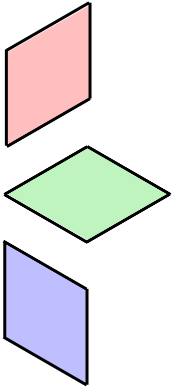
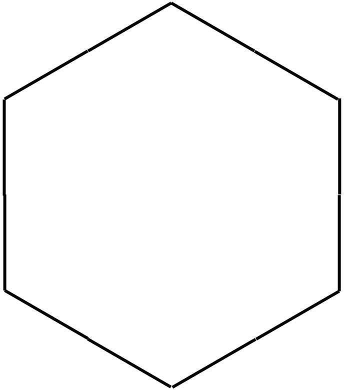
The problem of the calissons, by rewriting

Vincent van Oostrom
University of Sussex
vvo@sussex.ac.uk

The problem of the calissons (David & Tomei 89)

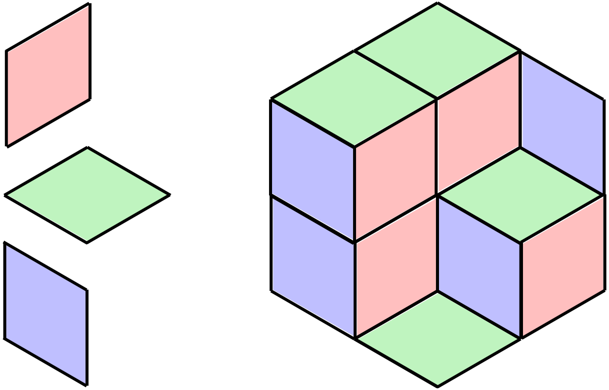


calissons

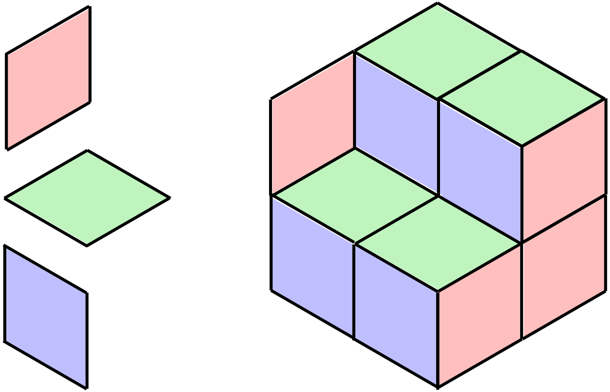


hexagonal box

The problem of the calissons

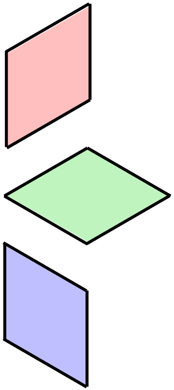


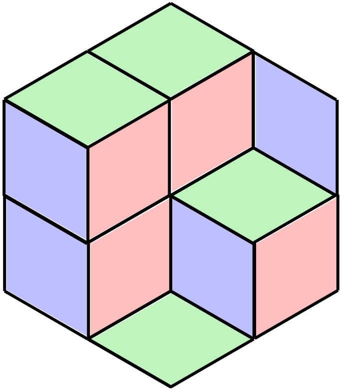
The problem of the calissons



box random generator

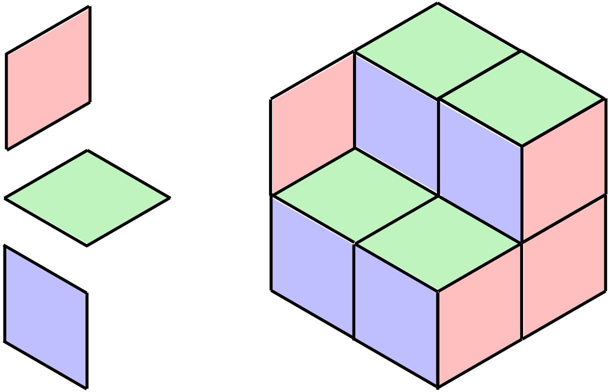
The problem of the calissons





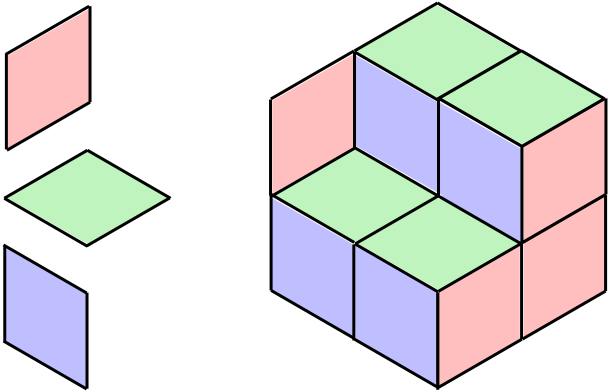
spectrum $(4, 4, 4)$

The problem of the calissons



spectrum $(4, 4, 4)$

The problem of the calissons by 4 confluence techniques



same box $\Longrightarrow$ same spectrum

Confluence

Definition

rewrite system $\to := \langle A, \Phi, \mathrm{src}, \mathrm{tgt} \rangle$ with objects A and steps Φ

$\phi : \; a \to b$ or $a \to_\phi b$ denotes step ϕ with source $\mathrm{src}(\phi) = a$, target $\mathrm{tgt}(\phi) = b$

Confluence

Definition

rewrite system $\rightarrow := \langle A, \Phi, \mathrm{src}, \mathrm{tgt} \rangle$ with objects A and steps Φ

rewrite systems have same data as multigraphs, quivers, pre-categories

Confluence

Definition

rewrite system $\to$ is confluent (CR) if $\forall a\, b\, c,\ b \twoheadleftarrow a \twoheadrightarrow c \implies \exists d,\ b \twoheadrightarrow d \twoheadleftarrow c$

$\twoheadrightarrow$ denotes a (finite) reduction; a sequence of consecutive steps of $\to$

CR after Chuch and Rosser for introducing and proving it, for $\lambda\beta$ (1936)

Confluence

Definition

rewrite system $\rightarrow$ is confluent (CR) if $\forall a\, b\, c,\ b \twoheadleftarrow a \twoheadrightarrow c \implies \exists d,\ b \twoheadrightarrow d \twoheadleftarrow c$

$$
\begin{tikzcd}
 & a \arrow[dl] \arrow[dr] & \\
b \arrow[dr, dotted] & \text{CR} & c \arrow[dl, dotted] \\
 & d &
\end{tikzcd}
$$

Confluence

Definition

rewrite system $\to$ is confluent (CR) if $\forall a\, b\, c,\ b \twoheadleftarrow a \twoheadrightarrow c \implies \exists d,\ b \twoheadrightarrow d \twoheadleftarrow c$

$$\begin{tikzcd}
 & a \arrow[dl, "\phi"'] \arrow[dr, "\psi"] & \\
b \arrow[dr, dotted, "\psi'"'] & \phi \cdot \psi' = \psi \cdot \phi' & c \arrow[dl, dotted, "\phi'"] \\
 & d &
\end{tikzcd}$$

algebra: existence of common multiple; concatenation is (typed) multiplication

Confluence

Definition

rewrite system $\to$ is confluent (CR) if $\forall a\, b\, c,\ b \twoheadleftarrow a \twoheadrightarrow c \implies \exists d,\ b \twoheadrightarrow d \twoheadleftarrow c$

$$\begin{tikzcd}
 & a \arrow[dl] \arrow[dr] & \\
b \arrow[dr, dotted] & a \twoheadrightarrow b, c \twoheadrightarrow d & c \arrow[dl, dotted] \\
 & d &
\end{tikzcd}$$

orders: existence of upperbound; $\twoheadrightarrow$ is quasi-order

Confluence

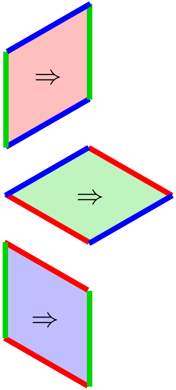
Definition

rewrite system $\to$ is confluent (CR) if $\forall a\, b\, c,\; b \twoheadleftarrow a \twoheadrightarrow c \implies \exists d,\; b \twoheadrightarrow d \twoheadleftarrow c$

$$\begin{tikzcd}
 & a \arrow[dl, "\phi"'] \arrow[dr, "\psi"] \arrow[dd, phantom, "\mathrm{CR}"] & \\
b \arrow[dr, dotted, "\psi/\phi"'] & & c \arrow[dl, dotted, "\phi/\psi"] \\
 & d &
\end{tikzcd}$$

categories: having a pushout (axioms); skolemisation is residuation $/$ (after)

