Commutation, motivation, localisation

Julian Nagele
Vincent van Oostrom

University of Innsbruck

Thursday July 6, 14:00–15:30, ISR 2017

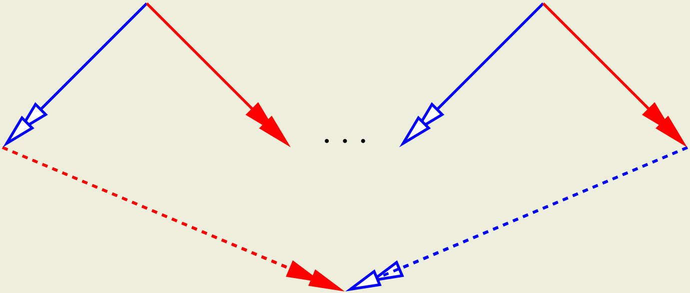
Contents

- Motivation
- Commutation in disguise
- Local commutation

Commutation

Definition

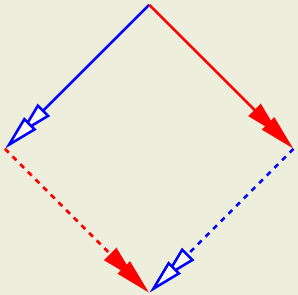
$\triangleright$, $\blacktriangleright$ have Church–Rosser property if $\triangleleft\!\blacktriangleright^* \subseteq \blacktriangleright\!\!\blacktriangleright \cdot \triangleleft\!\!\triangleleft$



Commutation

Definition

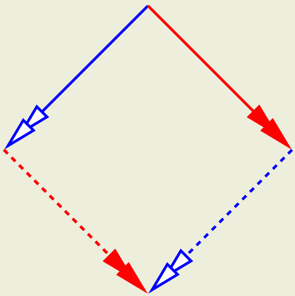
$\triangleright$ commutes with $\blacktriangleright$ if $\triangleleft\!\triangleleft \cdot \blacktriangleright\!\!\blacktriangleright \subseteq \blacktriangleright\!\!\blacktriangleright \cdot \triangleleft\!\triangleleft$



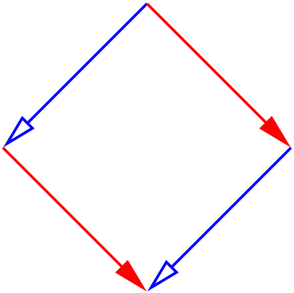
Commutation

Definition

$\forall a, b, c\, \exists d\;\; b \twoheadleftarrow a \twoheadrightarrow c \implies b \twoheadrightarrow d \twoheadleftarrow c$

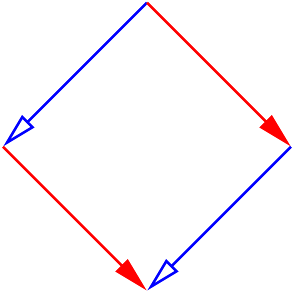


Commuting or not?



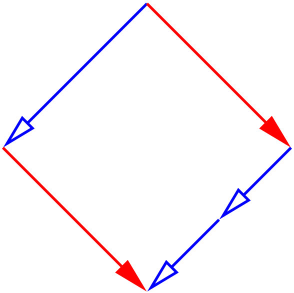
Do $\triangleright$, $\blacktriangleright$ commute?

Commuting or not?



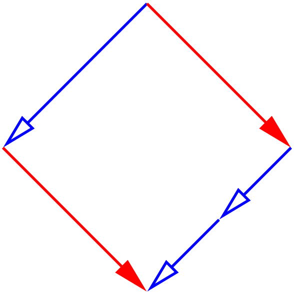
Do $\triangleright$, $\blacktriangleright$ commute? Yes

Commuting or not?



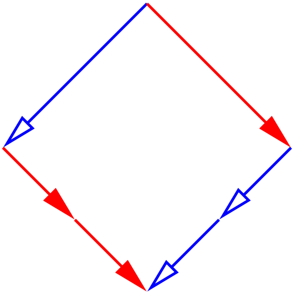
Do $\textcolor{blue}{\triangleright}$, $\textcolor{red}{\blacktriangleright}$ commute?

Commuting or not?



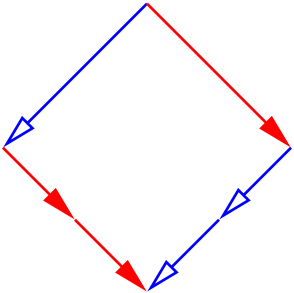
Do $\triangleright$, $\blacktriangleright$ commute? Yes

Commuting or not?



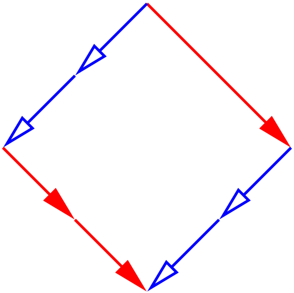
Do $\color{blue}{\triangleright}$, $\color{red}{\blacktriangleright}$ commute?

Commuting or not?



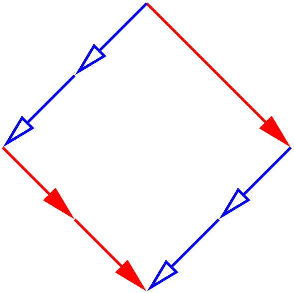
Do $\vartriangleright$, $\blacktriangleright$ commute? Yes

Commuting or not?



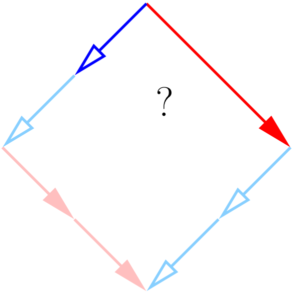
Do $\triangleright$, $\blacktriangleright$ commute?

Commuting or not?



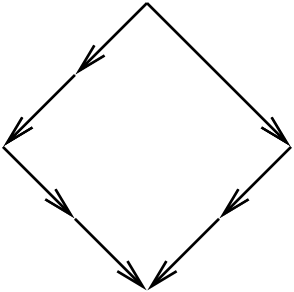
Do ▷, ▶ commute? No

Commuting or not?



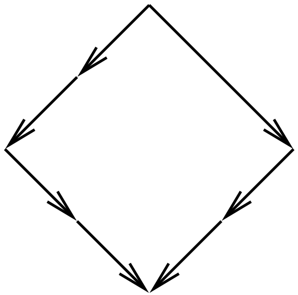
Do $\triangleright$, $\blacktriangleright$ commute? No

Commuting or not?



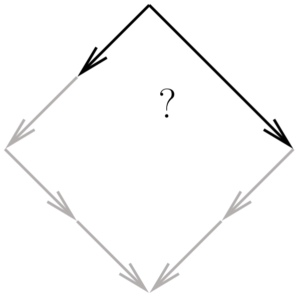
Is $\rightarrow$ confluent?

Commuting or not?



