# Rewrite Systems

## 11. Orthogonality

# Tentative Course Outline

1. Introduction
2. Termination
3. Church-Rosser
4. Orthogonality
5. Diagrams
6. Completion
7. Saturation
8. Modularity
9. Unification
10. Induction
11. Polynomials
12. Boolean Rings
13. Extensions
14. Open Problems

# Orthogonal Systems

- Left linear
  - no duplication of variables on left
- Non-overlapping
  - No critical pairs
  - (or only trivial ones)
- No extra variables
  - no variables on right that are not on left

# Equational Programming

Michael J. O'Donnell. *Equational Logic as a Programming Language.* MIT Press, Cambridge, MA, 1985.

# Properties

- Termination
  - Normalization
    - Has at least one normal form
- Confluence
  - Unique normalization
    - Has at most one normal form
- Order of evaluation may matter!!

# Why Nontermination?

- Interpreters
- Streams

# Interpreter

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

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

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

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

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

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

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

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

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

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

# Orthogonality

- Theory of functional programming
  - Confluence [Chap. 4]
  - Normalization [Chap. 4]
    - Optimality [Chap. 9]

# Main Theorems

- Orthogonal systems are Church-Rosser.
  - (5 proofs in chapter)
- Outermost rewriting is normalizing.

# Functional Programming Languages

- ML
- SASL
- Miranda
- Haskell
- Curry

# 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);

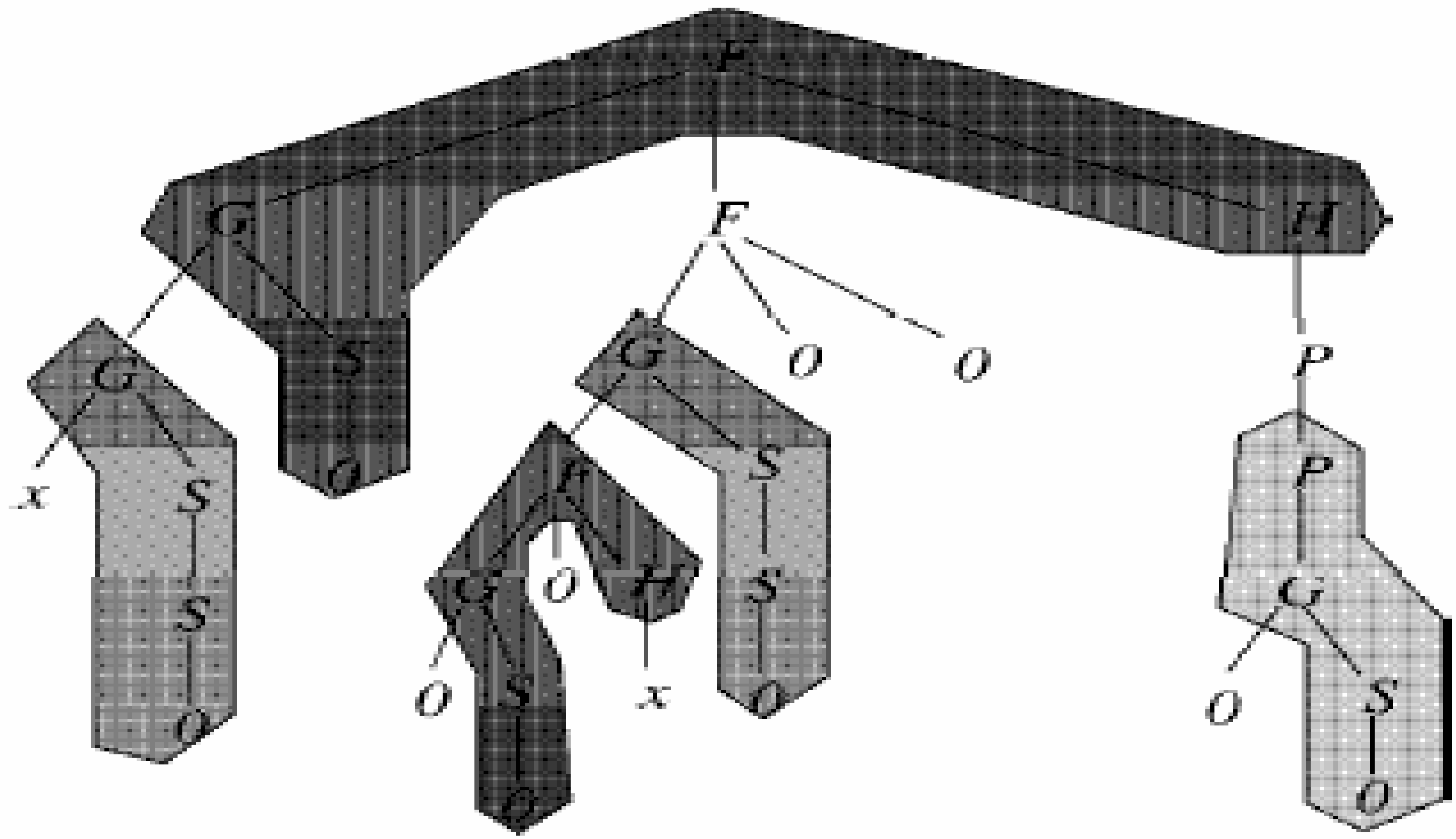
- fun reverse nil = nil
  | reverse (x::xs) =
  append ((reverse xs), [x]);