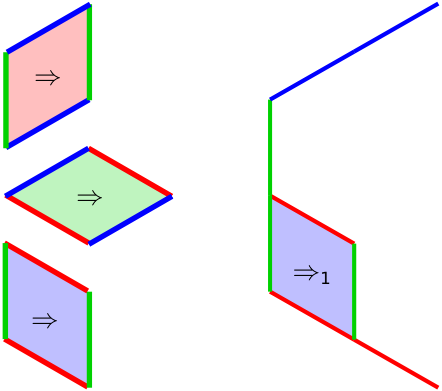
(1) random descent



filling rules

filling

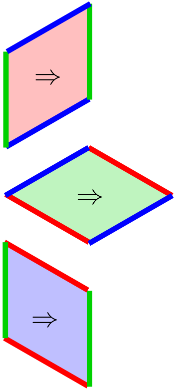
(1) random descent



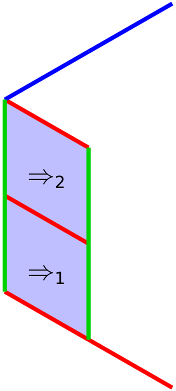
filling rules

filling

(1) random descent

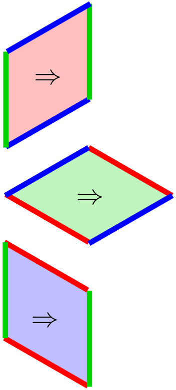


filling rules

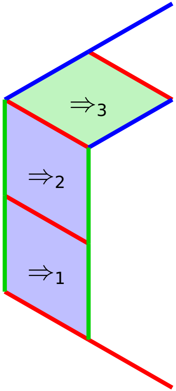


filling

(1) random descent

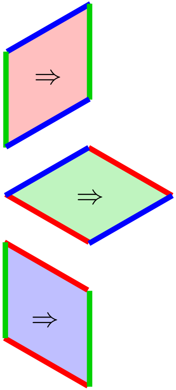


filling rules

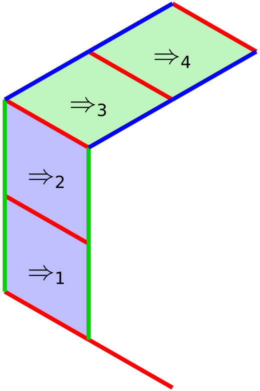


filling

(1) random descent

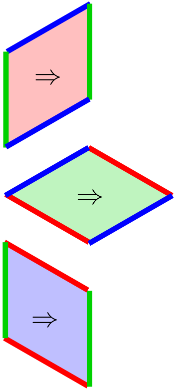


filling rules

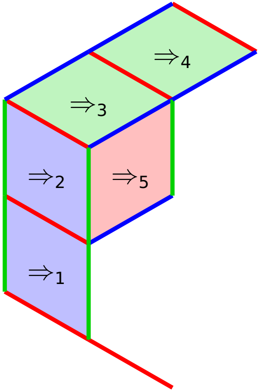


filling

(1) random descent

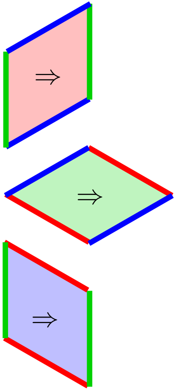


filling rules

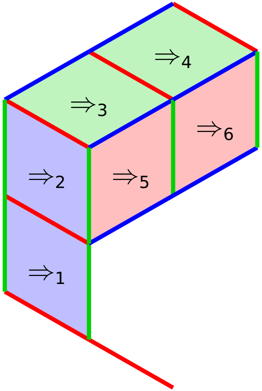


filling

(1) random descent

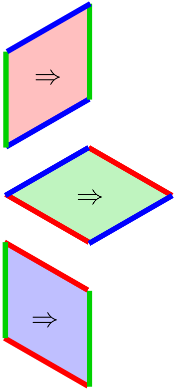


filling rules

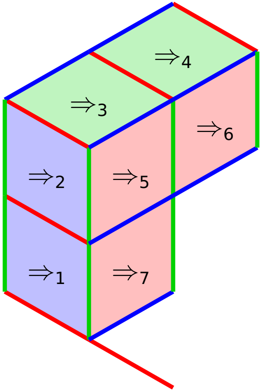


filling

(1) random descent

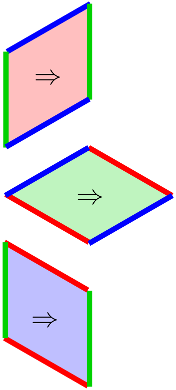


filling rules

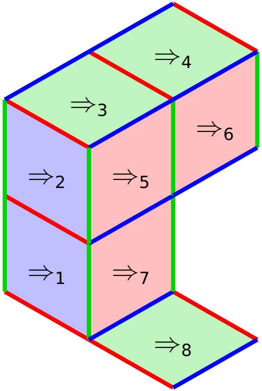


filling

(1) random descent

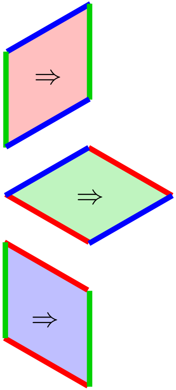


filling rules

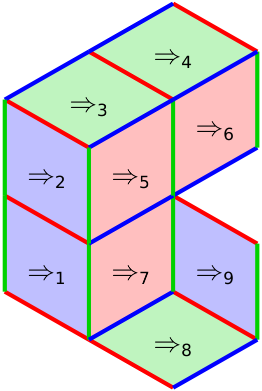


filling

(1) random descent

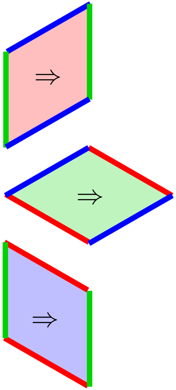


filling rules

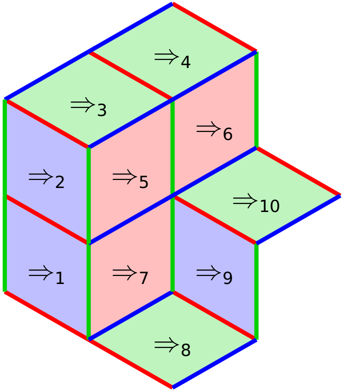


filling

(1) random descent

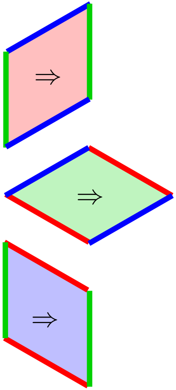


filling rules

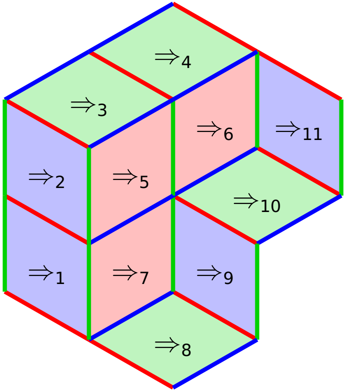


filling

(1) random descent

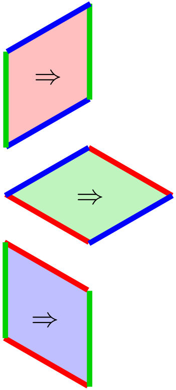


filling rules

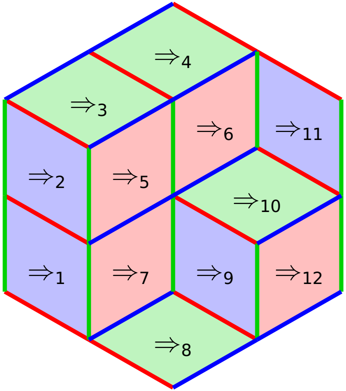


filling

(1) random descent

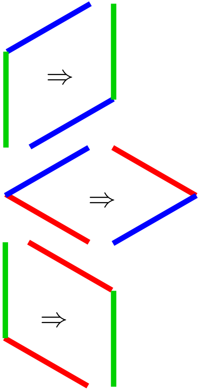


filling rules



filling

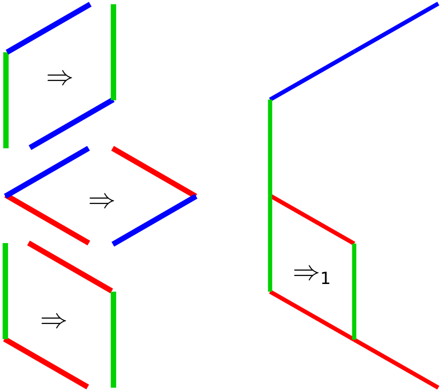
(1) random descent



string rewrite rules

string rewrite

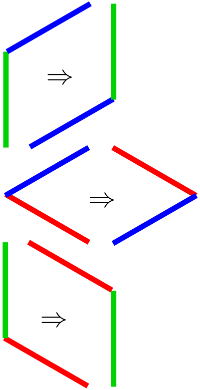
(1) random descent



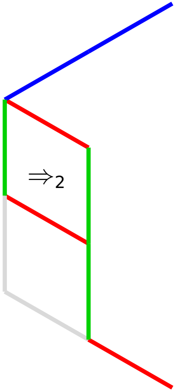
string rewrite rules

string rewrite

(1) random descent

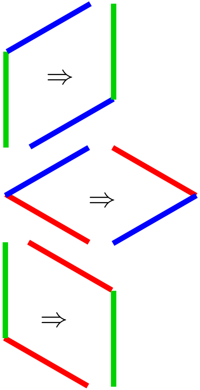


string rewrite rules

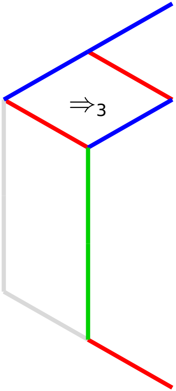


string rewrite

(1) random descent

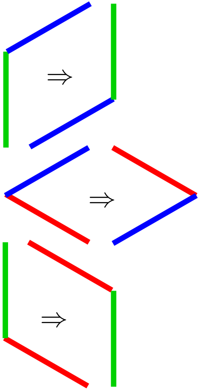


string rewrite rules

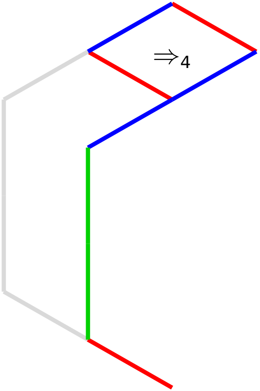


string rewrite

(1) random descent

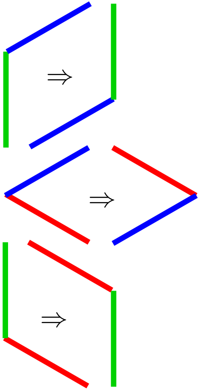


string rewrite rules

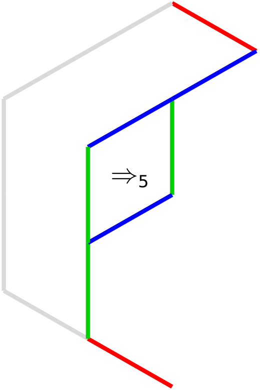


string rewrite

(1) random descent

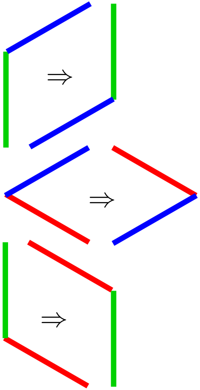


string rewrite rules

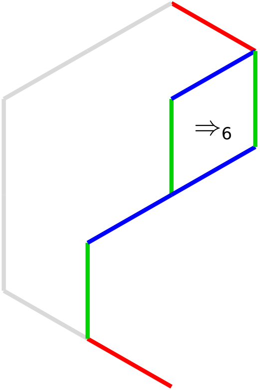


string rewrite

(1) random descent

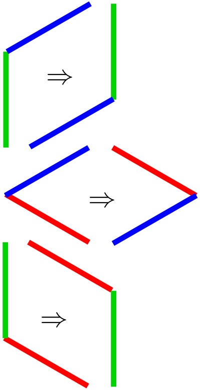


string rewrite rules

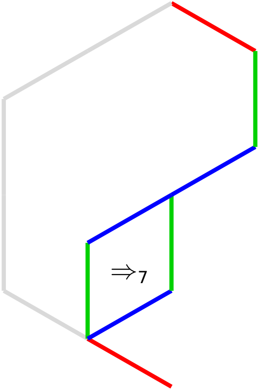