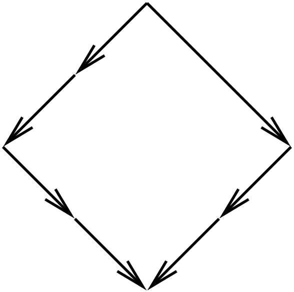
Is $\to$ confluent? Yes

Commuting or not?



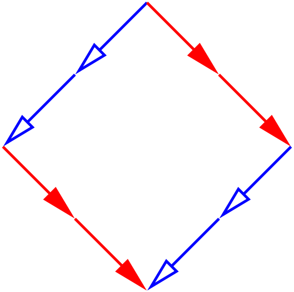
Is $\rightarrow$ confluent? Yes

Commuting or not?



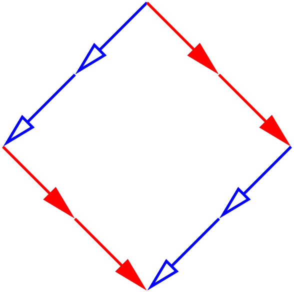
Is $\rightarrow$ confluent? Yes

Commuting or not?



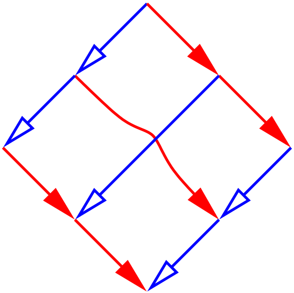
Do $\triangleright$, $\blacktriangleright$ commute?

Commuting or not?



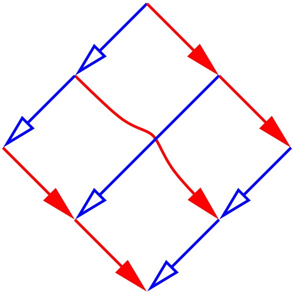
Do ▷, ▶ commute? No

Commuting or not?



Do $\color{blue}\triangleright$, $\color{red}\blacktriangleright$ commute?

Commuting or not?



Do $\triangleright$, $\blacktriangleright$ commute? No

Applications of commutation

- correctness of program transformation
  transformation $\triangleright$ commutes with semantics $\blacktriangleright$

Applications of commutation

- correctness of program transformation
  transformation ▷ commutes with semantics ▶
- optimisation of program transformation
  transformation ▷ ordered commutes with semantics ▶

Applications of commutation

- correctness of program transformation
  transformation $\triangleright$ commutes with semantics $\blacktriangleright$
- optimisation of program transformation
  transformation $\triangleright$ ordered commutes with semantics $\blacktriangleright$
- confluence by self-commutation
  if $\rightarrow$ commutes with $\rightarrow$, then $\rightarrow$ confluent

Applications of commutation

- correctness of program transformation
  transformation ▷ commutes with semantics ▶
- optimisation of program transformation
  transformation ▷ ordered commutes with semantics ▶
- confluence by self-commutation
  if $\to$ commutes with $\to$, then $\to$ confluent
- confluence modularly
  if $\forall i, j \to_i$ commutes with $\to_j \implies$, then $\bigcup_k \to_k$ confluent

Applications of commutation

- correctness of program transformation
  transformation $\triangleright$ commutes with semantics $\blacktriangleright$
- optimisation of program transformation
  transformation $\triangleright$ ordered commutes with semantics $\blacktriangleright$
- confluence by self-commutation
  if $\to$ commutes with $\to$, then $\to$ confluent
- confluence modularly
  if $\forall i, j \to_i$ commutes with $\to_j \implies$, then $\bigcup_k \to_k$ confluent
- termination modularly
  if $\triangleleft$,$\blacktriangleright$ commute lazily then $\triangleleft$,$\blacktriangleright$ terminating iff $\triangleleft \cup \blacktriangleright$ terminating

Applications of commutation

- correctness of program transformation
  transformation $\triangleright$ commutes with semantics $\blacktriangleright$
- optimisation of program transformation
  transformation $\triangleright$ ordered commutes with semantics $\blacktriangleright$
- confluence by self-commutation
  if $\to$ commutes with $\to$, then $\to$ confluent
- confluence modularly
  if $\forall i, j \to_i$ commutes with $\to_j \implies$, then $\bigcup_k \to_k$ confluent
- termination modularly
  if $\triangleleft, \blacktriangleright$ commute lazily then $\triangleleft, \blacktriangleright$ terminating iff $\triangleleft \cup \blacktriangleright$ terminating
- behavioural equivalence (bisimulation-up-to)
  $\triangleleft \cdot \sim \subseteq \sim \cdot \triangleleft$ and $\sim \cdot \triangleright \subseteq \triangleright \cdot \sim$ (R vs. $\sim \cdot R \cdot \sim$)

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\ \blacktriangleright\ y \\ a(s(x), y) &\ \blacktriangleright\ s(a(x, y)) \end{aligned}$$

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\ \blacktriangleright\ \ y \\ a(s(x),y) &\ \blacktriangleright\ \ s(a(x,y)) \end{aligned}$$

rewriting transformation

$$a(x,0) \ \triangleright\ \ x$$

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting transformation

$$a(x,0) \triangleright x$$

Lemma

transformation is correct

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\ {\color{red}\blacktriangleright}\ y \\ a(s(x), y) &\ {\color{red}\blacktriangleright}\ s(a(x, y)) \end{aligned}$$

rewriting transformation

$$a(x, 0) \ {\color{blue}\triangleright}\ x$$

Lemma

transformation is correct

Proof.

idea: transformation ${\color{blue}\triangleright}$ commutes with semantics ${\color{red}\blacktriangleright}$ $\square$

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting transformation

$$a(x,0) \triangleright x$$

Lemma

transformation is correct

Proof.

idea: transformation $\triangleright$ commutes with semantics $\blacktriangleright$ $\square$

Exercise

prove this

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\blacktriangleright y \\ a(s(x), y) &\blacktriangleright s(a(x, y)) \end{aligned}$$

rewriting transformation

$$a(a(x, y), z) \quad \triangleright \quad a(x, a(y, z))$$

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting transformation

$$a(a(x,y),z) \triangleright a(x,a(y,z))$$

Lemma

transformation is correct

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\;\blacktriangleright\; y \\ a(s(x), y) &\;\blacktriangleright\; s(a(x, y)) \end{aligned}$$

rewriting transformation

$$a(a(x, y), z) \;\triangleright\; a(x, a(y, z))$$

Lemma

transformation is correct

Proof.

idea: transformation $\triangleright$ commutes with semantics $\blacktriangleright$ $\square$

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\blacktriangleright y \\ a(s(x), y) &\blacktriangleright s(a(x, y)) \end{aligned}$$

rewriting transformation

$$a(a(x, y), z) \triangleright a(x, a(y, z))$$

Lemma

transformation is correct

Proof.

idea: transformation $\triangleright$ commutes with semantics $\blacktriangleright$ $\square$

Exercise

prove this does not hold. what does hold?