# Notions

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

# 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
G
O
O
P
x
S
O
R
S
P
G
O
H
S
S
G
O
x
O
O
S
O
O

# Combinatory Logic

$$Ix \to x$$

$$(Kx)y \to x$$

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

# Combinatory Logic

$$Ix \rightarrow x$$

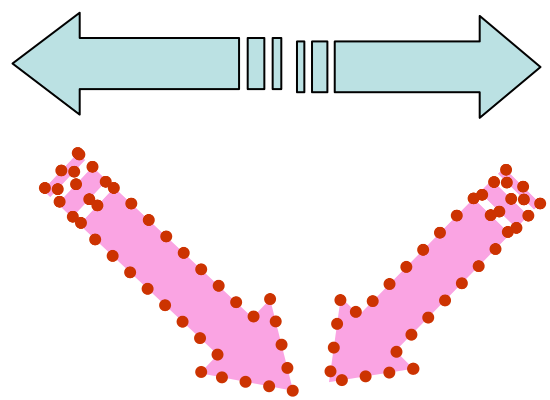
$$(Kx)y \rightarrow x$$

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

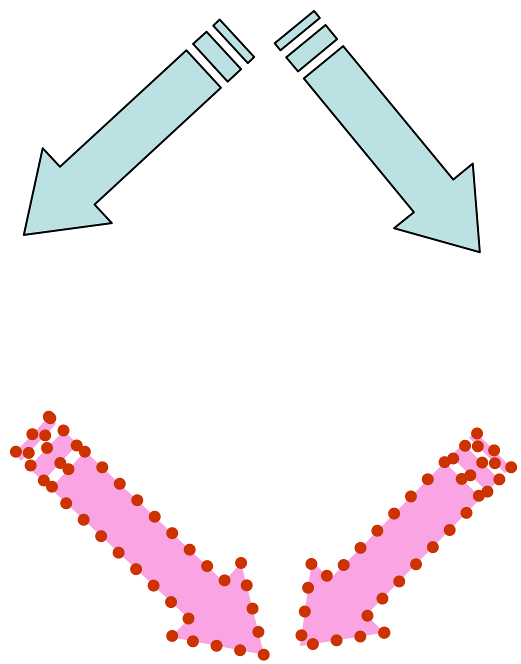
$$(Ex)x \rightarrow 1$$

$$(Ex)(Sx) \rightarrow 0$$

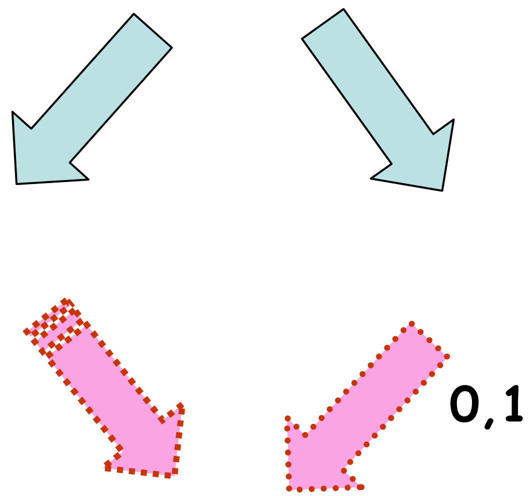
# Church-Rosser



# Confluence



# Strong Confluence



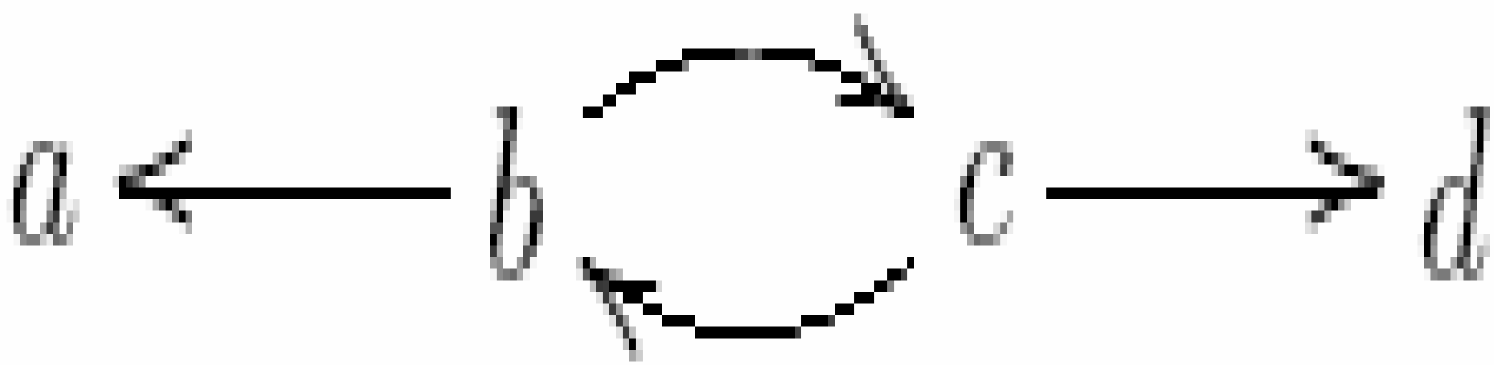
# Triangle Property

(vi) $a \in A$ has the *triangle property* if $\exists a' \in A \; \forall b \in A \; (a \rightarrow b \Rightarrow b \rightarrow a')$. (The name 'triangle property' refers to cases in which we have $a \rightarrow a'$.) The reduction relation $\rightarrow$ has the *triangle property* (notation TP) if every $a \in A$ has the *triangle property*.