string rewrite

(1) random descent

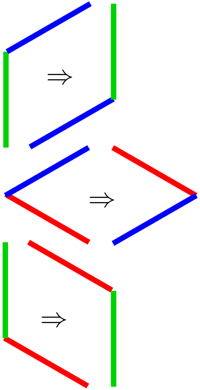


string rewrite rules

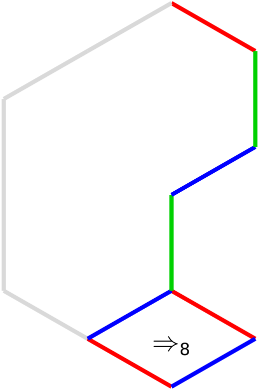


string rewrite

(1) random descent

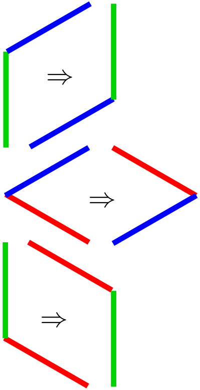


string rewrite rules

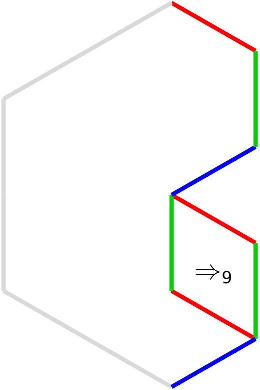


string rewrite

(1) random descent

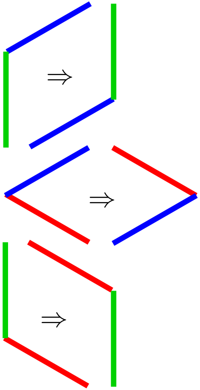


string rewrite rules

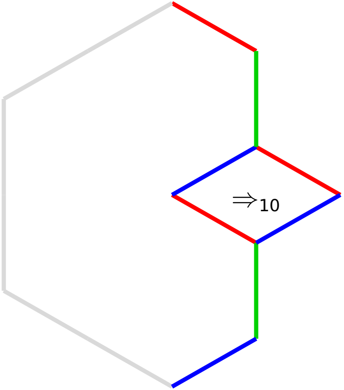


string rewrite

(1) random descent

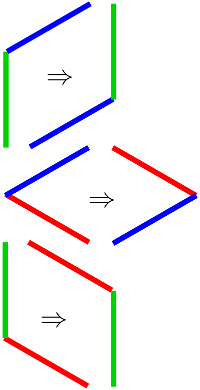


string rewrite rules

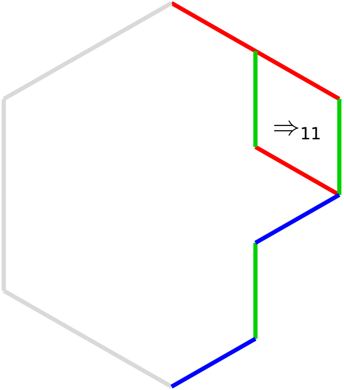


string rewrite

(1) random descent

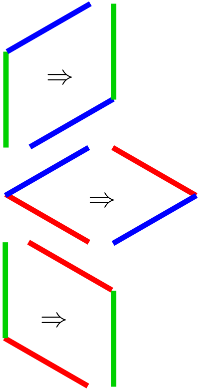


string rewrite rules

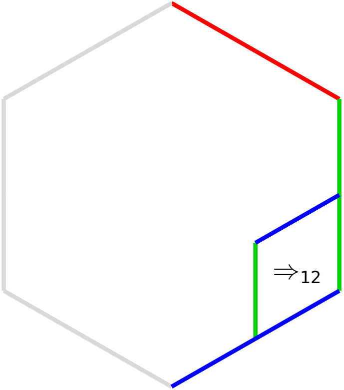


string rewrite

(1) random descent

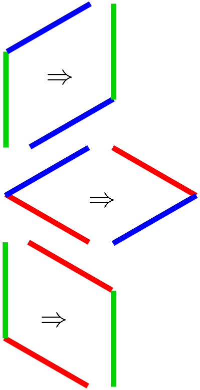


string rewrite rules

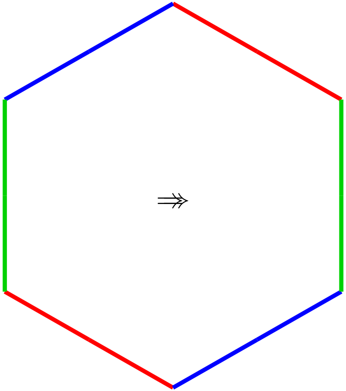


string rewrite

(1) random descent



string rewrite rules



string rewrite

(1) random descent

- filling $\Rightarrow$ is string rewrite system over  with rules

(recover hexagonal shape from associating colours to angles of lines; Logo)

(1) random descent

- filling $\Rightarrow$ is string rewrite system over $\{$$\}$ with rules

- filled box B iff exists  $\Rrightarrow$  filling B
  (any partial filling allows some filling step toward that B)

(1) random descent

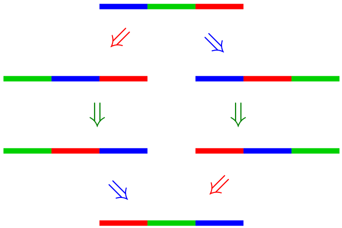
- filling $\Rightarrow$ is string rewrite system over $\{$, , $\}$ with rules

- filled box B iff exists $\twoheadrightarrow$ filling B
- filling $\Rightarrow$ has random descent (RD) for measure on steps

$$\Rightarrow \;\mapsto (1,0,0) \qquad \Rightarrow \;\mapsto (0,1,0) \qquad \Rightarrow \;\mapsto (0,0,1)$$

(measure: mapping steps to (non-zero) elements of a derivation monoid)

critical peak



legs same measure (1,1,1)

(1) random descent

- filling $\Rightarrow$ is string rewrite system over $\{{\color{red}\rule{1.5em}{0.4em}}, {\color{green}\rule{1.5em}{0.4em}}, {\color{blue}\rule{1.5em}{0.4em}}\}$ with rules

$$
{\color{blue}\rule{1.5em}{0.4em}}{\color{green}\rule{1.5em}{0.4em}} \;{\color{red}\Rightarrow}\; {\color{green}\rule{1.5em}{0.4em}}{\color{blue}\rule{1.5em}{0.4em}} \qquad {\color{blue}\rule{1.5em}{0.4em}}{\color{red}\rule{1.5em}{0.4em}} \;{\color{green}\Rightarrow}\; {\color{red}\rule{1.5em}{0.4em}}{\color{blue}\rule{1.5em}{0.4em}} \qquad {\color{green}\rule{1.5em}{0.4em}}{\color{red}\rule{1.5em}{0.4em}} \;{\color{blue}\Rightarrow}\; {\color{red}\rule{1.5em}{0.4em}}{\color{green}\rule{1.5em}{0.4em}}
$$