Correctness of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting transformation

$$a(a(x,y),z) \triangleright a(x,a(y,z))$$

Lemma

transformation is correct

Proof.

idea: transformation $\triangleright \cup \blacktriangleright$ commutes with semantics $\blacktriangleright$ $\square$

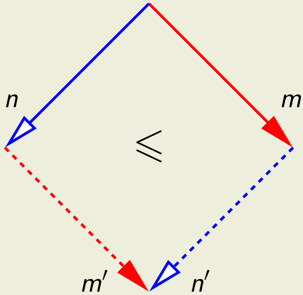
Exercise

prove this does hold.

Ordered commutation

Definition

$\rhd$ ordered commutes with $\blacktriangleright$ if ${}^{n}\!\lhd \cdot \blacktriangleright^{m} \subseteq \blacktriangleright^{m'} \cdot {}^{n'}\!\lhd$, $n + m' \leqslant m + n'$



Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\quad \blacktriangleright \quad y \\ a(s(x), y) &\quad \blacktriangleright \quad s(a(x, y)) \end{aligned}$$

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\;\blacktriangleright\; y \\ a(s(x),y) &\;\blacktriangleright\; s(a(x,y)) \end{aligned}$$

rewriting optimisation

$$a(x,0) \;\triangleright\; x$$

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\ \blacktriangleright\ y \\ a(s(x), y) &\ \blacktriangleright\ s(a(x, y)) \end{aligned}$$

rewriting optimisation

$$a(x, 0) \ \triangleright\ x$$

Lemma

transformation is optimisation

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\;\blacktriangleright\; y \\ a(s(x), y) &\;\blacktriangleright\; s(a(x, y)) \end{aligned}$$

rewriting optimisation

$$a(x, 0) \;\triangleright\; x$$

Lemma

transformation is optimisation

Proof.

idea: transformation $\triangleright$ ordered commutes with semantics $\blacktriangleright$ $\square$

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{array}{rcl} a(0,y) & \blacktriangleright & y \\ a(s(x),y) & \blacktriangleright & s(a(x,y)) \end{array}$$

rewriting optimisation

$$a(x,0) \quad \triangleright \quad x$$

Lemma

transformation is optimisation

Proof.

idea: transformation $\triangleright$ ordered commutes with semantics $\blacktriangleright$ $\square$

Exercise

prove this

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\blacktriangleright y \\ a(s(x), y) &\blacktriangleright s(a(x, y)) \end{aligned}$$

rewriting optimisation

$$a(a(x, y), z) \triangleright a(x, a(y, z))$$

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{array}{rcl} a(0,y) & \blacktriangleright & y \\ a(s(x),y) & \blacktriangleright & s(a(x,y)) \end{array}$$

rewriting optimisation

$$a(a(x,y),z) \quad \triangleright \quad a(x,a(y,z))$$

Lemma

optimisation is correct

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0, y) &\blacktriangleright y \\ a(s(x), y) &\blacktriangleright s(a(x, y)) \end{aligned}$$

rewriting optimisation

$$a(a(x, y), z) \triangleright a(x, a(y, z))$$

Lemma

optimisation is correct

Proof.

idea: optimisation $\triangleright$ ordered commutes with semantics $\blacktriangleright$ $\square$

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting optimisation

$$a(a(x,y),z) \vartriangleright a(x,a(y,z))$$

Lemma

optimisation is correct

Proof.

idea: optimisation $\vartriangleright$ ordered commutes with semantics $\blacktriangleright$ □

Exercise

prove this does not hold. what does hold?

Optimisation of program transformation

rewriting semantics (meaning is closed normal form)

$$\begin{aligned} a(0,y) &\blacktriangleright y \\ a(s(x),y) &\blacktriangleright s(a(x,y)) \end{aligned}$$

rewriting optimisation

$$a(a(x,y),z) \triangleright a(x,a(y,z))$$

Lemma

optimisation is correct

Proof.

idea: optimisation $\triangleright \cup \blacktriangleright$ ordered commutes with semantics $\blacktriangleright$ $\square$

Exercise

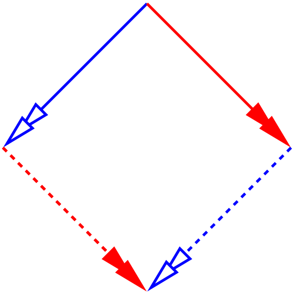
prove this does hold.

Contents

- Motivation
- Commutation in disguise
- Local commutation

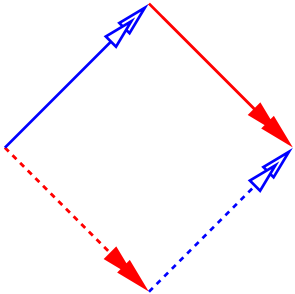
Factorisation

if $\textcolor{blue}{\lhd\!\!\lhd} \cdot \textcolor{red}{\blacktriangleright\!\!\blacktriangleright} \subseteq \textcolor{red}{\blacktriangleright\!\!\blacktriangleright} \cdot \textcolor{blue}{\lhd\!\!\lhd}$, then $\textcolor{blue}{\rhd}$ commutes with $\textcolor{red}{\blacktriangleright}$



Factorisation

if $\triangleright\!\!\triangleright \cdot \blacktriangleright\!\!\blacktriangleright \subseteq \blacktriangleright\!\!\blacktriangleright \cdot \triangleright\!\!\triangleright$, then $\blacktriangleright$,$\triangleright$ factorisation holds



Postponement

λ-calculus with β and η-reduction

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

η-steps can be postponed until after β-steps

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Proof.

Idea: repeatedly replace $M \to_{\eta} P \to_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ $\square$

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Proof.

Idea: repeatedly replace $M \to_{\eta} P \to_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ $\square$

Exercise

Would this idea be sufficient to prove postponement?
Rephrased: is this process terminating/normalising in general/here?

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Proof.

Idea: repeatedly replace $M \to_{\eta} P \to_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ □

Exercise

Would this idea be sufficient to prove postponement?
Rephrased: is this process terminating/normalising in general/here?

Answer

No, e.g. $01 \Rightarrow 1100$ is not terminating

$$\underline{01}1 \Rightarrow 110\underline{01} \Rightarrow 11101100 \Rightarrow \ldots$$

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Proof.

$C[\lambda x.Mx] \rightarrow_{\eta} C[M] \rightarrow_{\beta} N$, distinguish on where β-step is

- if $M \rightarrow_{\beta} P$, then $C[\lambda x.Mx] \rightarrow_{\beta} C[\lambda x.Px] \rightarrow_{\eta} C[P]$
- if $C \rightarrow_{\beta} D$, then $C[\lambda x.Mx] \rightarrow_{\beta} D[\lambda x.Px] \twoheadrightarrow_{\eta} C[P]$
- if overlaps both C, M, $C[\lambda x.Mx] \rightarrow_{\beta} P \rightarrow_{\beta} N$

$\square$

Postponement

λ-calculus with β and η-reduction

Theorem (η-postponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

Proof.