# Newmann's Lemma

$$LC + SN \Rightarrow CR$$

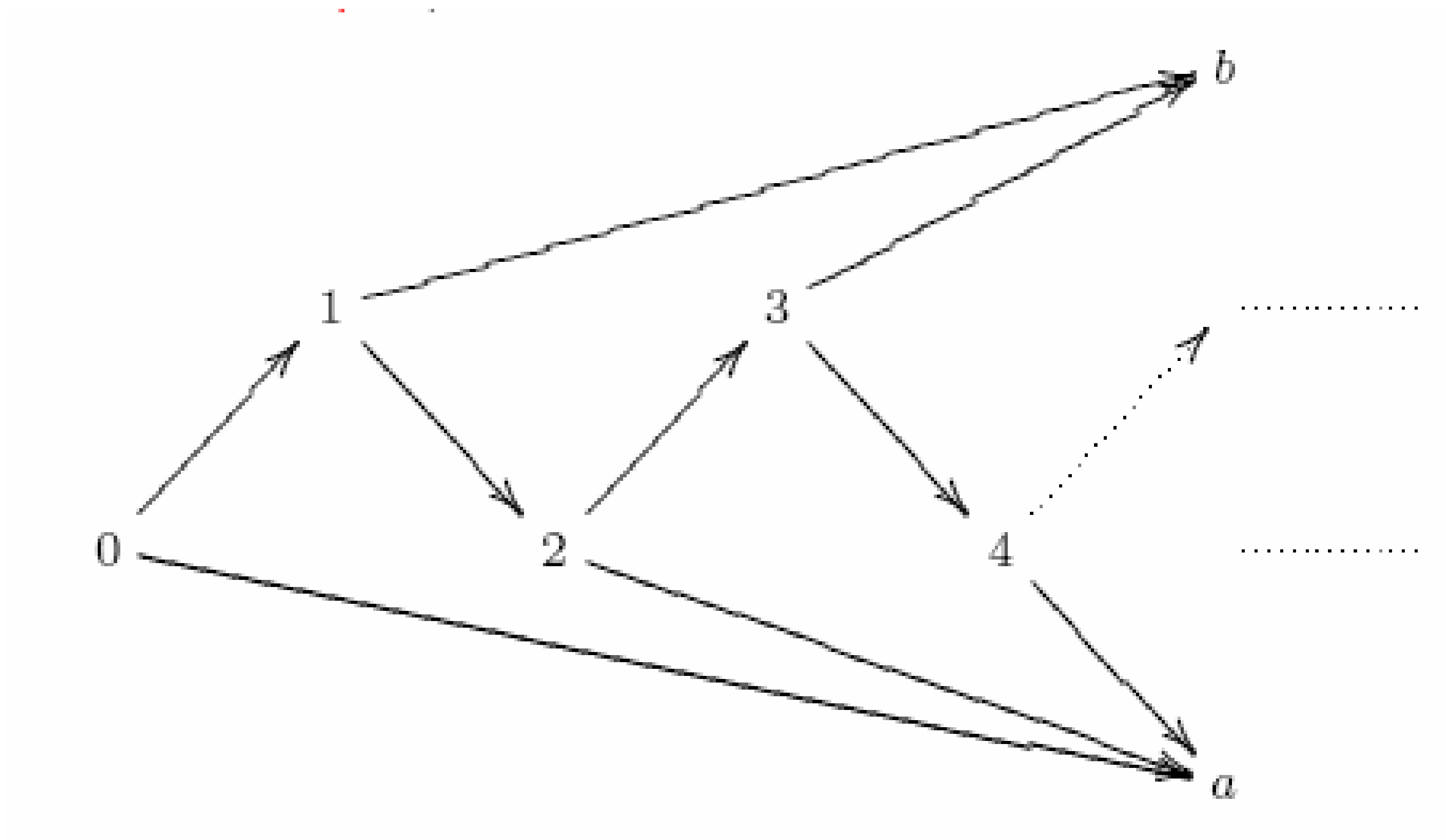
# Counterexample [Curry]