- filled box B iff exists ${\color{blue}\rule{3em}{0.4em}}{\color{green}\rule{3em}{0.4em}}{\color{red}\rule{3em}{0.4em}} \Rightarrow\!\!\!\Rightarrow {\color{red}\rule{3em}{0.4em}}{\color{green}\rule{3em}{0.4em}}{\color{blue}\rule{3em}{0.4em}}$ filling B
- filling $\Rightarrow$ has random descent (RD) for measure on steps

$$
{\color{red}\Rightarrow} \mapsto ({\color{red}1}, {\color{green}0}, {\color{blue}0}) \qquad {\color{green}\Rightarrow} \mapsto ({\color{red}0}, {\color{green}1}, {\color{blue}0}) \qquad {\color{blue}\Rightarrow} \mapsto ({\color{red}0}, {\color{green}0}, {\color{blue}1})
$$

  RD: if reduction ends in nf then all maximal such do with same measure

(1) random descent

- filling $\Rightarrow$ is string rewrite system over $\{\textcolor{red}{\rule{2em}{0.4em}}, \textcolor{green}{\rule{2em}{0.4em}}, \textcolor{blue}{\rule{2em}{0.4em}}\}$ with rules

$$\textcolor{blue}{\rule{2em}{0.4em}}\textcolor{green}{\rule{2em}{0.4em}} \textcolor{red}{\Rightarrow} \textcolor{green}{\rule{2em}{0.4em}}\textcolor{blue}{\rule{2em}{0.4em}} \qquad \textcolor{blue}{\rule{2em}{0.4em}}\textcolor{red}{\rule{2em}{0.4em}} \textcolor{green}{\Rightarrow} \textcolor{red}{\rule{2em}{0.4em}}\textcolor{blue}{\rule{2em}{0.4em}} \qquad \textcolor{green}{\rule{2em}{0.4em}}\textcolor{red}{\rule{2em}{0.4em}} \textcolor{blue}{\Rightarrow} \textcolor{red}{\rule{2em}{0.4em}}\textcolor{green}{\rule{2em}{0.4em}}$$

- filled box B iff exists $\textcolor{blue}{\rule{4em}{0.4em}}\textcolor{green}{\rule{4em}{0.4em}}\textcolor{red}{\rule{4em}{0.4em}} \Rrightarrow \textcolor{red}{\rule{4em}{0.4em}}\textcolor{green}{\rule{4em}{0.4em}}\textcolor{blue}{\rule{4em}{0.4em}}$ filling B
- filling $\Rightarrow$ has random descent (RD) for measure on steps

$$\textcolor{red}{\Rightarrow} \mapsto (\textcolor{red}{1}, \textcolor{green}{0}, \textcolor{blue}{0}) \qquad \textcolor{green}{\Rightarrow} \mapsto (\textcolor{red}{0}, \textcolor{green}{1}, \textcolor{blue}{0}) \qquad \textcolor{blue}{\Rightarrow} \mapsto (\textcolor{red}{0}, \textcolor{green}{0}, \textcolor{blue}{1})$$

  RD: if reduction ends in nf then all maximal such do with same spectrum
- filling $\Rightarrow$ is weakly normalising (WN) so filling fills
  ($\Rightarrow$ is sorting-by-swapping; bubblesort shows WN)

(1) random descent

- filling $\Rightarrow$ is string rewrite system over $\{\textcolor{red}{\rule{2em}{0.5em}}, \textcolor{green}{\rule{2em}{0.5em}}, \textcolor{blue}{\rule{2em}{0.5em}}\}$ with rules

$$\textcolor{blue}{\rule{2em}{0.5em}}\textcolor{green}{\rule{2em}{0.5em}} \textcolor{red}{\Rightarrow} \textcolor{green}{\rule{2em}{0.5em}}\textcolor{blue}{\rule{2em}{0.5em}} \qquad \textcolor{blue}{\rule{2em}{0.5em}}\textcolor{red}{\rule{2em}{0.5em}} \textcolor{green}{\Rightarrow} \textcolor{red}{\rule{2em}{0.5em}}\textcolor{blue}{\rule{2em}{0.5em}} \qquad \textcolor{green}{\rule{2em}{0.5em}}\textcolor{red}{\rule{2em}{0.5em}} \textcolor{blue}{\Rightarrow} \textcolor{red}{\rule{2em}{0.5em}}\textcolor{green}{\rule{2em}{0.5em}}$$

- filled box B iff exists $\textcolor{blue}{\rule{4em}{0.5em}}\textcolor{green}{\rule{4em}{0.5em}}\textcolor{red}{\rule{4em}{0.5em}} \Rrightarrow \textcolor{red}{\rule{4em}{0.5em}}\textcolor{green}{\rule{4em}{0.5em}}\textcolor{blue}{\rule{4em}{0.5em}}$ filling B
- filling $\Rightarrow$ has random descent (RD) for measure on steps

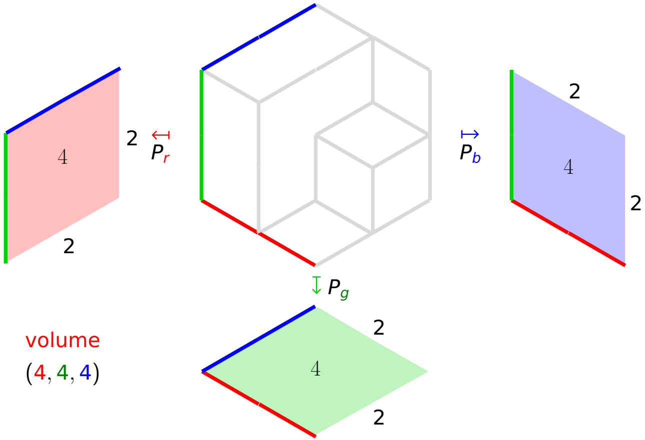
$$\textcolor{red}{\Rightarrow} \mapsto (\textcolor{red}{1}, \textcolor{green}{0}, \textcolor{blue}{0}) \qquad \textcolor{green}{\Rightarrow} \mapsto (\textcolor{red}{0}, \textcolor{green}{1}, \textcolor{blue}{0}) \qquad \textcolor{blue}{\Rightarrow} \mapsto (\textcolor{red}{0}, \textcolor{green}{0}, \textcolor{blue}{1})$$

  RD: if reduction ends in nf then all maximal such do with same spectrum
- filling $\Rightarrow$ is weakly normalising (WN) so filling fills

Theorem ($\Vvee$ 22 ; more on it in second half)

