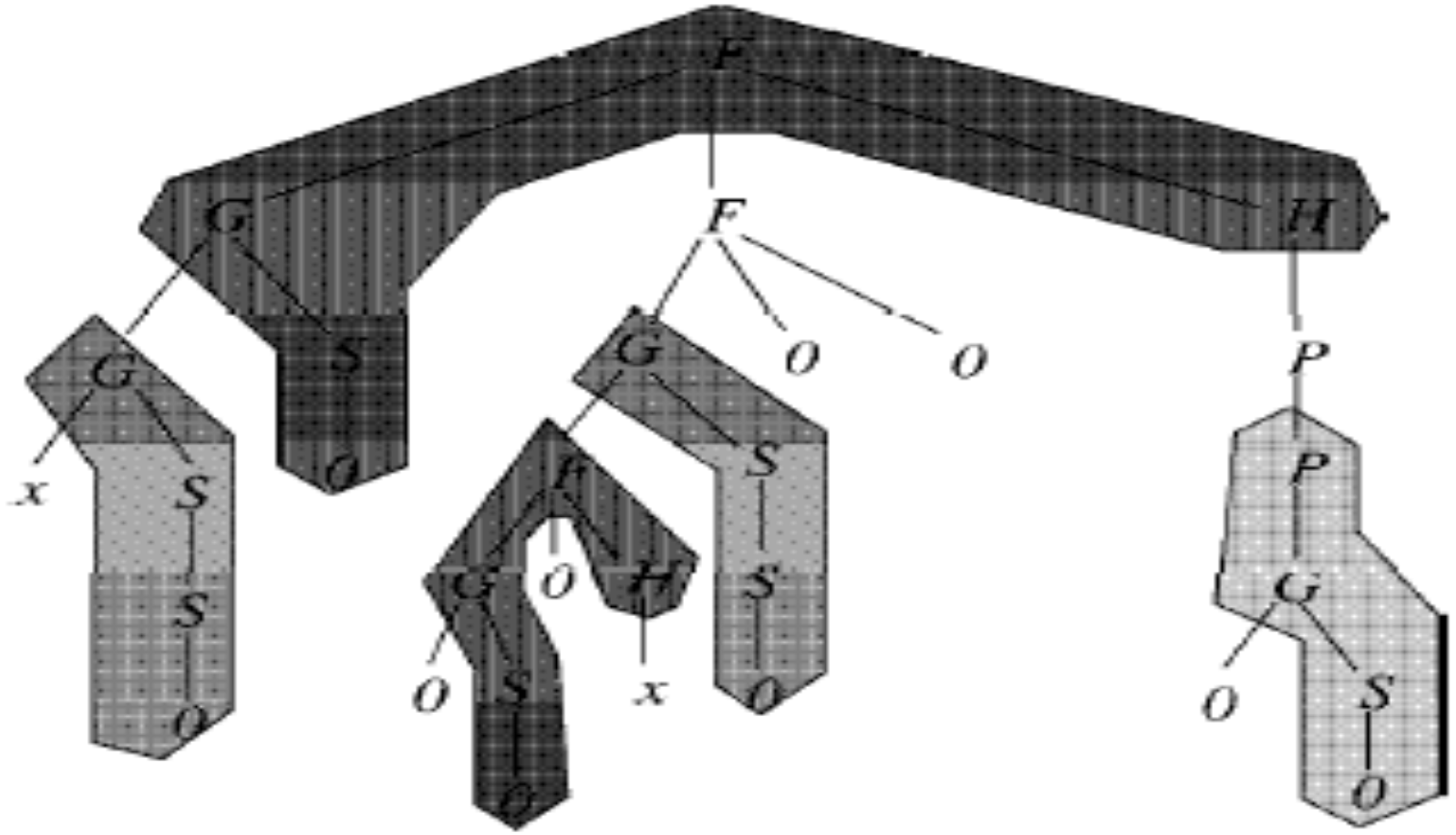
A system is orthogonal if it's left sides are linear and there is no overlap between left sides.

A term is linear if no variable occurs more than once.

A term overlaps another if it unifies with a non-variable subterm.

Example

$$
\begin{aligned}
F(G(x, S(0)), y, H(z)) &\rightarrow x \\
G(x, S(S(0))) &\rightarrow 0 \\
P(G(x, S(0))) &\rightarrow S(0)
\end{aligned}
$$


F
G
F
H
G
S
O
O
P
x
S
O
G
F
S
P
G
O
H
S
G
S
O
O
S
x
O
O
S
O
O
O

Combinatory Logic

$$Ix \rightarrow x$$

$$(Kx)y \rightarrow x$$

$$((Sx)y)z \rightarrow (xz)(yz)$$

Combinatory Logic

$$Ix \to x$$

$$(Kx)y \to x$$

$$((Sx)y)z \to (xz)(yz)$$

$$(Ex)x \to 1$$

$$(Ex)(Sx) \to 0$$

Currying

Treat a function of many parameters as many functions of one parameter

cf. message-passing in object-oriented languages

Orthogonality

- Theory of functional programming
  - Confluence
  - Normalization

Main Theorems

- Orthogonal systems are Church-Rosser.
  - The lambda calculus is CR.