# Counterexample

$$f(x,x) \to a$$
$$f(x,g(x)) \to b$$
$$c \to g(c)$$

# Counterexample



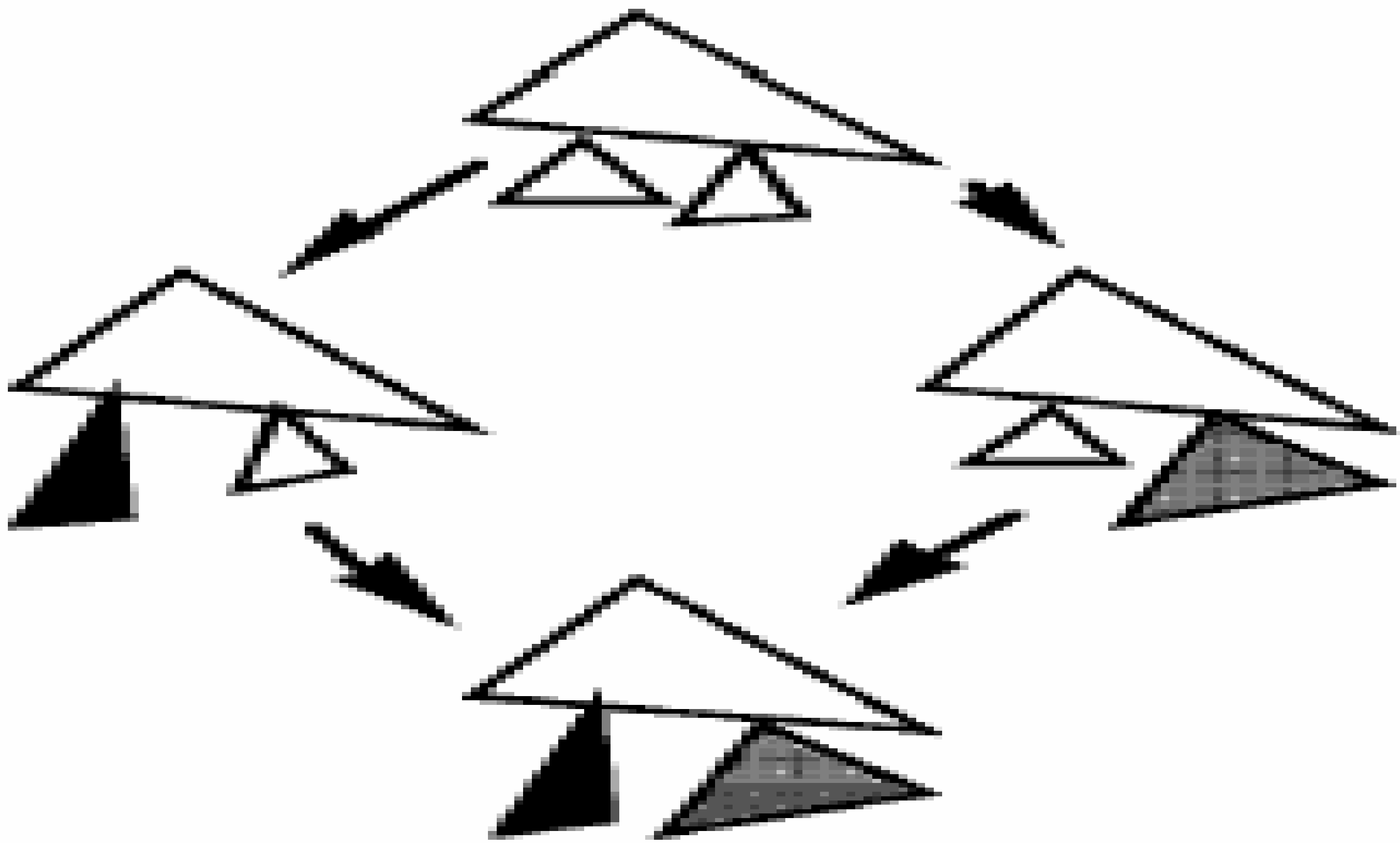
# Counterexample

$$f(x,x) \to a$$
$$g(x) \to f(x,g(x))$$
$$c \to g(c)$$

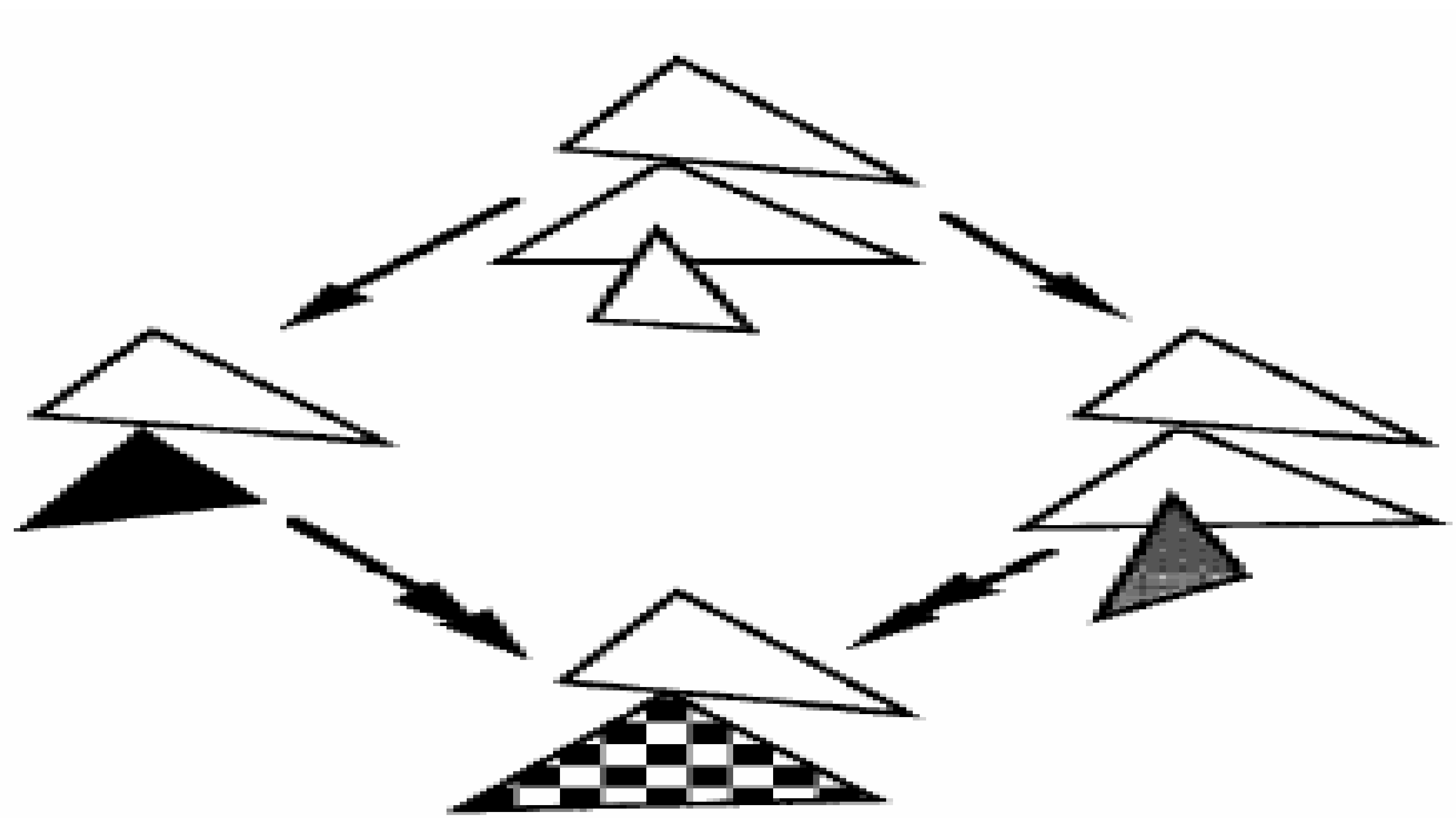
# Critical Pair Lemma

A rewrite system is locally confluent
iff
all its critical pairs are joinable