confluent & terminating (SN) $\iff$ *random descent & normalising (WN)*

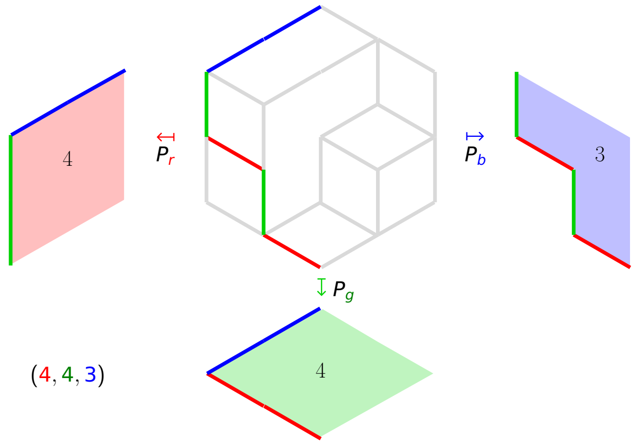
(2) proof order



volume

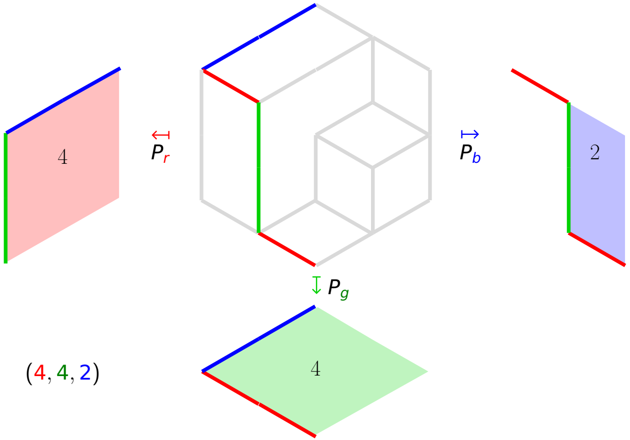
$(4, 4, 4)$

(2) proof order



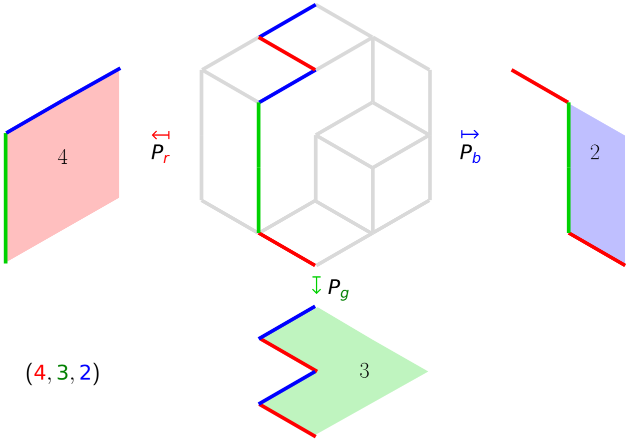
$(4, 4, 3)$

(2) proof order

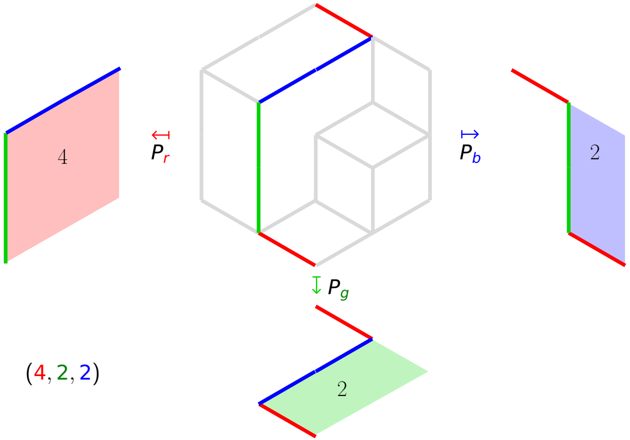


(4, 4, 2)

(2) proof order

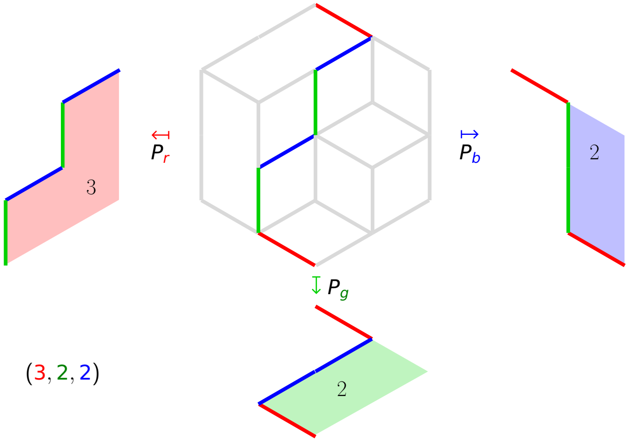


(2) proof order



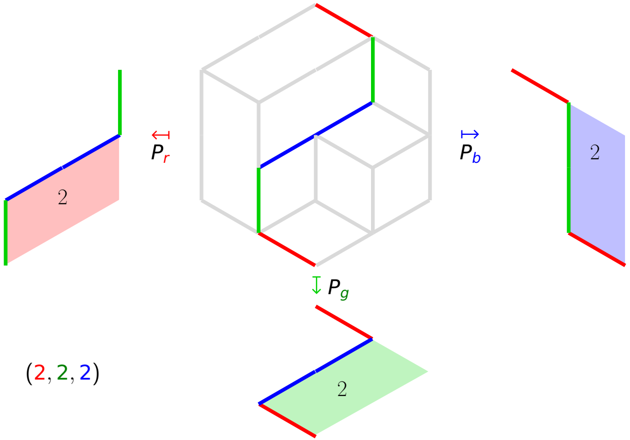
$(4, 2, 2)$

(2) proof order

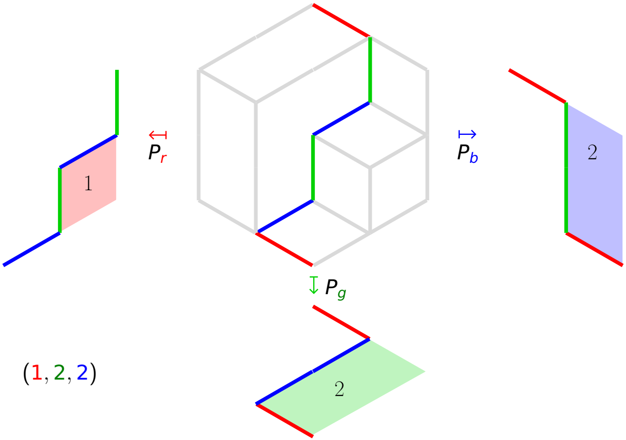


$(3, 2, 2)$

(2) proof order

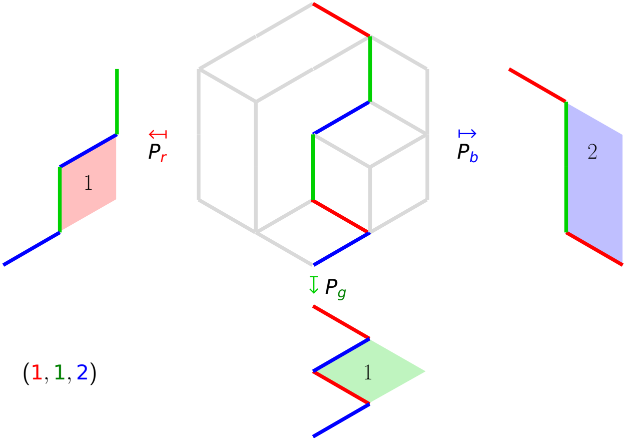


(2) proof order



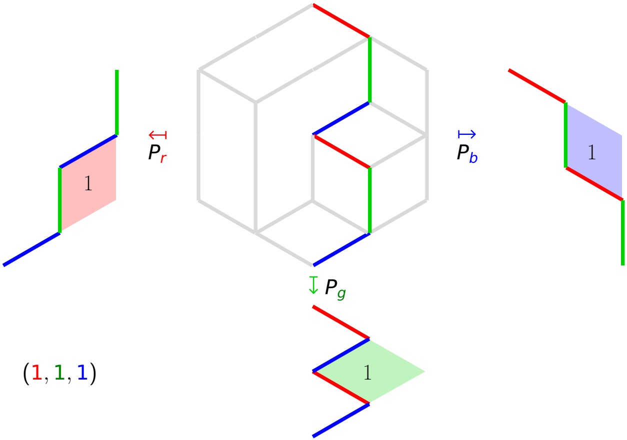
$(1, 2, 2)$

(2) proof order

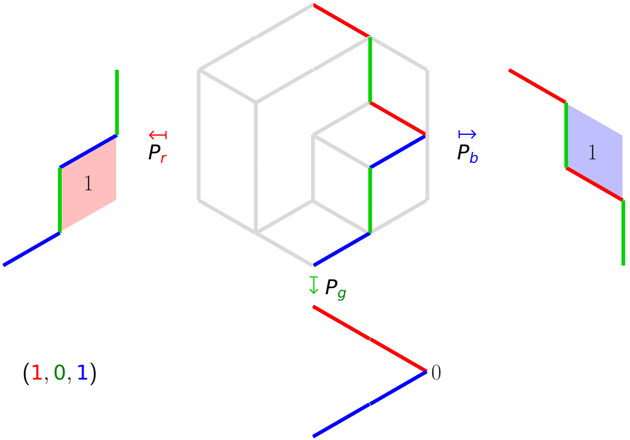


$(1, 1, 2)$

(2) proof order

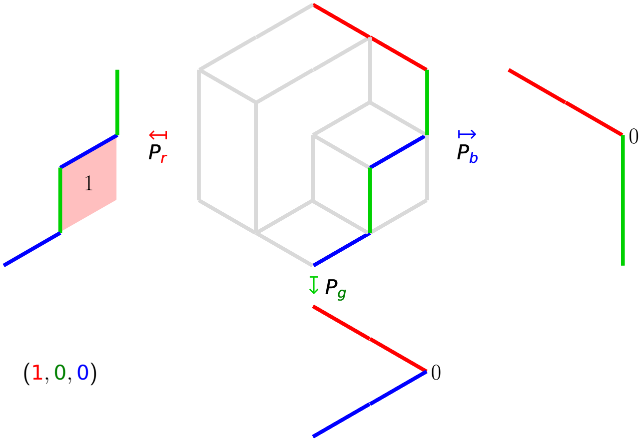


(2) proof order



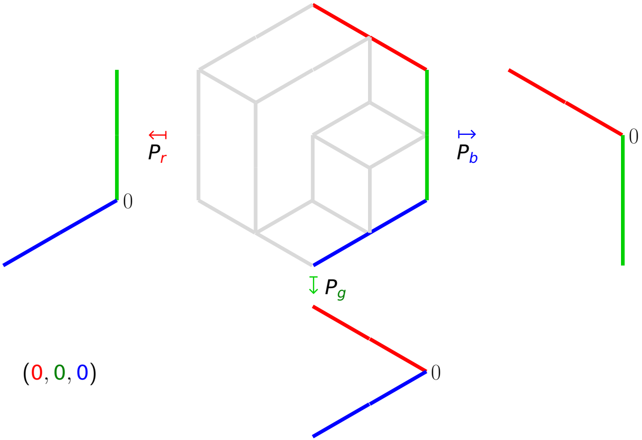
$(1, 0, 1)$

(2) proof order



$(1, 0, 0)$

(2) proof order



$(0, 0, 0)$

(2) proof order

- filling $\Rightarrow$ is string rewrite system over $\{$$\}$ with rules

- filled box B iff exists  filling B
- filling $\Rightarrow$ is WN so filling fills