- Outermost rewriting is normalizing.

Notions

- Redex
- Descendant of redex
- A redex that is a descendant of a redex is a residual
- If not, it is created

Counterexample

$$f(x,x) \rightarrow a$$

$$f(x,g(x)) \rightarrow b$$

$$c \rightarrow g(c)$$

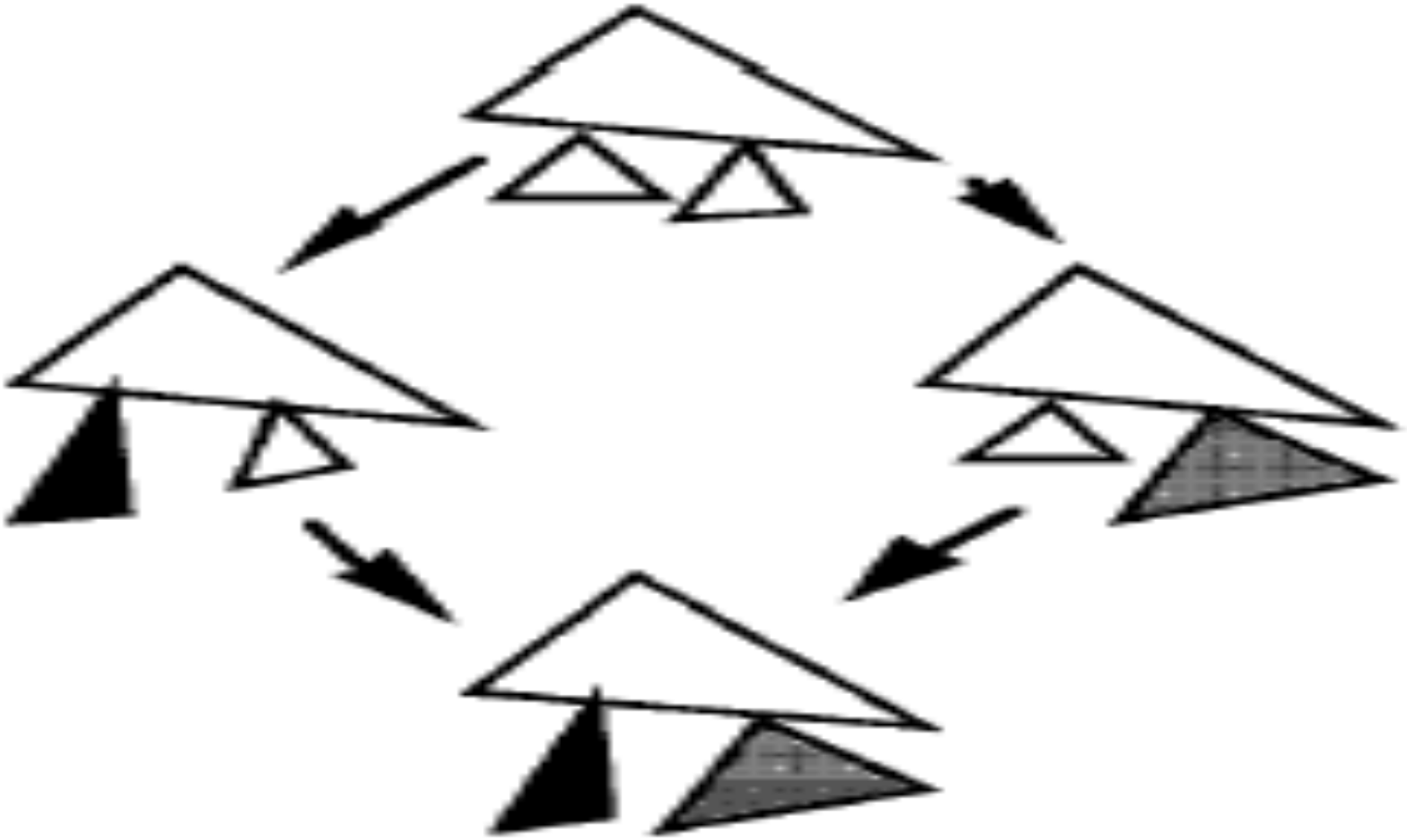
Counterexample

$$f(x,x) \rightarrow a$$