Idea: repeatedly replace $M \multimap\!\!\!\rightarrow_{\eta} P \to_{\beta} N$ by $M \to_{\beta}^{+} Q \multimap\!\!\!\rightarrow_{\eta} N$ $\square$

Preponement for termination

λ-calculus with β and η-reduction

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

β-steps can be preponed before β-steps

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so termination of $\beta\eta$ follows from termination of β,η

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so termination of $\beta\eta$ follows from termination of β,η

Proof.

Idea: replace leftmost $M \rightarrow_{\eta} P \rightarrow_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ $\square$

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so termination of $\beta\eta$ follows from termination of β,η

Proof.

Idea: replace leftmost $M \to_{\eta} P \to_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ □

Exercise

Would idea be sufficient to prove preponement, in general/here?

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so termination of $\beta\eta$ follows from termination of β,η

Proof.

Idea: replace leftmost $M \to_{\eta} P \to_{\beta} N$ by $M \twoheadrightarrow_{\beta} Q \twoheadrightarrow_{\eta} N$ □

Exercise

Would idea be sufficient to prove preponement, in general/here?

Answer

No, e.g. $b \to^{0} a \to^{0,1} a' \to^{1} c$ (01 $\Rightarrow$ 00,01 $\Rightarrow$ 11 not terminating)

$$\underline{01}1 \to 0\underline{01} \to 011 \to \ldots$$

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so termination of $\beta\eta$ follows from termination of β,η

Proof.

$C[\lambda x.Mx] \to_\eta C[M] \to_\beta N$

- if $M \to_\beta P$, then $C[\lambda x.Mx] \to_\beta C[\lambda x.Px] \to_\eta C[P]$
- if $C \to_\beta D$, then $C[\lambda x.Mx] \to_\beta C[\lambda x.Px] \to_\eta C[P]$
- if overlaps both C,M, $C[\lambda x.Mx] \to_\beta P \to_\beta N$

$\square$

Preponement for termination

λ-calculus with β and η-reduction

Theorem (β-preponement)

$M \twoheadrightarrow_{\beta\eta} N \implies M \twoheadrightarrow_{\beta} \cdot \twoheadrightarrow_{\eta} N$

so *termination of $\beta\eta$ follows from termination of β,η*

Proof.

$C[\lambda x.Mx] \to_{\eta} C[M] \to_{\beta} N$

- if $M \to_{\beta} P$, then $C[\lambda x.Mx] \to_{\beta} C[\lambda x.Px] \to_{\eta} C[P]$
- if $C \to_{\beta} D$, then $C[\lambda x.Mx] \to_{\beta} C[\lambda x.Px] \to_{\eta} C[P]$
- if overlaps both C,M, $C[\lambda x.Mx] \to_{\beta} P \to_{\beta} N$

Idea: repeatedly replace $M \to_{\eta} P \to_{\beta} N$ by $M \to_{\beta} Q \twoheadrightarrow_{\beta\eta} N$ $\square$

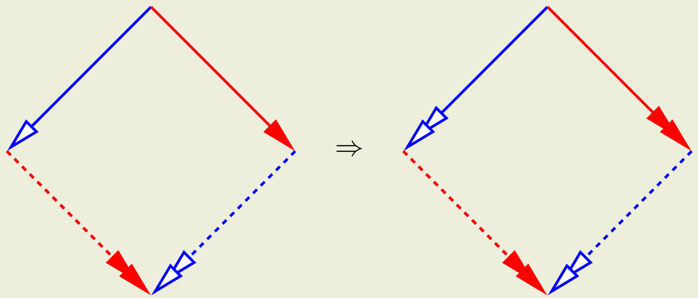
Contents

- Motivation
- Commutation in disguise
- Local commutation

Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\rightarrow = \vartriangleright \cup \blacktriangleright$ *terminating*



Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\to = \triangleright \cup \blacktriangleright$ *terminating*

Proof.

negative: infinite tiling impossible

positive: $\forall a$, $\triangleleft\ a\ \blacktriangleright \subseteq\ \blacktriangleright\!\!\blacktriangleright \cdot \lhd\!\!\lhd$ by induction on a, ordered by $\to^{+}$ $\square$

Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\rightarrow = \textcolor{blue}{\triangleright} \cup \textcolor{red}{\blacktriangleright}$ *terminating*

Exercise

show that without termination commutation need not hold

Commuting version of Newman's Lemma

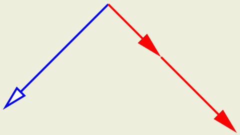
Theorem

local commutation implies commutation, if $\to \; = \; \triangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \overset{\lhd}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

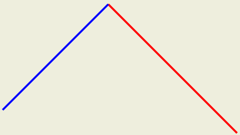
Theorem

local commutation implies commutation, if $\to \,=\, \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \overset{\lhd}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

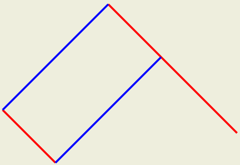
Theorem

local commutation implies commutation, if $\to \; = \; \vartriangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \vartriangleleft a \overset{\vartriangleleft}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

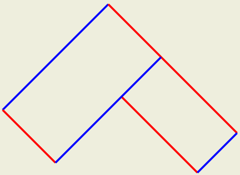
Theorem

local commutation implies commutation, if $\to = \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \mathrel{\substack{\lhd \\ \blacktriangleright}} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

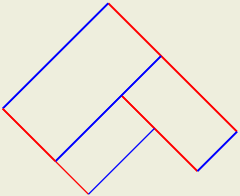
Theorem

local commutation implies commutation, if $\to\; = \;\rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \overset{\lhd}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

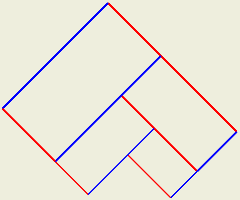
Theorem

local commutation implies commutation, if $\to\; = \;\triangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \triangleleft a \underset{\blacktriangleright}{\overset{\triangleleft}{}} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

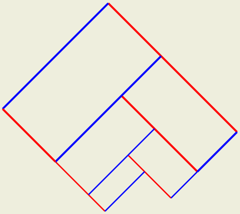
Theorem

local commutation implies commutation, if $\to = \vartriangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \vartriangleleft a \overset{\vartriangleleft}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

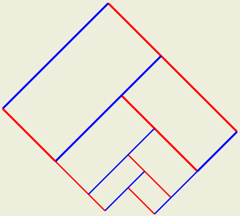
Theorem

local commutation implies commutation, if $\to\; = \;\vartriangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \vartriangleleft a \underset{\blacktriangleright}{\vartriangleleft} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

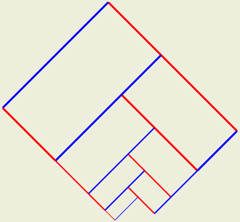
Theorem

local commutation implies commutation, if $\to \, = \, \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \mathrel{\substack{\lhd \\ \blacktriangleright}} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

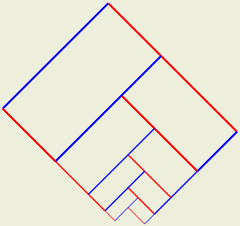
Theorem

local commutation implies commutation, if $\to\, = \textcolor{blue}{\triangleright} \cup \textcolor{red}{\blacktriangleright}$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \textcolor{blue}{\triangleleft} a \overset{\textcolor{blue}{\triangleleft}}{\underset{\textcolor{red}{\blacktriangleright}}{}} a' \textcolor{red}{\blacktriangleright} c$$