(2) proof order

- filling $\Rightarrow$ is string rewrite system over $\{\textcolor{red}{\rule{1em}{0.3em}}, \textcolor{green}{\rule{1em}{0.3em}}, \textcolor{blue}{\rule{1em}{0.3em}}\}$ with rules

$$\textcolor{blue}{\rule{1em}{0.3em}}\textcolor{green}{\rule{1em}{0.3em}} \;\textcolor{red}{\Rightarrow}\; \textcolor{green}{\rule{1em}{0.3em}}\textcolor{blue}{\rule{1em}{0.3em}} \qquad \textcolor{blue}{\rule{1em}{0.3em}}\textcolor{red}{\rule{1em}{0.3em}} \;\textcolor{green}{\Rightarrow}\; \textcolor{red}{\rule{1em}{0.3em}}\textcolor{blue}{\rule{1em}{0.3em}} \qquad \textcolor{green}{\rule{1em}{0.3em}}\textcolor{red}{\rule{1em}{0.3em}} \;\textcolor{blue}{\Rightarrow}\; \textcolor{red}{\rule{1em}{0.3em}}\textcolor{green}{\rule{1em}{0.3em}}$$

- filled box B iff exists $\textcolor{blue}{\rule{2em}{0.3em}}\textcolor{green}{\rule{2em}{0.3em}}\textcolor{red}{\rule{2em}{0.3em}} \Rrightarrow \textcolor{red}{\rule{2em}{0.3em}}\textcolor{green}{\rule{2em}{0.3em}}\textcolor{blue}{\rule{2em}{0.3em}}$ filling B
- filling $\Rightarrow$ is WN so filling fills
- filling $\Rightarrow$ decrements (one component of) volume $(\textcolor{red}{r}, \textcolor{green}{g}, \textcolor{blue}{b})$ of path P
  (volume of trichrome path P: triple of areas of projections $P_{\textcolor{red}{r}}, P_{\textcolor{green}{g}}, P_{\textcolor{blue}{b}}$
  area of dichrome path P: #missing calissons)

(2) proof order

- filling $\Rightarrow$ is string rewrite system over $\{$$\}$ with rules

- filled box B iff exists  $\Rrightarrow$  filling B
- filling $\Rightarrow$ is WN so filling fills
- filling $\Rightarrow$ decrements volume (r, g, b) of path P so SN
- volume of normal form path is $(0, 0, 0)$ so spectrum $=$ volume of initial path (initial path only depends on hexagon / box, not on filling / filled box)

(2) proof order

- filling $\Rightarrow$ is string rewrite system over $\{\ ,\ ,\ \}$ with rules

- filled box B iff exists $\Rightarrow\!\!\!\Rightarrow$ filling B
- filling $\Rightarrow$ is WN so filling fills
- filling $\Rightarrow$ decrements volume (r, g, b) of path P so SN
- volume of normal form path is $(0, 0, 0)$ so spectrum $=$ volume of initial path

remark

proof order (Bachmaier & Dershowitz 94) as involutive monoid homomorphism
area proof order to triple (ℓ, a, r) with #missing calissons a (Felgenhauer & 13)

(2) proof order

- filling $\Rightarrow$ is string rewrite system over $\{\ ,\ ,\ \}$ with rules

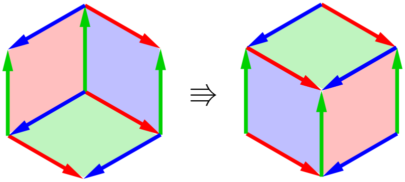
  $\Rightarrow$ $\qquad$ $\Rightarrow$ $\qquad$ $\Rightarrow$

- filled box B iff exists $\Rrightarrow$ filling B
- filling $\Rightarrow$ is WN so filling fills
- filling $\Rightarrow$ decrements volume (r, g, b) of path P so SN
- volume of normal form path is $(0, 0, 0)$ so spectrum $=$ volume of initial path

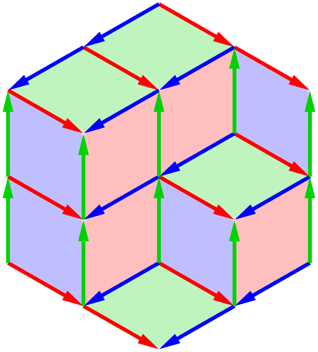
remark

proof order as involutive monoid homomorphism
area proof order to triple (ℓ, a, r) with #missing calissons a
proofs by random descent and proof order show spectrum independent of filling
but can different fillings be related?

(3) bricklaying

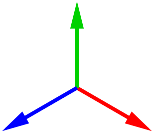


bricklaying rule

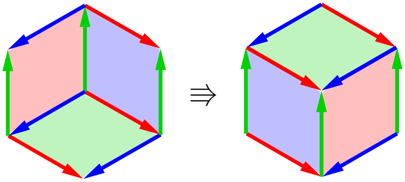


bricklaying step possible?

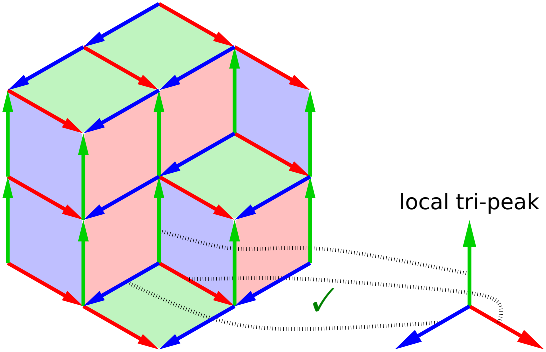
local tri-peak



(3) bricklaying

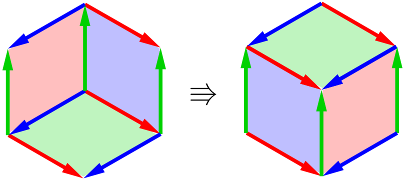


bricklaying rule

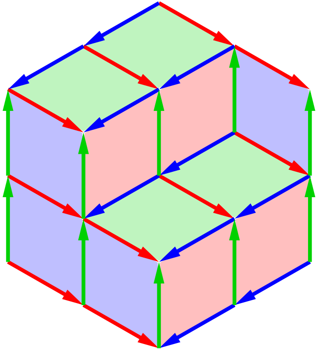


bricklaying step

(3) bricklaying

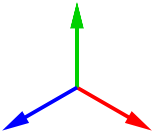


bricklaying rule

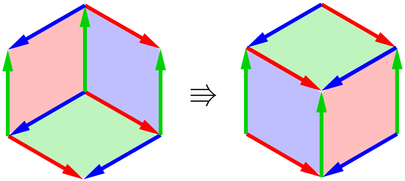


bricklaying step possible?

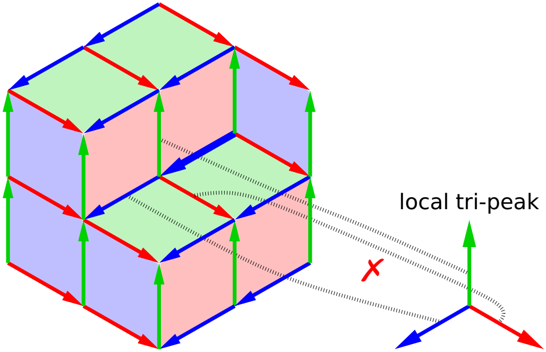
local tri-peak



(3) bricklaying

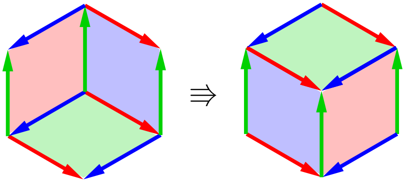


bricklaying rule

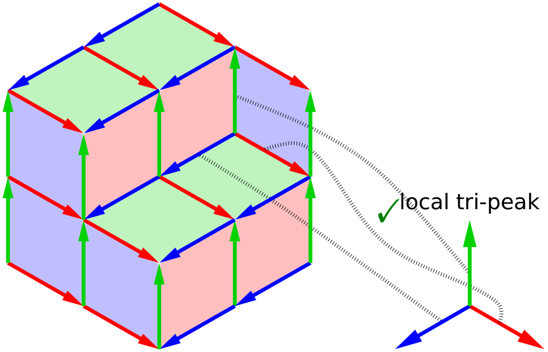


bricklaying step possible?

(3) bricklaying

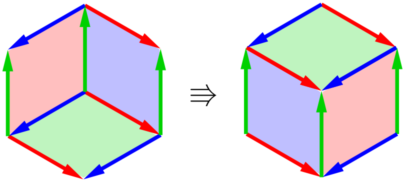


bricklaying rule

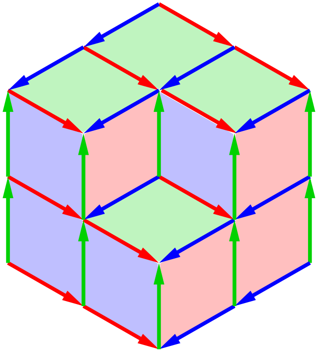


bricklaying step

(3) bricklaying

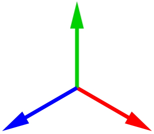


bricklaying rule

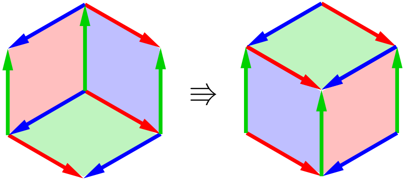


bricklaying step possible?