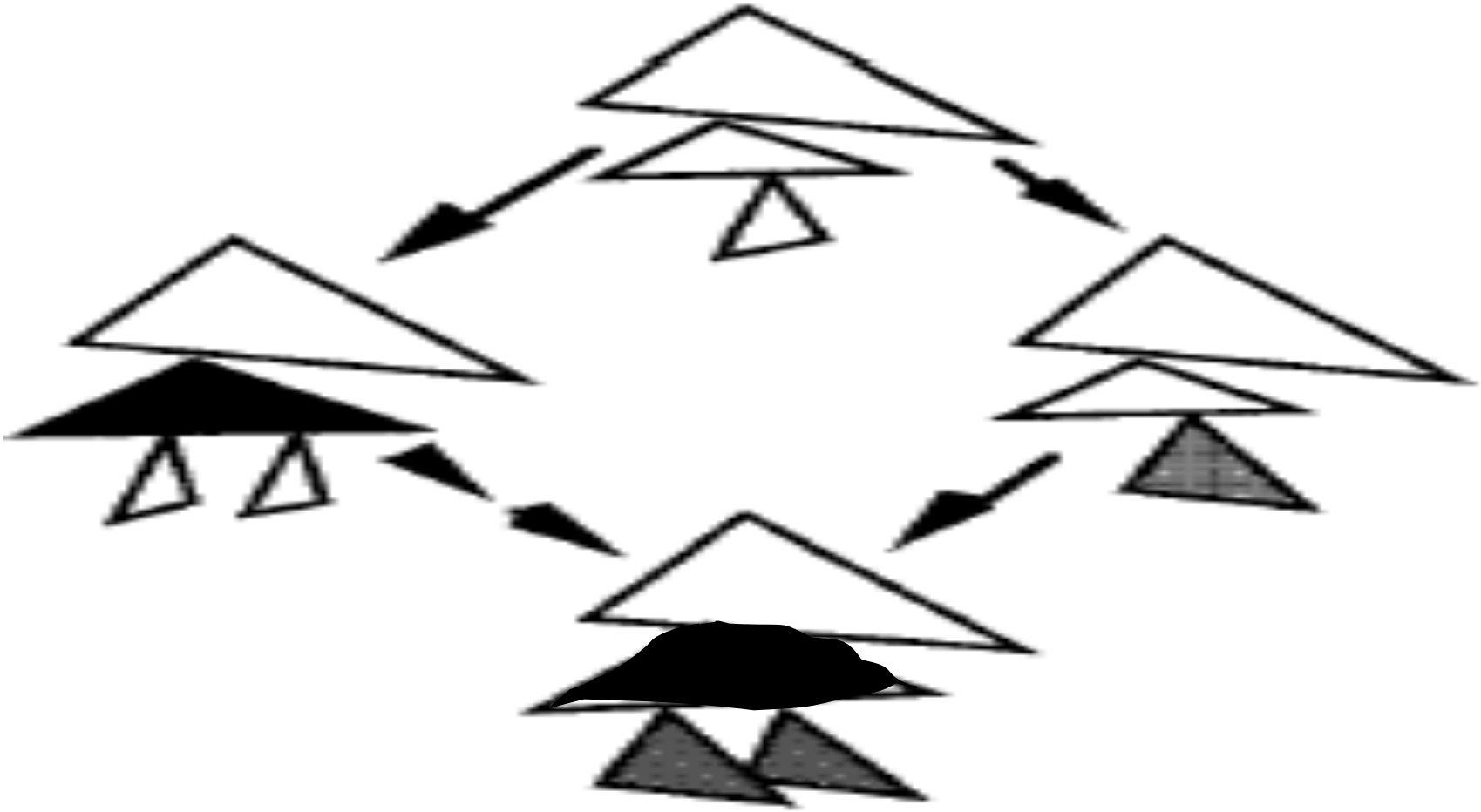
$$g(x) \rightarrow f(x,g(x))$$

$$c \rightarrow g(c)$$

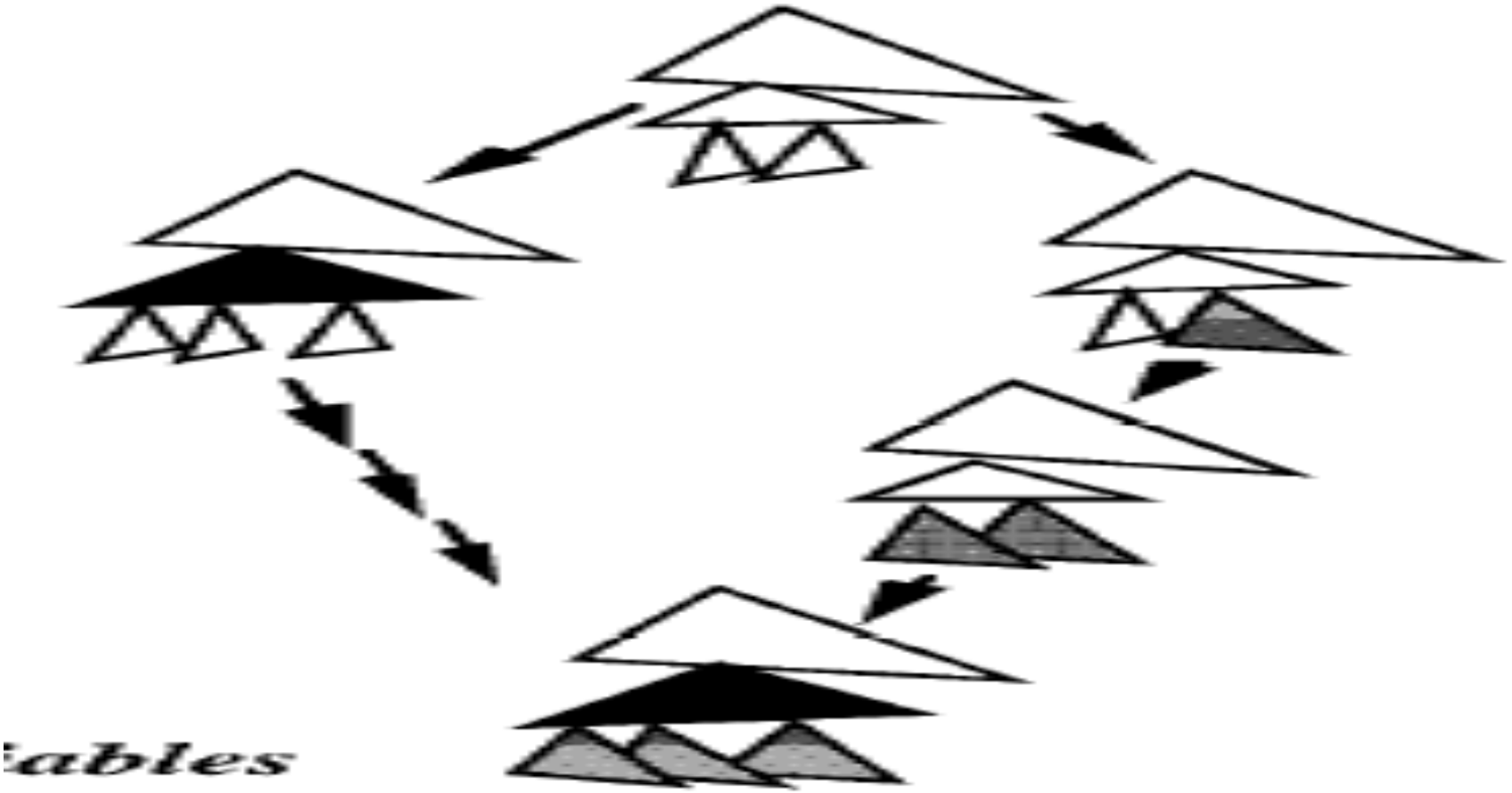
Disjoint Redexes



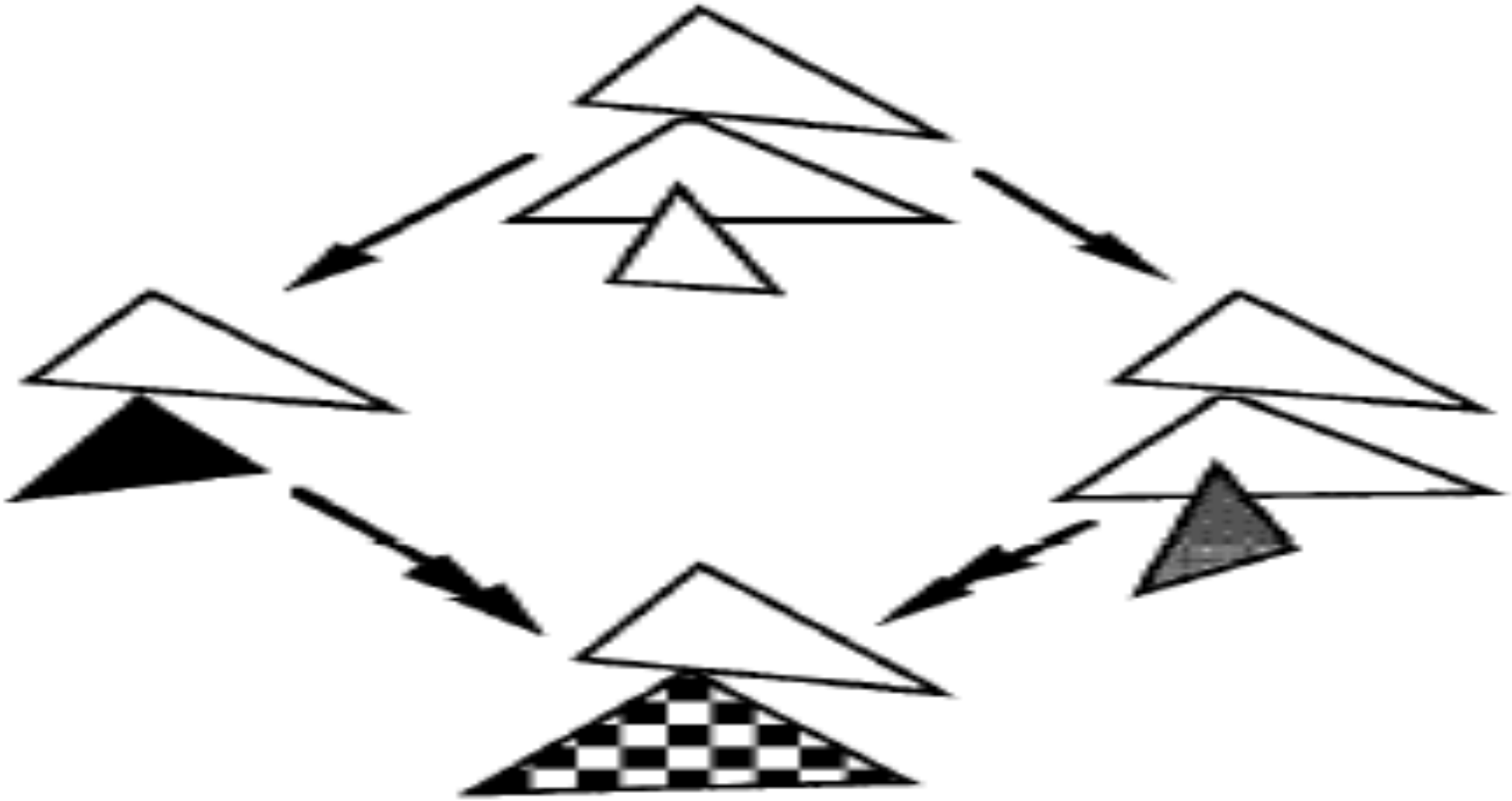
Nested Redexes



Cont'd



Overlapping Redexes



Intermediate Relation

- Suppose $A \subseteq B \subseteq A^*$
- Then

$$\frac{B \text{ CR}}{A \text{ CR}}$$

- In particular

$$\frac{B \text{ SC}}{A \text{ CR}}$$

Multistep Reduction

$$\frac{}{x \to^\circ x}$$

$$\frac{s1 \to^\circ t1, \ldots, sn \to^\circ tn}{f(s1 \ldots sn) \to^\circ f(t1 \ldots tn)}$$

$$\frac{l \to r \in R, \sigma \to^\circ \tau}{l\sigma \to^\circ r\tau}$$

Proof

1. $\rightarrow \subseteq \rightarrow^{\circ} \subseteq \rightarrow^{*}$

2. $\rightarrow^{\circ}$ is SC

3. So $\rightarrow$ CR

$\rightarrow^{\circ}$ is SC

- If you do some,
- then do rest to resolve.

Normalization

- Ambiguous – has more than one nf
- (Weakly) normalizing (WN) – at least one nf

- Strongly normalizing (SN) – not immortal
- Perpetual – if immortal before rewrite, then also after