Commuting version of Newman's Lemma

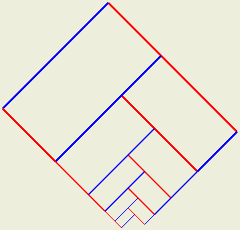
Theorem

local commutation implies commutation, if $\rightarrow\; = \;\triangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \triangleleft a \overset{\triangleleft}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

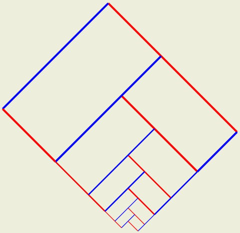
Theorem

local commutation implies commutation, if $\to\ =\ \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \mathrel{\substack{\lhd \\ \blacktriangleright}} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

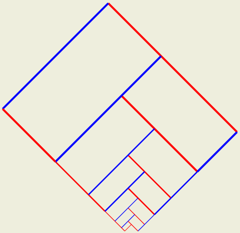
Theorem

local commutation implies commutation, if $\to \,=\, \triangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \triangleleft a \underset{\blacktriangleright}{\triangleleft} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

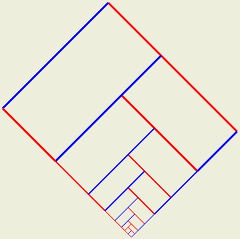
Theorem

local commutation implies commutation, if $\to \; = \; \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \mathrel{\substack{\lhd \\ \blacktriangleright}} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

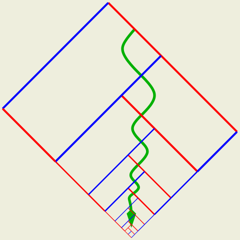
Theorem

local commutation implies commutation, if $\to \; = \; \rhd \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

$$b \lhd a \overset{\lhd}{\blacktriangleright} a' \blacktriangleright c$$



Commuting version of Newman's Lemma

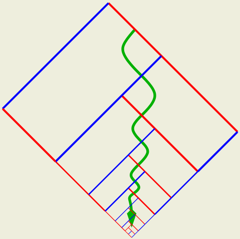
Theorem

local commutation implies commutation, if $\to = \triangleright \cup \blacktriangleright$ *terminating*

Exercise

show that without termination commutation need not hold

exercise: does commutation hold if $\triangleright^+ \cdot \blacktriangleright^+$ terminating?



Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\triangleright^{+} \cdot \blacktriangleright^{+}$ terminating

Proof.

idea: use decreasing diagrams with self-labelling

- label $a \triangleright b$ as $a \triangleright_{a \triangleright b} b$
- label $a \blacktriangleright b$ as $a \blacktriangleright_{a \blacktriangleright b} b$
- order $\succ$ generated by 'reachability':

$$(a \triangleright b) \succ (c \blacktriangleright d), \text{ if } b \twoheadrightarrow c$$

$$(a \blacktriangleright b) \succ (c \triangleright d), \text{ if } b \twoheadrightarrow c$$

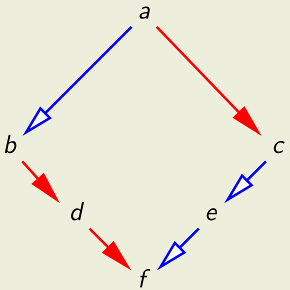
with $\rightarrow = \triangleright \cup \blacktriangleright$, well-founded because of assumption

Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\triangleright^{+} \cdot \blacktriangleright^{+}$ *terminating*

Proof.



$\square$

Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\triangleright^+ \cdot \blacktriangleright^+$ *terminating*

Proof.

$$
\begin{tikzcd}[column sep=small]
 & & a \arrow[dll, "(a \triangleright b)"'] \arrow[drr, "(a \blacktriangleright c)"] & & \\
b \arrow[dr, "(b \blacktriangleright d)"'] & & & & c \arrow[dl, "(c \triangleright e)"] \\
 & d \arrow[dr, "(d \blacktriangleright f)"'] & & e \arrow[dl, "(e \triangleright f)"] & \\
 & & f & &
\end{tikzcd}
$$

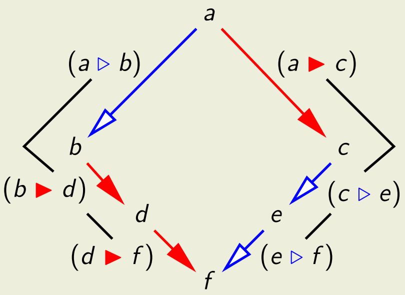
$\square$

Commuting version of Newman's Lemma

Theorem

local commutation implies commutation, if $\rhd^+ \cdot \blacktriangleright^+$ terminating

Proof.



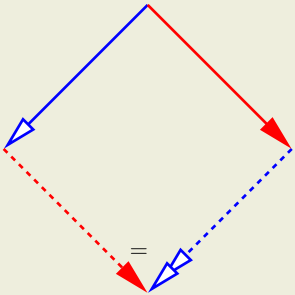
$b \twoheadrightarrow b \twoheadrightarrow d$ and $c \twoheadrightarrow c \twoheadrightarrow e$

$\square$

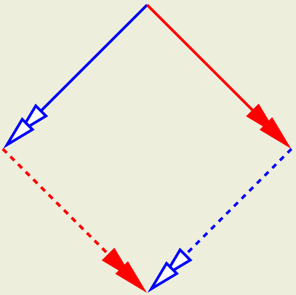
Lemma of Hindley

Theorem (Hindley 1964)

strong commutation implies commutation



$\Rightarrow$



Lemma of Hindley

Proof.

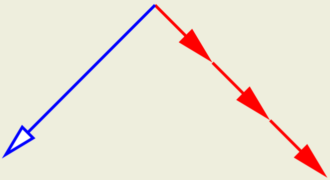
intuition: tiling terminates since only $\triangleright$ steps are split