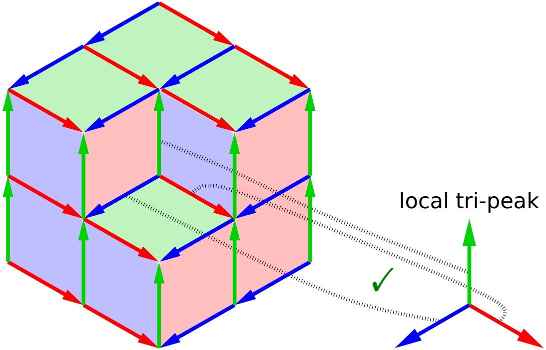
local tri-peak



(3) bricklaying

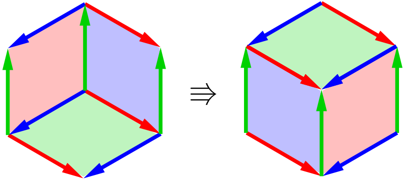


bricklaying rule

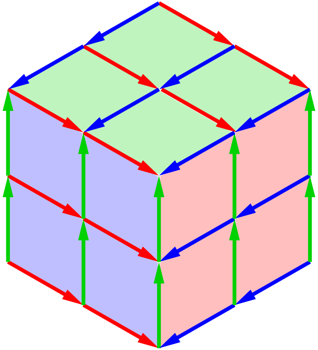


bricklaying step

(3) bricklaying

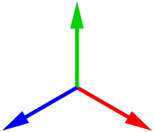


bricklaying rule

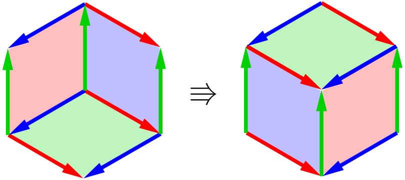


bricklaying step possible?

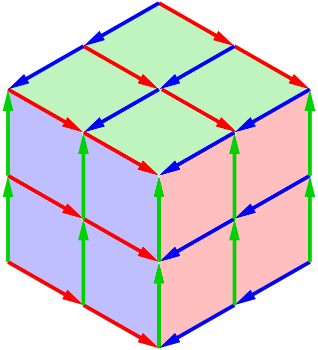
local tri-peak



(3) bricklaying

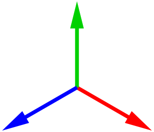


bricklaying rule



big brick (bricklayer)

local tri-peak



(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
  (bed: plane bed-graph; bed-graph: dag obtained by tiling; ☞ 23)

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum per construction preserved by bricklaying $\Rrightarrow$ steps

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
  (trivial; calissons closer to their origin)

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
- bricklaying $\Rrightarrow$ normal form iff big brick
  (out-degree edges $\leq$ 3; if some tri-peak $\implies$ bricklaying step found by following back in-edges; if no tri-peaks $\implies$ big brick; holds for bed-graphs)

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
- bricklaying $\Rrightarrow$ normal form iff big brick
- big brick unique for hexagon; filled boxes $\Rrightarrow$-convertible so same spectrum (4 calissons of each colour)

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
- bricklaying $\Rrightarrow$ normal form iff big brick
- big brick unique for hexagon; filled boxes $\Rrightarrow$-convertible so same spectrum

remark

conversions : (2-dimensional) tiling = beds : (3-dimensional) bricklaying ; 23

(3) bricklaying

- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
- bricklaying $\Rrightarrow$ normal form iff big brick
- big brick unique for hexagon; filled boxes $\Rrightarrow$-convertible so same spectrum

remark

conversions : tiling = beds : bricklaying

bricklaying reduces all fillings to $\Rrightarrow$-normal form, a big brick, unique for hexagon but characterisation of big bricks?

(3) bricklaying

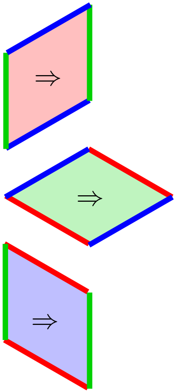
- bricklaying $\Rrightarrow$ is graph rewrite system over beds
- spectrum preserved by bricklaying $\Rrightarrow$ steps
- bricklaying $\Rrightarrow$ terminating
- bricklaying $\Rrightarrow$ normal form iff big brick
- big brick unique for hexagon; filled boxes $\Rrightarrow$-convertible so same spectrum

remark

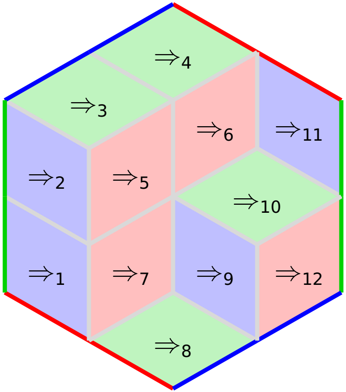
conversions : tiling = beds : bricklaying

bricklaying reduces all fillings to $\Rrightarrow$-normal form, a big brick, unique for hexagon
filling ($\Rrightarrow$) equivalent iff projection ($\Downarrow$) equivalent; big brick least $\Downarrow$-upperbound

(4) local undercutting; from ⇛-filling to ⇓-projection

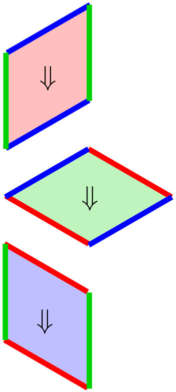


filling rules

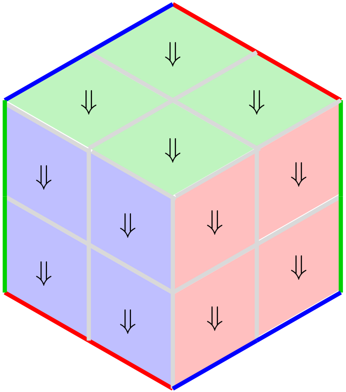


from ⇛-filling

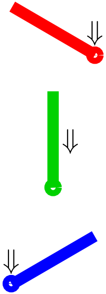
(4) local undercutting; from $\Rightarrow$-filling to $\Downarrow$-projection



zap rules

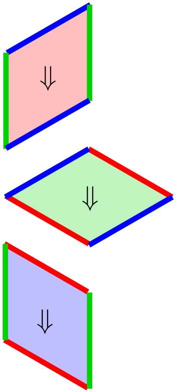


to $\Downarrow$-projection with same spectrum

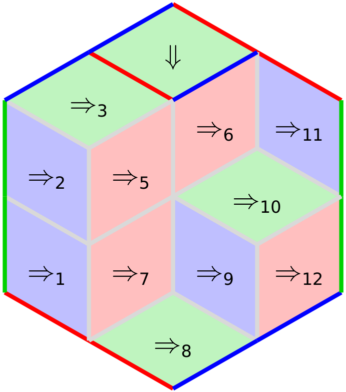


trivial zap rules

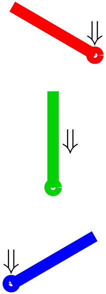
(4) local undercutting



zap rules

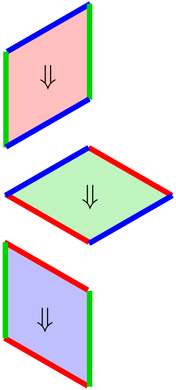


zap step

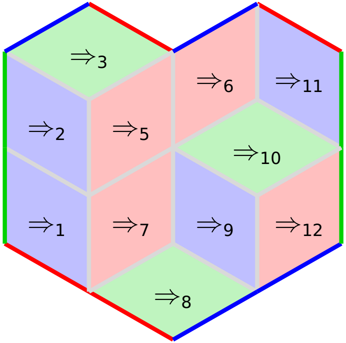


trivial zap rules

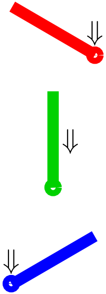
(4) local undercutting



zap rules

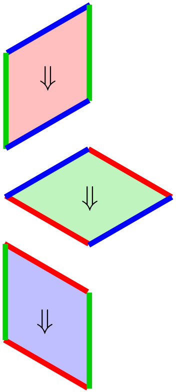


foliage (for cyclic conversion)