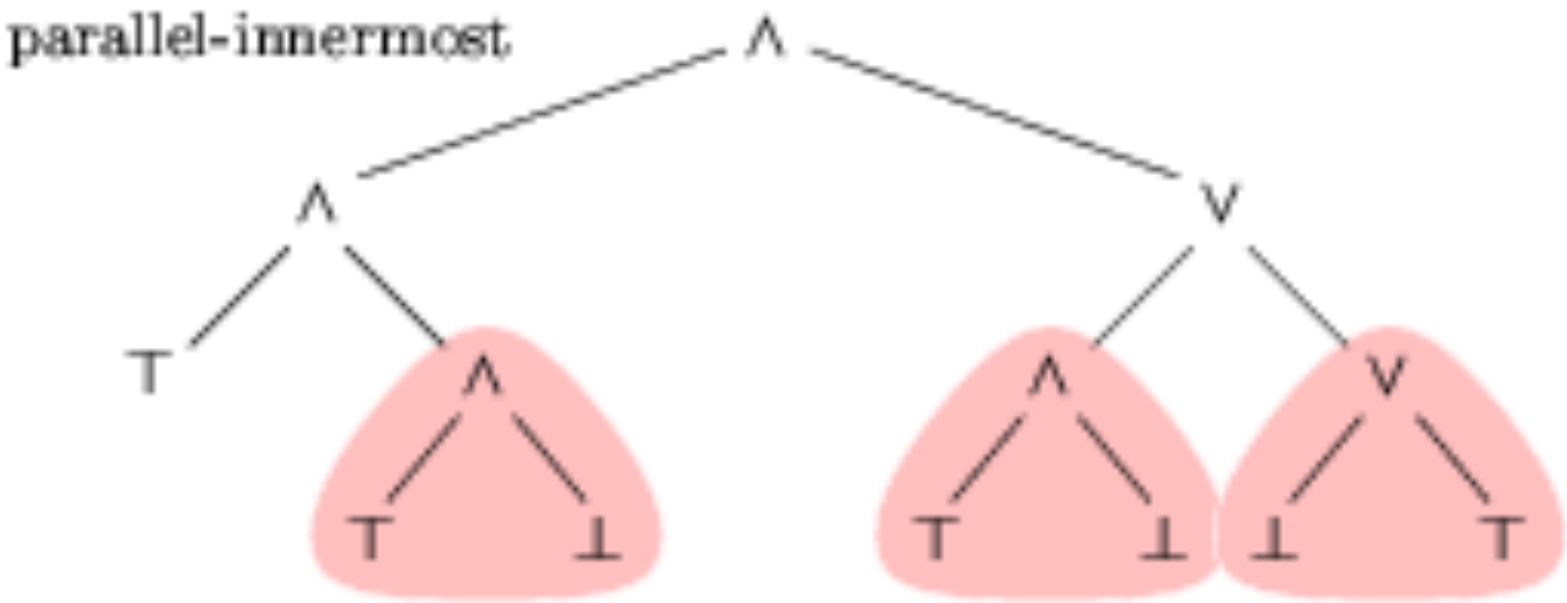
Normalization Strategies

- Normalizing (n) – computes at least one nf
- Perpetual (p) – if immortal before step, then also after

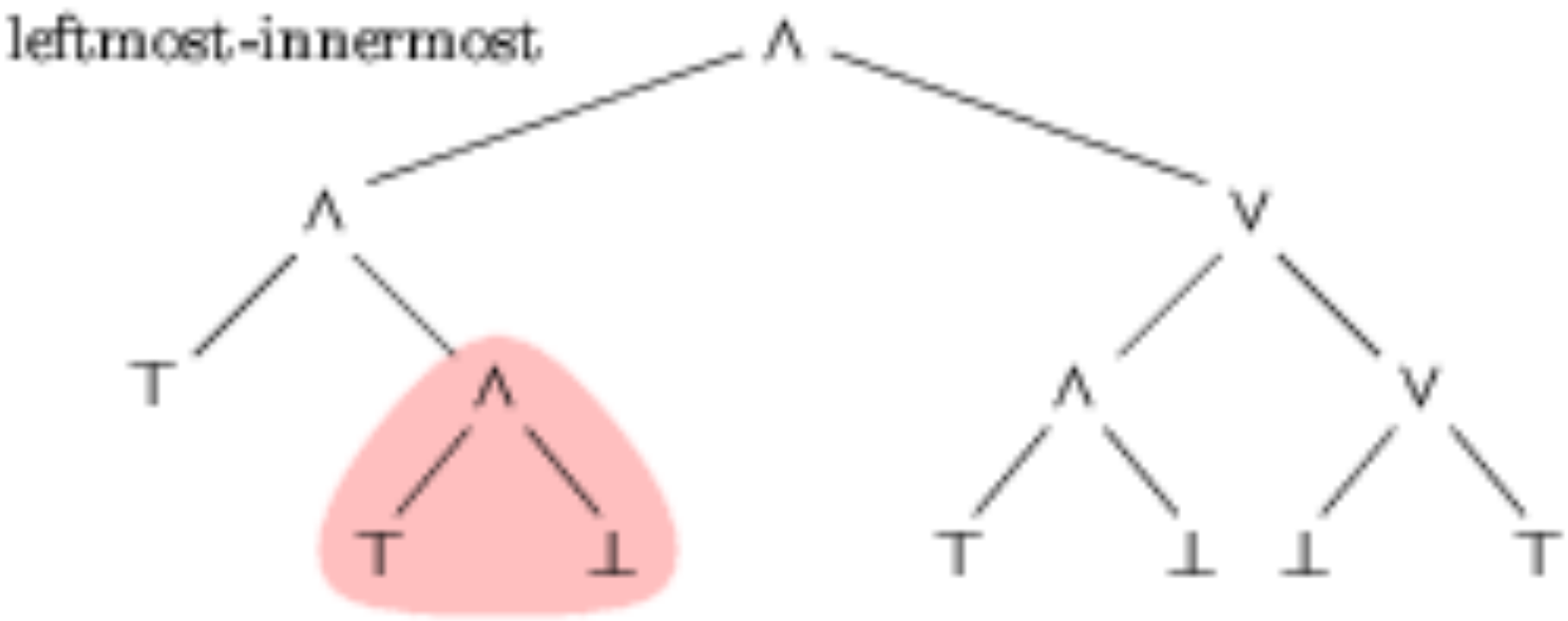
Example

$$
\begin{aligned}
\wedge(\top, x) &\rightarrow x \\
\wedge(\bot, x) &\rightarrow \bot \\
\vee(\top, x) &\rightarrow \top \\
\vee(\bot, x) &\rightarrow x
\end{aligned}
$$