$\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

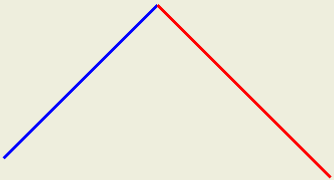


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

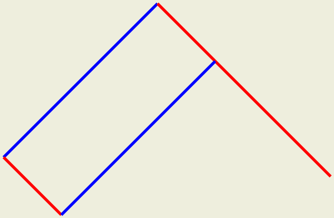


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

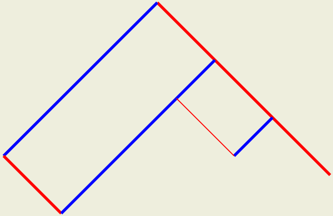


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

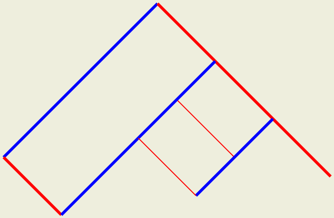


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

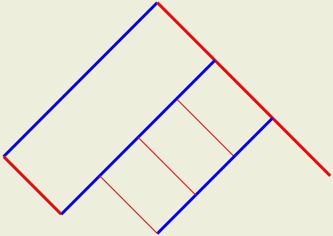


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

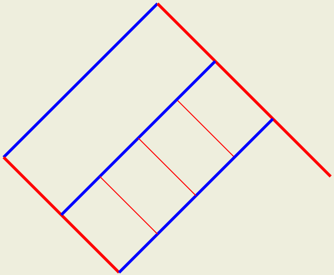


repeat: fill in local peak with local diagram □

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

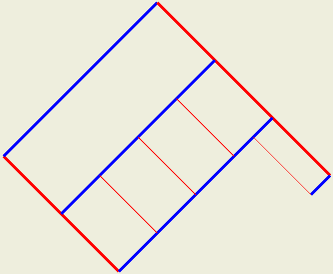


repeat: fill in local peak with local diagram □

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

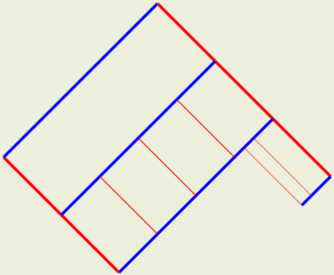


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

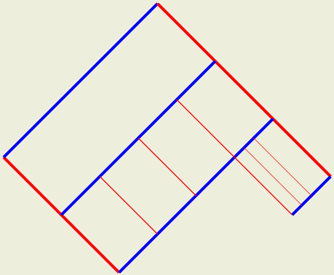


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

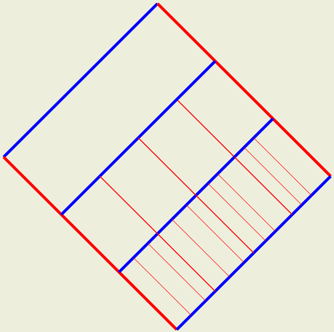


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

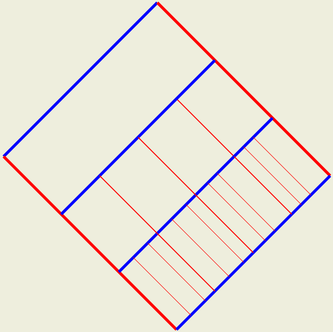


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only $\triangleright$ steps are split

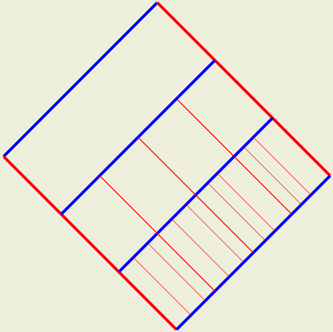


repeat: fill in local peak with local diagram $\square$

Lemma of Hindley

Proof.

intuition: tiling terminates since only ▷ steps are split

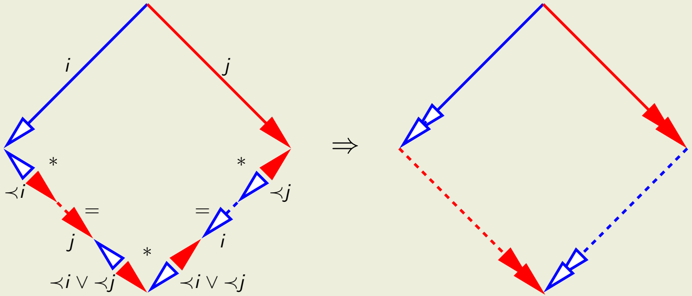


must stop: each ▶ stripe is eventualy filled □

Local decreasingness

Theorem

locally decreasing $\Rightarrow$ *commutation*

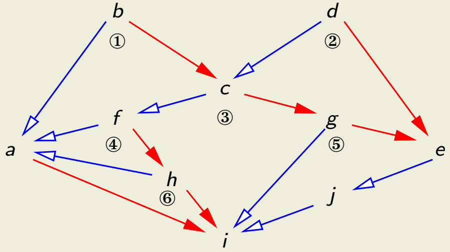


$\rhd = \bigcup_{i \in I} \rhd_i$, $\blacktriangleright = \bigcup_{j \in J} \blacktriangleright_j$, $\prec$ *well-founded order on* $I \cup J$

Locally decreasing?

Exercise

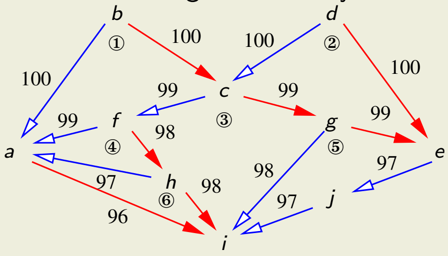
find 4 different labellings; how many labels needed?



Locally decreasing?

Exercise

find 4 different labellings; how many labels needed?

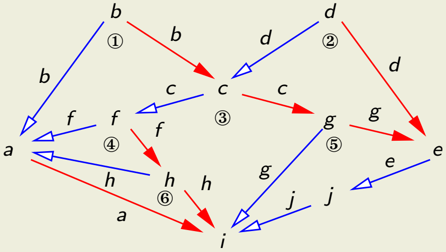


Hans' labelling: top–bottom, high as possible

Locally decreasing?

Exercise

find 4 different labellings; how many labels needed?

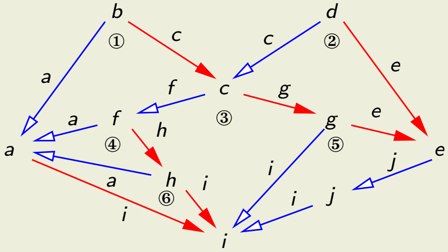


$b \succ d \succ c \succ f \succ g \succ h \succ e \succ a \succ j \succ i$ (topological sort)

Locally decreasing?

Exercise

find 4 different labellings; how many labels needed?

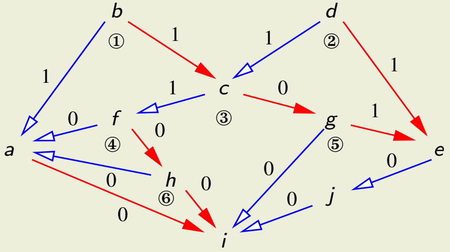


$b \succ d \succ c \succ f \succ g \succ h \succ e \succ a \succ j \succ i$ (topological sort)

Locally decreasing?

Exercise

find 4 different labellings; how many labels needed?

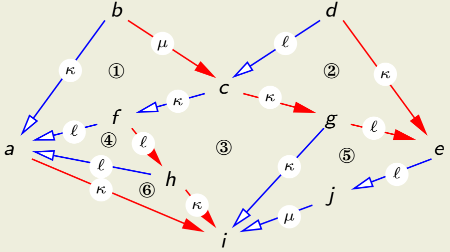


two labels: $1 > 0$

Locally decreasing?