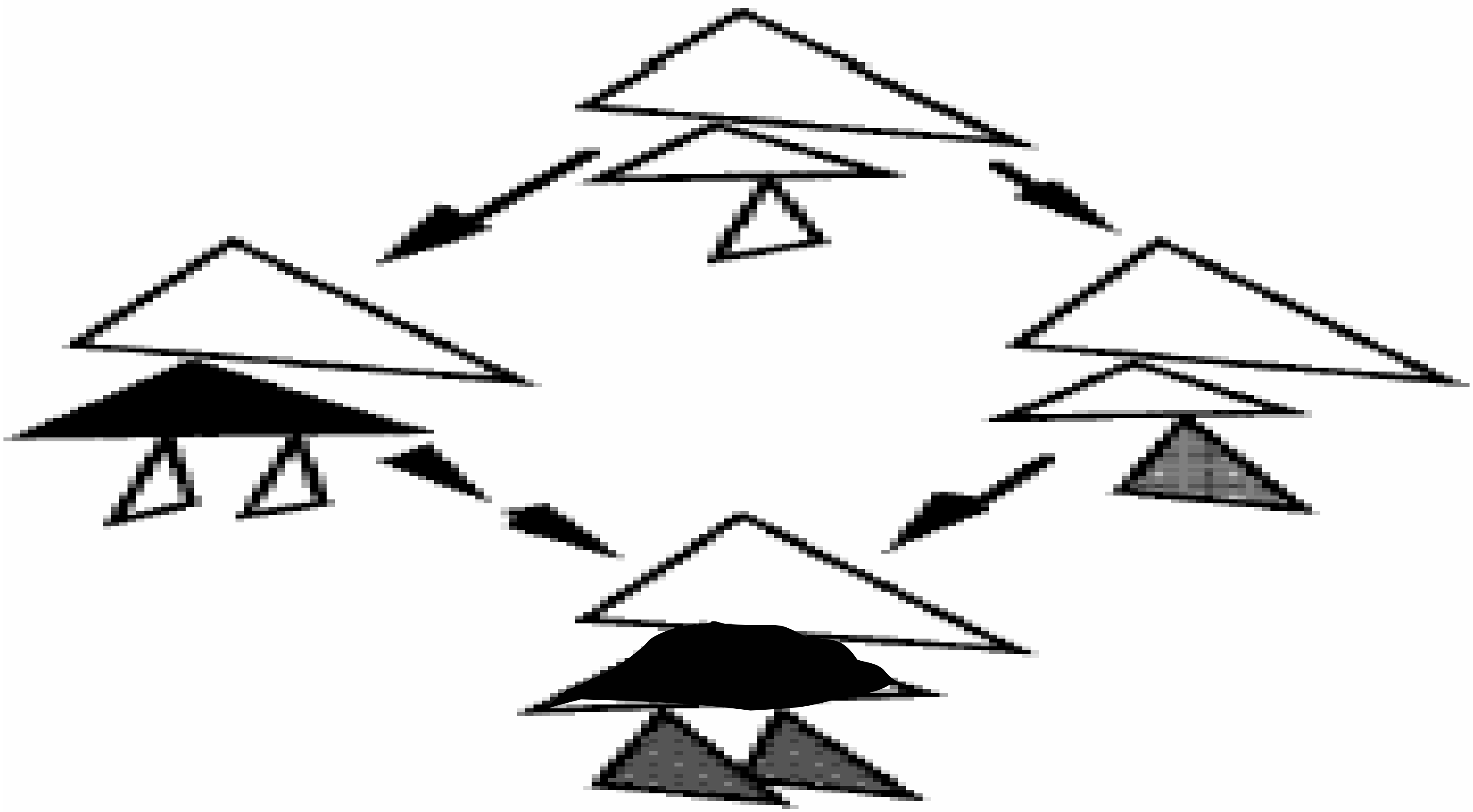
# Disjoint Redexes



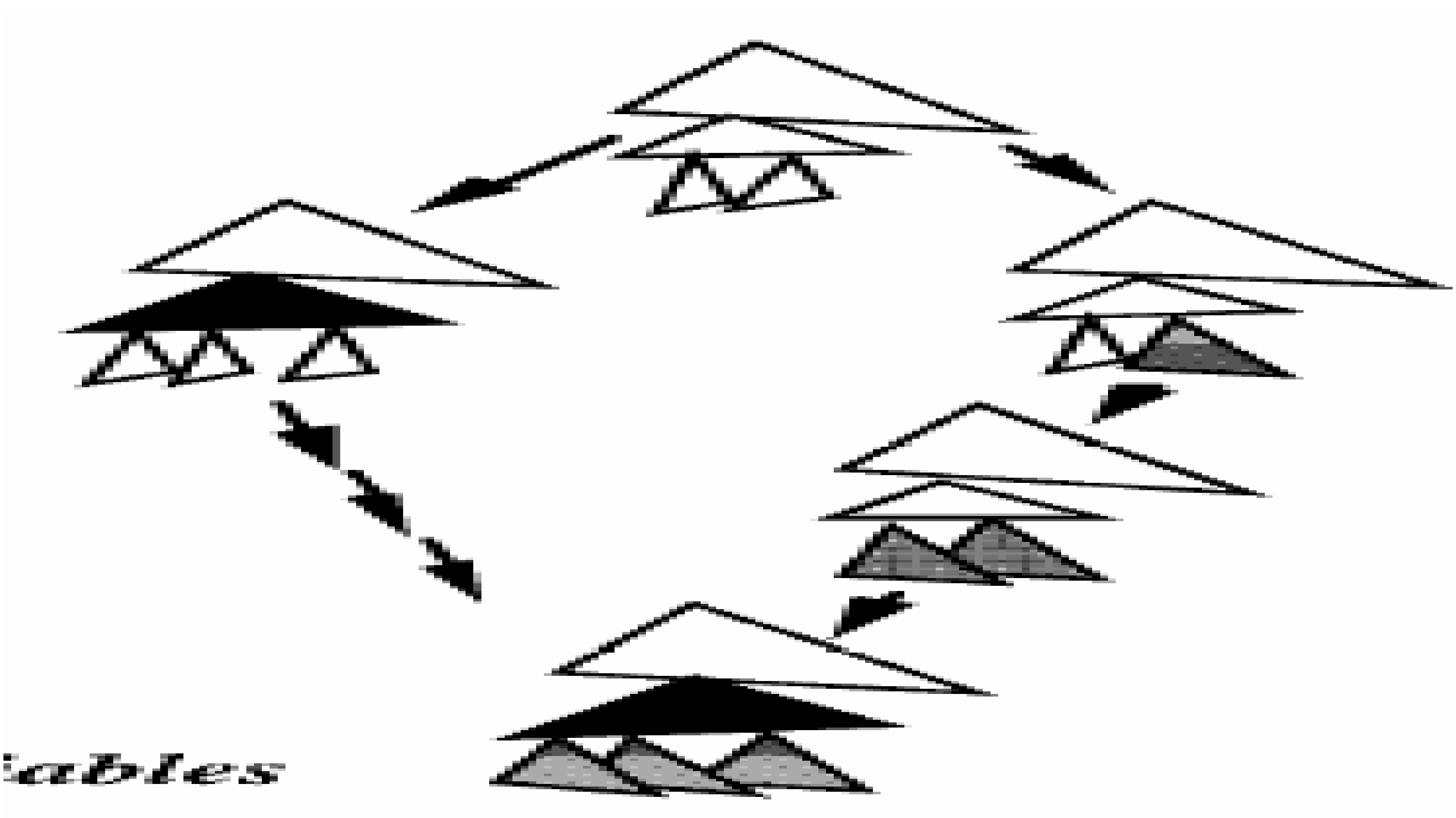
# Overlapping Redexes



# Nested Redexes



# Cont'd



# Intermediate Relation

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

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

- In particular

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