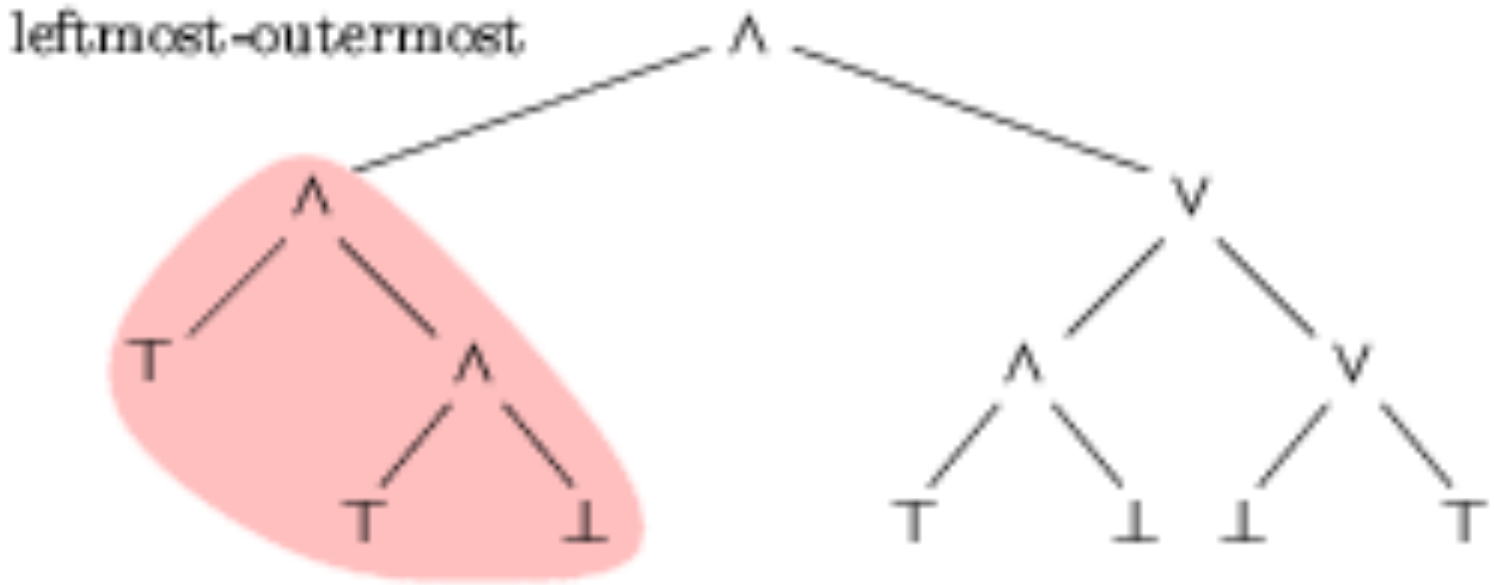
Strategy: Parallel Innermost



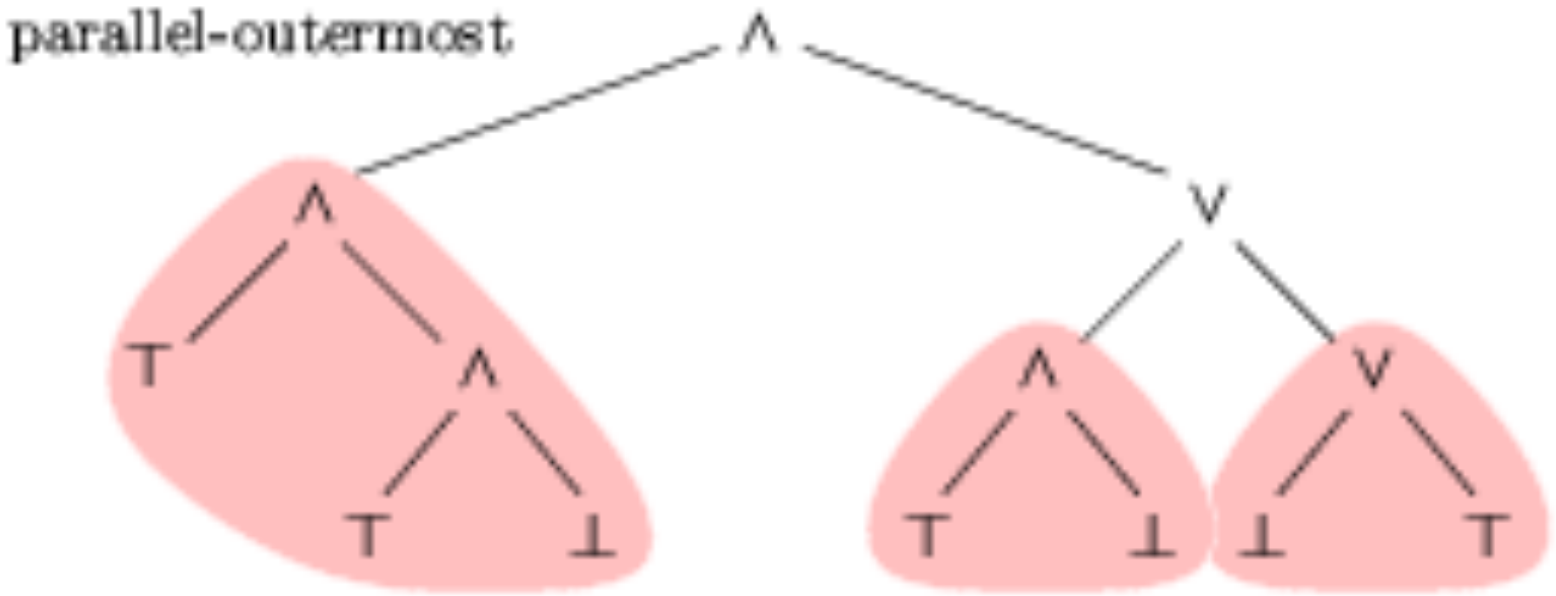
Strategy: Leftmost Innermost



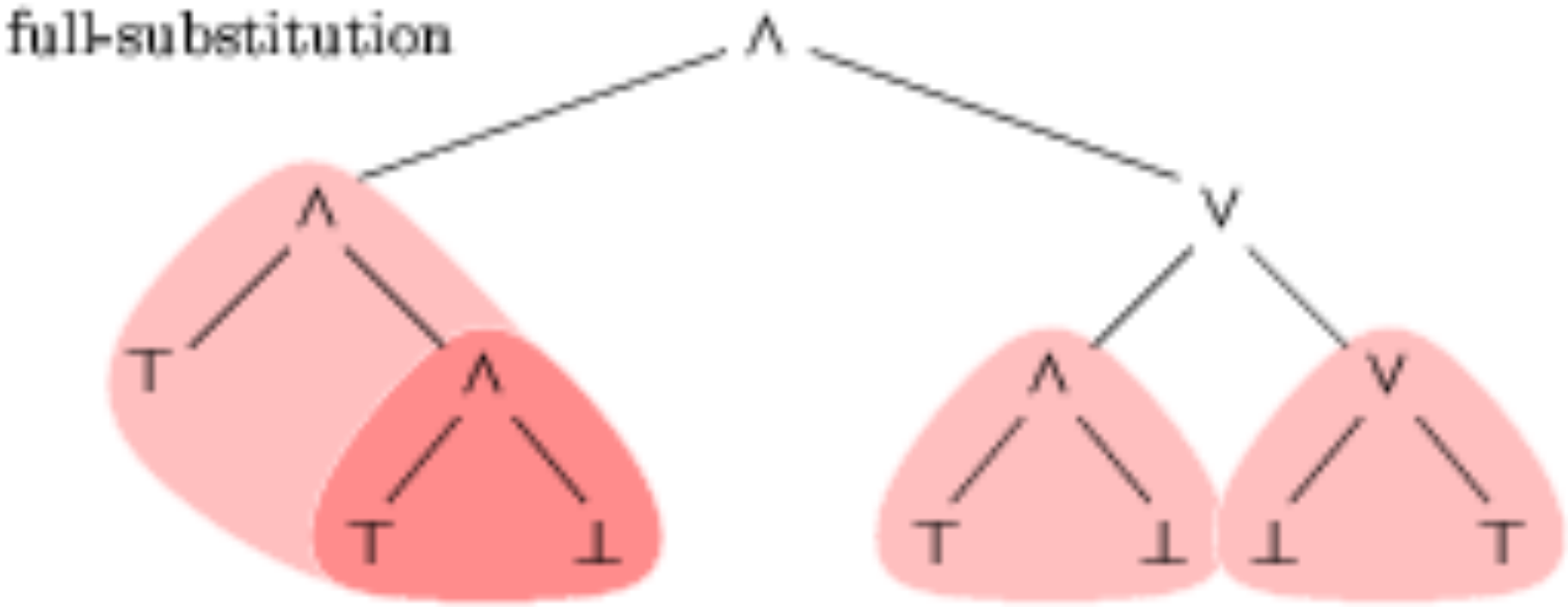
Strategy: Leftmost Outermost



Strategy: Parallel Outermost



Strategy: Full Substitution



Outermost Rewriting

- Outermost is normalizing.
- Parallel outermost is normalizing.
- Fair outermost is normalizing.

Counter/example

$$F(x,B) \rightarrow D$$

$$A \rightarrow B$$

$$C \rightarrow C$$

$$F(C,A)$$

Fixpoints

Outermost and full substitution compute the least fixpoint of recursive definitions.

Critical Pair Lemma

A rewrite system is locally confluent

iff

all its critical pairs are joinable

Summary

	orthogonal	orthogonal left-normal	orthogonal non-erasing
leftmost–innermost	p	p	p n
parallel–innermost	p	p	p n
leftmost–outermost (leftmost–fair)		n	p n
parallel–outermost (outermost–fair)	n	n	p n
full substitution (fair)	n c	n c	p n c