Exercise

find ordering on these labels to make local peaks decreasing



$$\kappa \succ \ell, \mu$$

Locally decreasing?

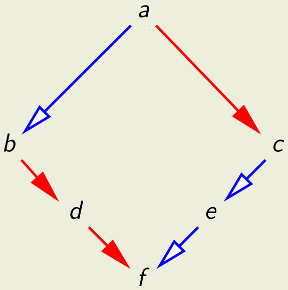
Exercise

- find suitable labelling to prove Newman's Lemma

Locally decreasing?

Exercise

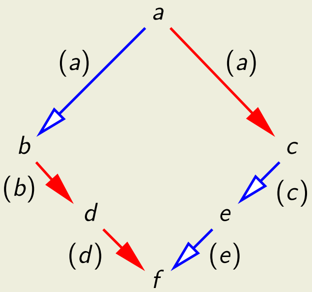
- find suitable labelling to prove Newman's Lemma
  Answer: label steps by source, order by $\to^+$ with $\to = \triangleright \cup \blacktriangleright$



Locally decreasing?

Exercise

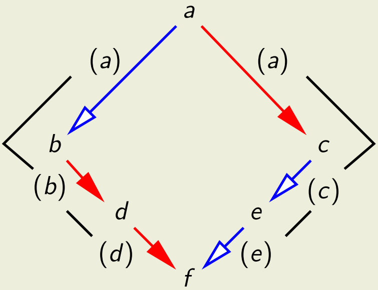
- find suitable labelling to prove Newman's Lemma
  Answer: label steps by source, order by $\rightarrow^{+}$ with $\rightarrow = \triangleright \cup \blacktriangleright$



Locally decreasing?

Exercise

- find suitable labelling to prove Newman's Lemma
  Answer: label steps by source, order by $\to^+$ with $\to = \vartriangleright \cup \blacktriangleright$



$a \to^+ b \to^+ d$ and $a \to^+ c \to^+ e$

Locally decreasing?

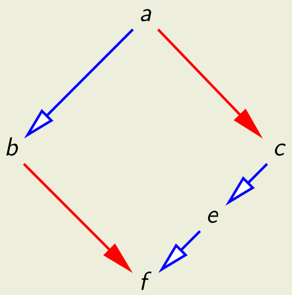
Exercise

- find suitable labelling to prove Hindley's Lemma

Locally decreasing?

Exercise

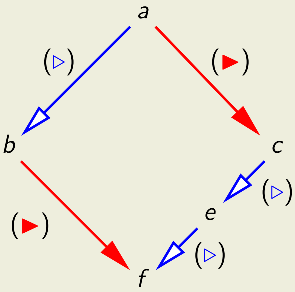
- find suitable labelling to prove Hindley's Lemma
  Answer: order ▶-steps above ▷-steps



Locally decreasing?

Exercise

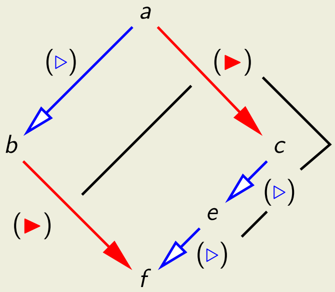
- find suitable labelling to prove Hindley's Lemma
  Answer: order ▶-steps above ▷-steps



Locally decreasing?

Exercise

- find suitable labelling to prove Hindley's Lemma
  Answer: order ▶-steps above ▷-steps

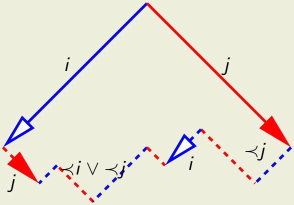


$$(\blacktriangleright) \succ (\triangleright)$$

Local peak may be non-base case

Proof.

local peak may not be base case

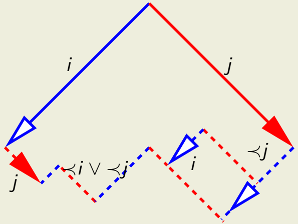


□

Local peak may be non-base case

Proof.

but its peaks can be filled in by induction

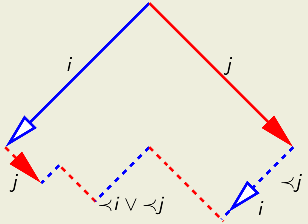


□

Local peak may be non-base case

Proof.

but its peaks can be filled in by induction.

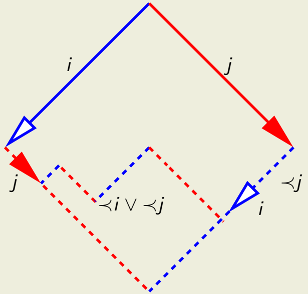


□

Local peak may be non-base case

Proof.

but its peaks can be filled in by induction..

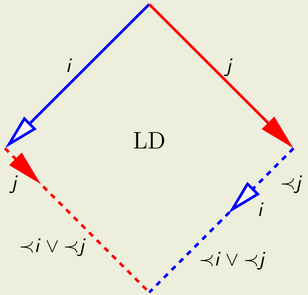


□

Local peak may be non-base case

Proof.

giving in the end a decreasing diagram

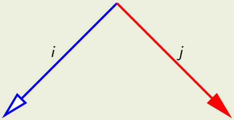


□

Local peak may be non-base case

Proof.

idea: combine label with labels it still has to commute with

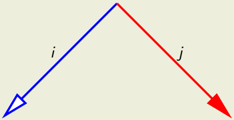


□

Local peak may be non-base case

Proof.

idea: combine label with labels it still has to commute with

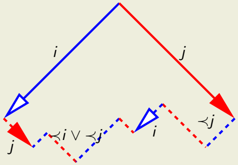


i still has to commute with j; j still has to commute with i □

Local peak may be non-base case

Proof.

idea: combine label with labels it still has to commute with

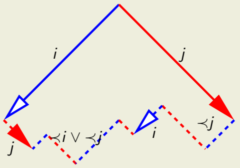


□

Local peak may be non-base case

Proof.

idea: combine label with labels it still has to commute with



i has to commute with $\prec j$
j does not have to commute anymore
$\prec i, j$ have to commute among themselves $\square$

Local peak may be non-base case

Proof.

formally: compare label strings of conversions by $s \gg_{\bullet} t$ with

$$s \gg_{\bullet} t \quad \text{if} \quad \langle s \rangle^{f} \; ((\succ, \gg_{\bullet})_{lex})_{mul} \; \langle t \rangle^{f}$$

- $\langle s \rangle^{f} = [(\acute{\ell}, q) \mid s = p\acute{\ell}q] \cup [(\grave{\ell}, p) \mid s = p\grave{\ell}q]$
  collects acute/grave letters together with suffix/prefix
- $\gg_{mul}$ the multiset extension of $\gg$
- $(\gg_1, \gg_2)_{lex}$ the lexicographic product of $\gg_1$, $\gg_2$

$\square$

Local peak may be non-base case

Proof.