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

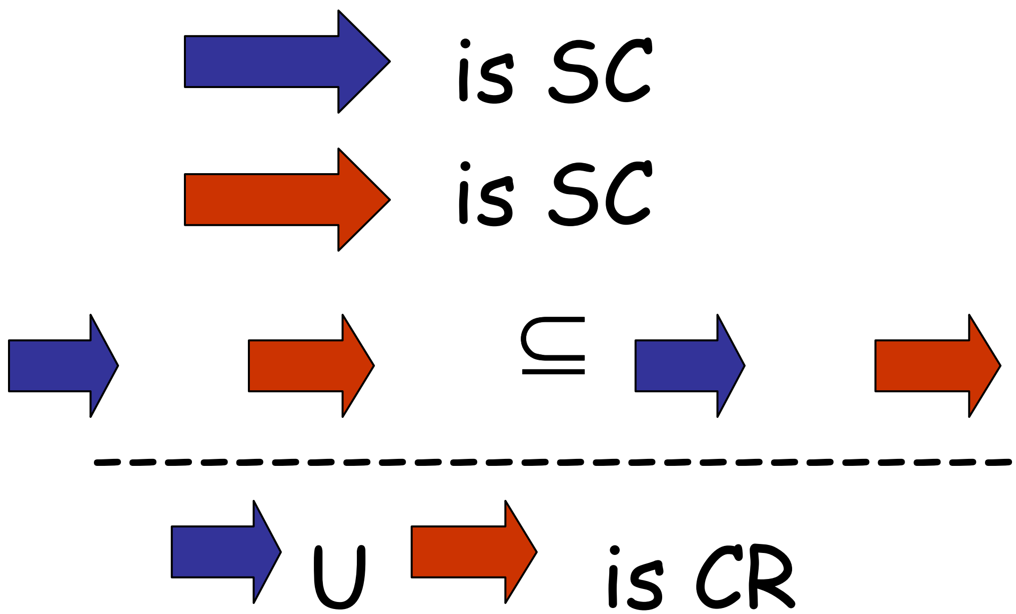
2. $\rightarrow^{\circ}$ is SC

3. So $\rightarrow$ CR

# $\rightarrow^{\circ}$ is SC

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

# Hindley-Rosen Lemma



# 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

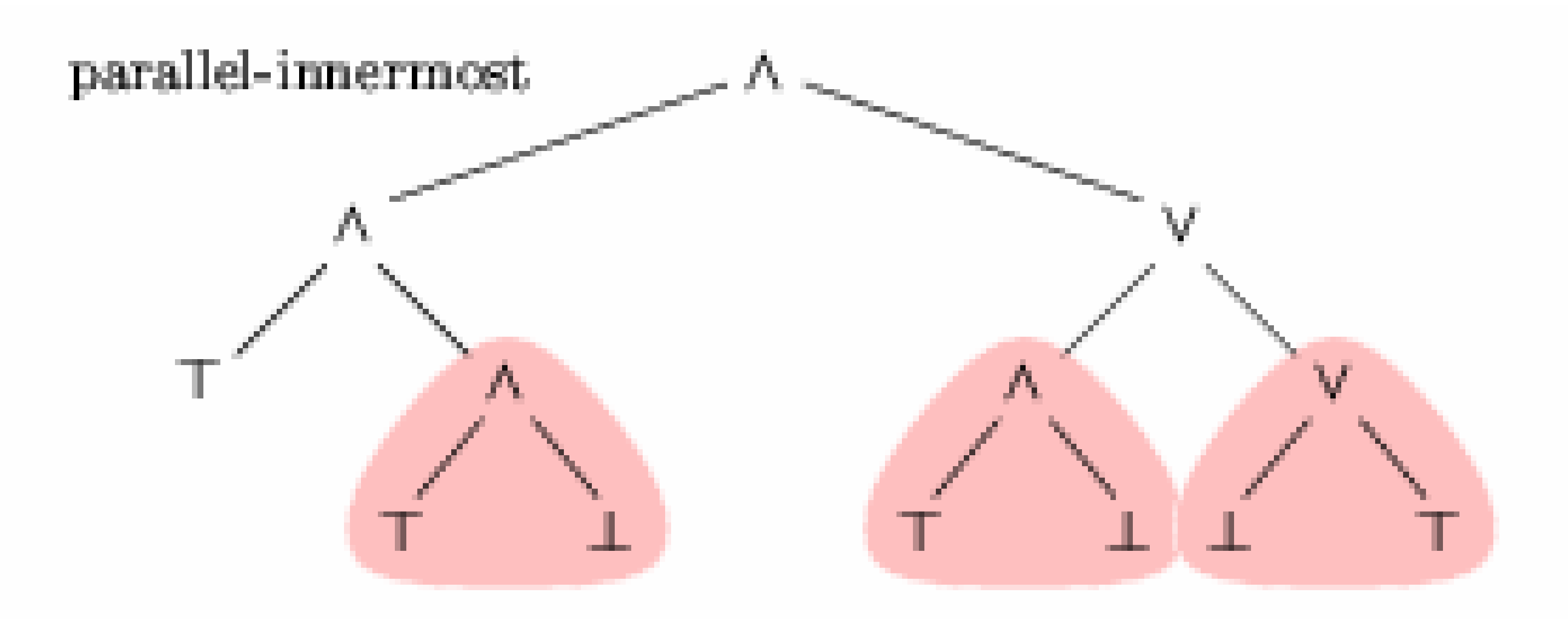