formally: compare label strings of conversions by $s \gg_\bullet t$ with

$$s \gg_\bullet t \quad \text{if} \quad \langle s\rangle^f \; ((\succ, \gg_\bullet)_{lex})_{mul} \; \langle t\rangle^f$$

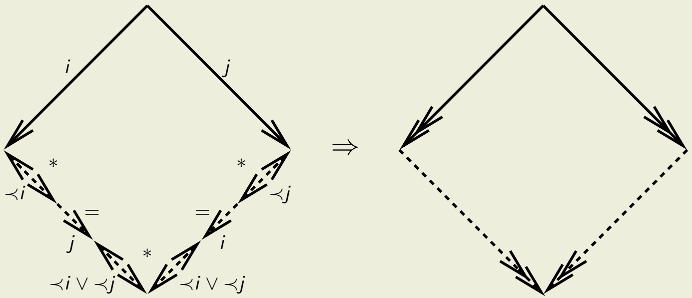
- $\langle s\rangle^f = [(\acute{\ell}, q) \mid s = p\acute{\ell}q] \cup [(\grave{\ell}, p) \mid s = p\grave{\ell}q]$
  collects acute/grave letters together with suffix/prefix
- $\gg_{mul}$ the multiset extension of $\gg$
- $(\gg_1, \gg_2)_{lex}$ the lexicographic product of $\gg_1$, $\gg_2$

$\gg_\bullet$ well-founded prooforder $\square$

Locally decreasing for self

Theorem

locally decreasing $\Rightarrow$ *confluence*

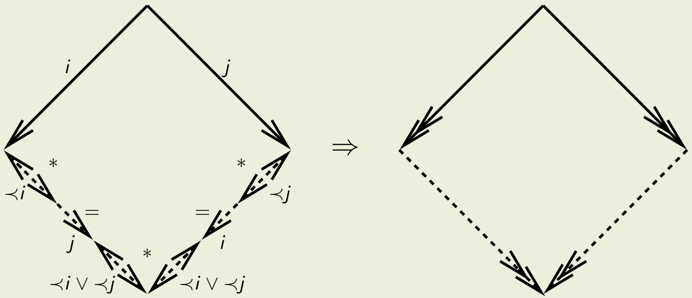


$\to = \bigcup_{i \in I} \to_i$, $\prec$ *well-founded order on* I

Locally decreasing for self

Theorem

locally decreasing $\Rightarrow$ *confluence*



$\rightarrow = \bigcup_{i \in I} \rightarrow_i$, $\prec$ *well-founded order on* I

complete for countable systems

(In)completeness of decreasing diagrams

Theorem

if a countable rewrite relation is confluent, then it can be proven so by decreasing diagrams.
only 2 labels are needed

(In)completeness of decreasing diagrams

decreasing diagrams is incomplete for commutation

Example

$$d \blacktriangleleft b \triangleleft a_1 \overset{\blacktriangleright}{\triangleleft} a_2 \blacktriangleright c \triangleright d$$

Proof by contradiction.

consider triples of shape $b \triangleleft_i a_1 \overset{\blacktriangleright}{\triangleleft}_j a_2 \blacktriangleright_k c$ with labels $[i, j, k]$. suppose w.l.o.g. $a_1 \blacktriangleright_j a_2$. then $b \triangleleft_i a_1 \blacktriangleright_j a_2$ can only be closed by $b \triangleleft_{i'} a_1 \triangleleft_{j'} a_2$. distinguish cases on the origin of the label j':

- if $j' < j$, then consider the triple with labels $[i, j', k]$.
- suppose $j' = i$. if $i' < i$ consider the triple with labels $[i', j, k]$, else $i' < j$ and consider the triple with labels $[i', i, k]$.

$\square$

(In)completeness of decreasing diagrams

decreasing diagrams is incomplete for commutation

Example

$$d \blacktriangleleft b \triangleleft a_1 \overset{\blacktriangleright}{\triangleleft} a_2 \blacktriangleright c \triangleright d$$

Proof by contradiction.

consider triples of shape $b \triangleleft_i a_1 \overset{\blacktriangleright}{\triangleleft}_j a_2 \blacktriangleright_k c$ with labels $[i, j, k]$. suppose w.l.o.g. $a_1 \blacktriangleright_j a_2$. then $b \triangleleft_i a_1 \blacktriangleright_j a_2$ can only be closed by $b \triangleleft_{i'} a_1 \triangleleft_{j'} a_2$. distinguish cases on the origin of the label j':

- if $j' < j$, then consider the triple with labels $[i, j', k]$.
- suppose $j' = i$. if $i' < i$ consider the triple with labels $[i', j, k]$, else $i' < j$ and consider the triple with labels $[i', i, k]$.

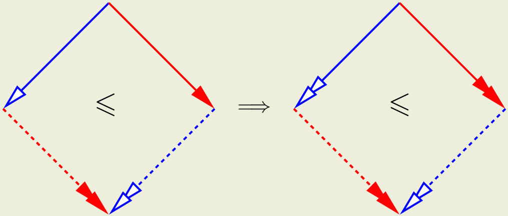
$\square$

even if works, arbitrarily many labels may be needed.

Ordered commutation

Theorem

ordered local commutation $\implies$ *ordered commutation*

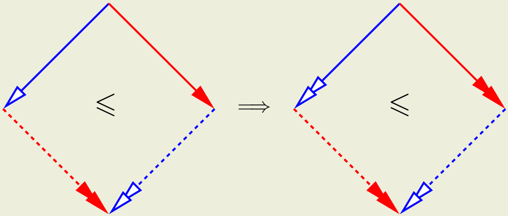


how to read: either joinable, or the longer side is infinite

Ordered commutation

Theorem

ordered local commutation $\Longrightarrow$ $\triangleright$ *is better than* $\blacktriangleright$



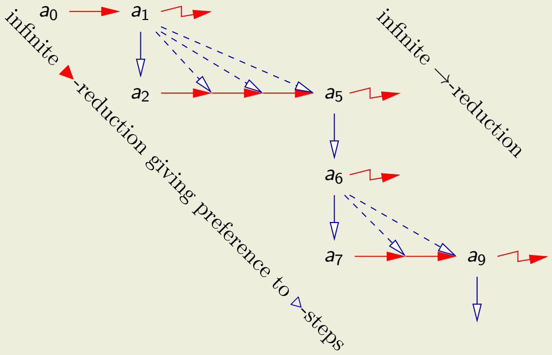
how to read: either joinable, or the longer side is infinite

Lazy commutation

Theorem

if $\triangleright \cdot \blacktriangleright \subseteq \triangleright \cup (\blacktriangleright \cdot \twoheadrightarrow)$, *then* $\rightarrow = \triangleright \cup \blacktriangleright$ *terminating if* $\triangleright, \blacktriangleright$ *are*

Proof.



$\square$

Theorem

if $\triangleright$, $\blacktriangleright$ are TRSs, $\triangleright$ is right-linear, $\blacktriangleright$ is left-linear and no overlap $\triangleright \cup \blacktriangleright$ is terminating iff $\triangleright$,$\blacktriangleright$ terminating

Proof.

by lazy commutation □