# Normalization Strategies

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

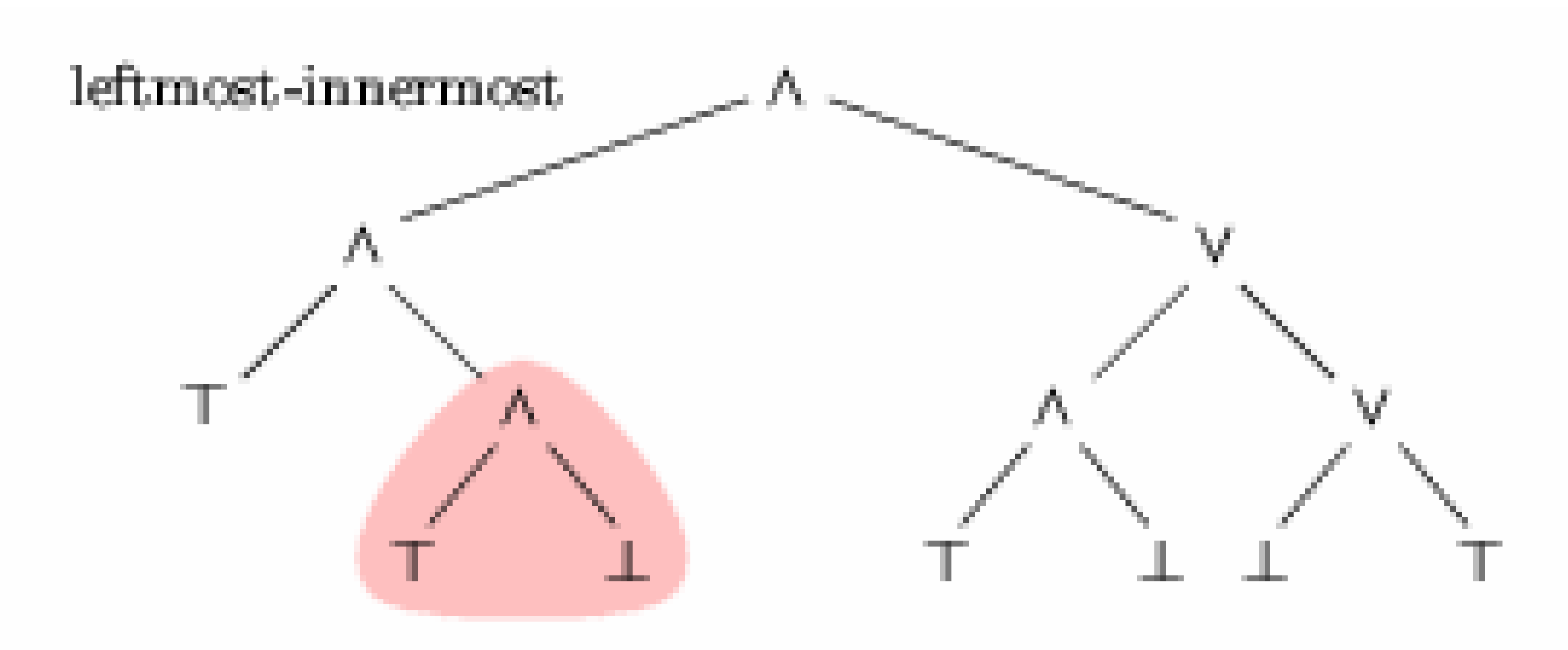
# Example

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

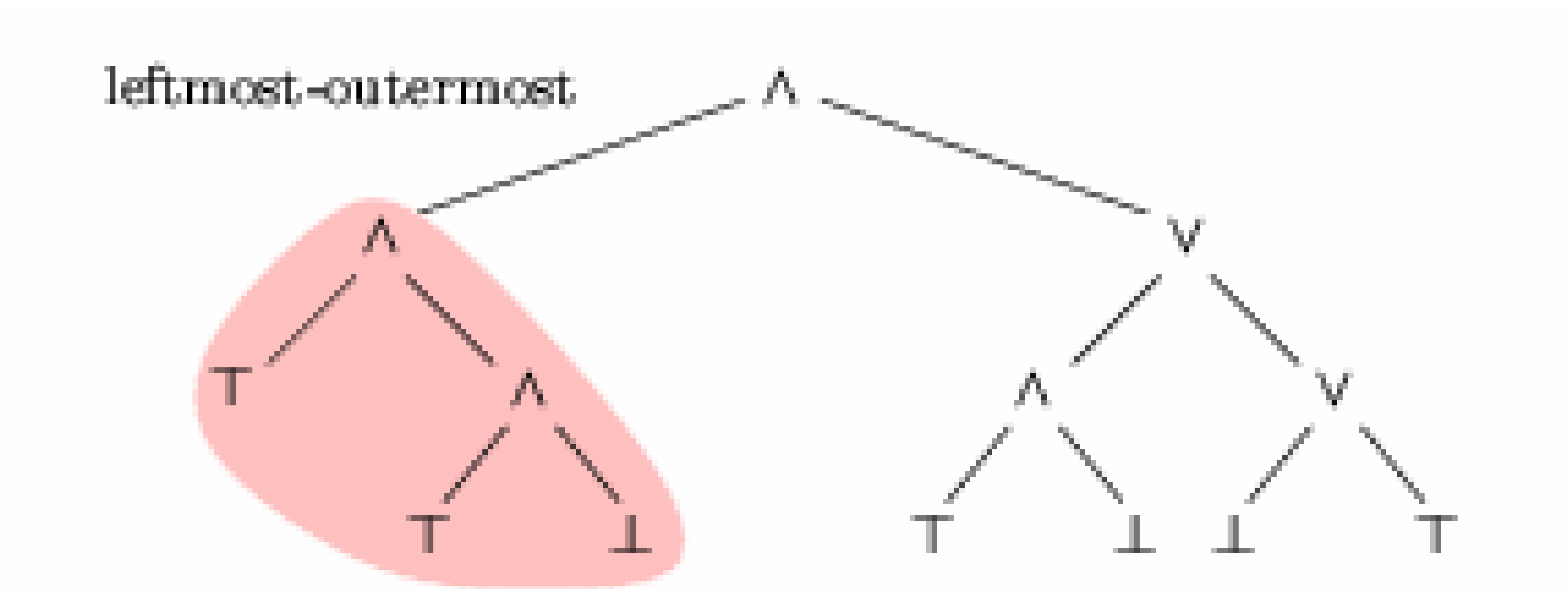
# Strategy: Parallel Innermost



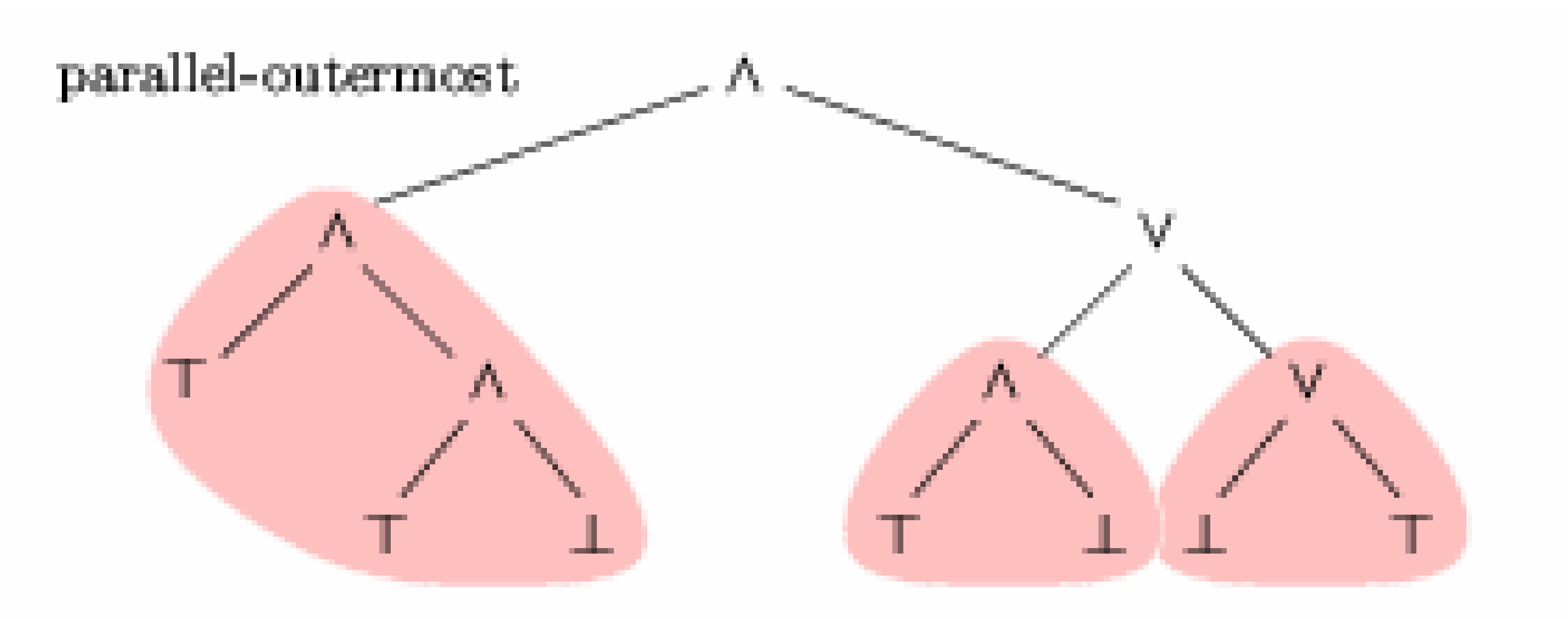
# Strategy: Leftmost Innermost



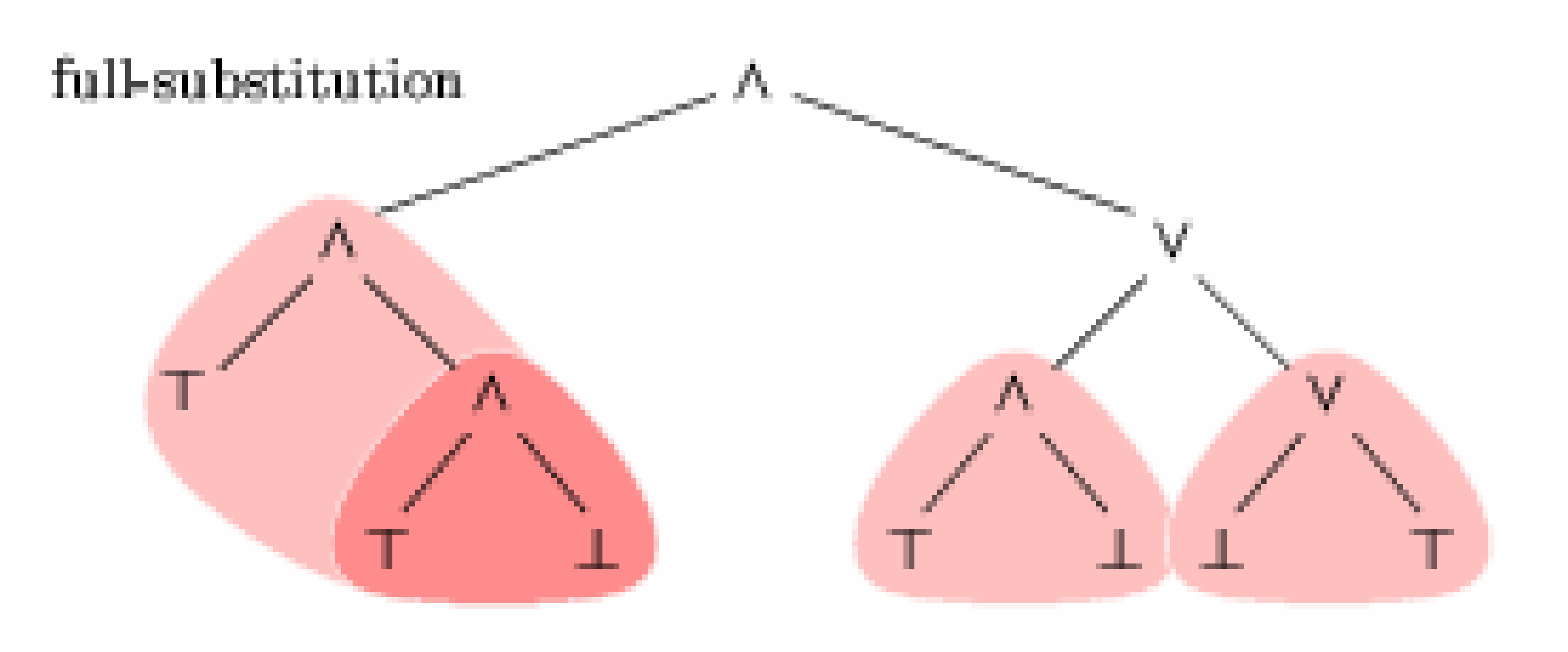
# Strategy: Leftmost Outermost



# Strategy: Parallel Outermost



# Strategy: Full Substitution



# Outermost Rewriting

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

# Counterexample

$$F(x,B) \to D$$
$$A \to B$$
$$C \to C$$

$$F(C,A)$$

# 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 |

Table 4.2: Some properties of strategies