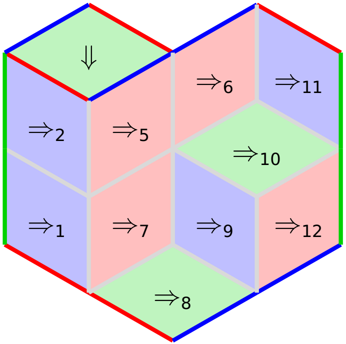


trivial zap rules

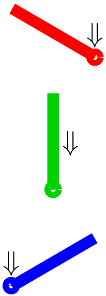
(4) local undercutting



zap rules

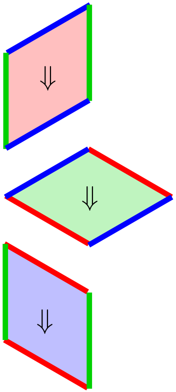


zap step

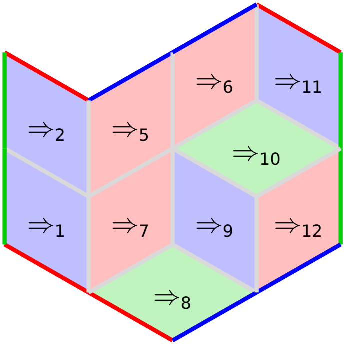


trivial zap rules

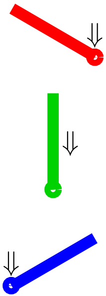
(4) local undercutting



zap rules

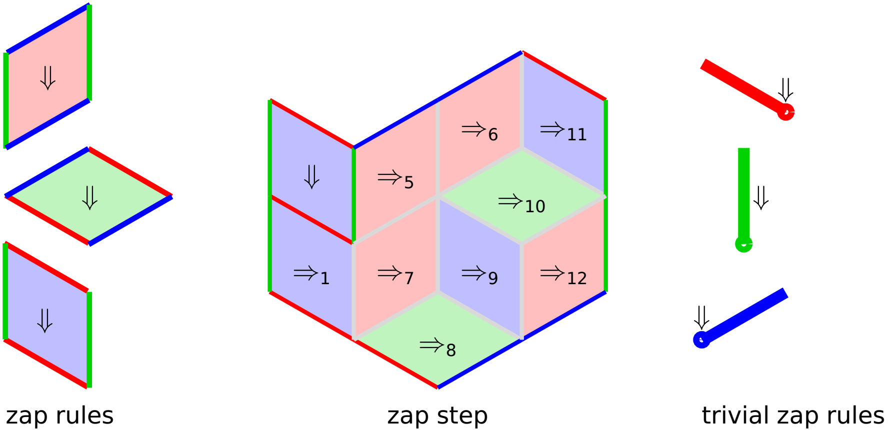


foliage

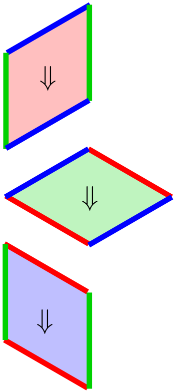


trivial zap rules

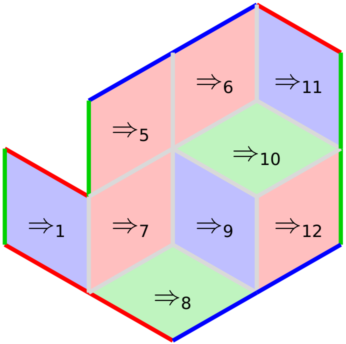
(4) local undercutting



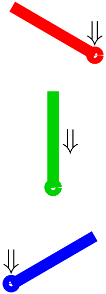
(4) local undercutting



zap rules

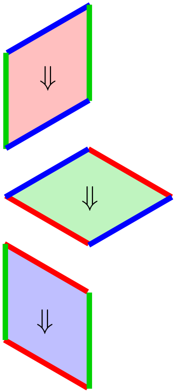


foliage

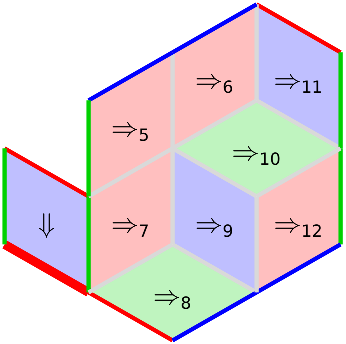


trivial zap rules

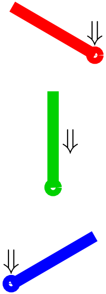
(4) local undercutting



zap rules

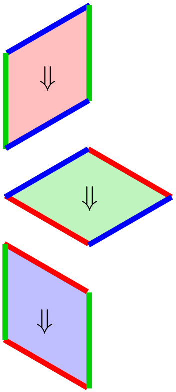


zap step

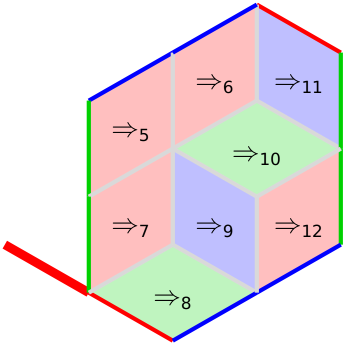


trivial zap rules

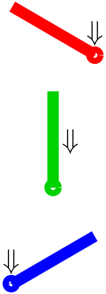
(4) local undercutting



zap rules

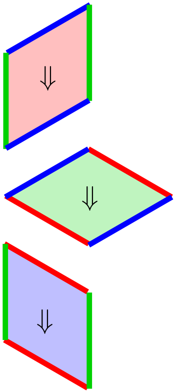


foliage

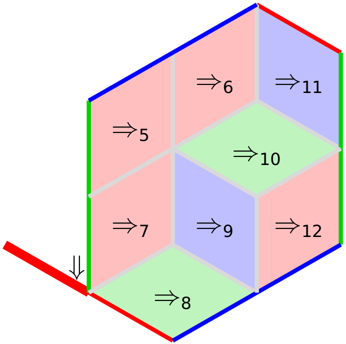


trivial zap rules

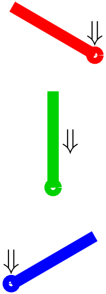
(4) local undercutting



zap rules

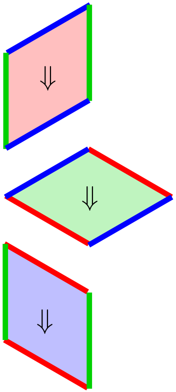


trivial zap step

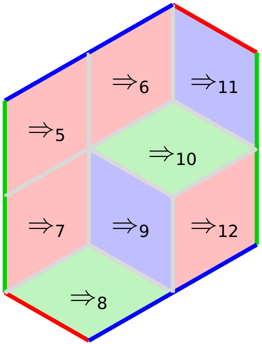


trivial zap rules

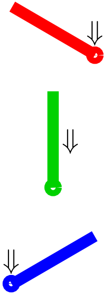
(4) local undercutting



zap rules

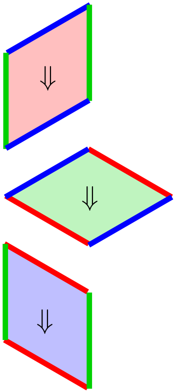


foliage

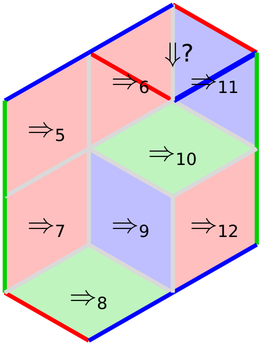


trivial zap rules

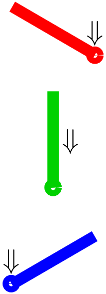
(4) local undercutting



zap rules

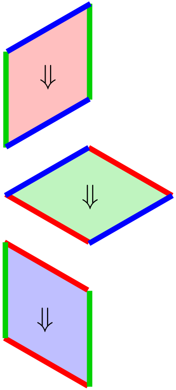


zap step incompatible with ⇒



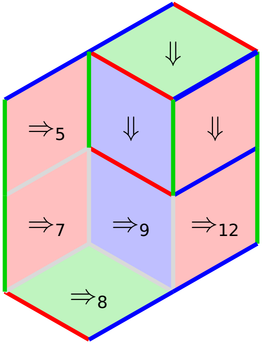
trivial zap rules

(4) local undercutting

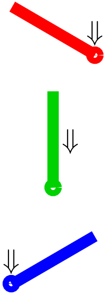


zap rules

preserves spectrum

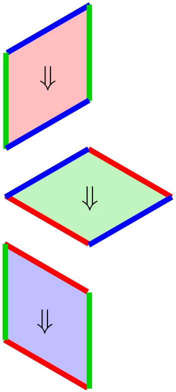


⇓ locally undercuts ⇒

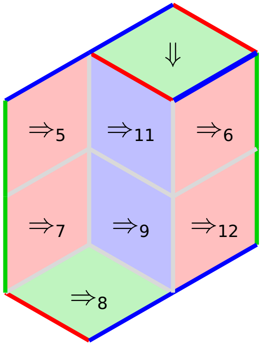


trivial zap rules

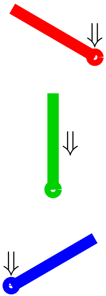
(4) local undercutting



zap rules

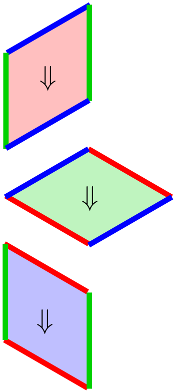


zap step compatible with ⇒

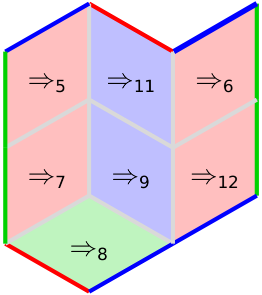


trivial zap rules

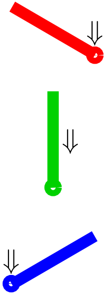
(4) local undercutting



zap rules

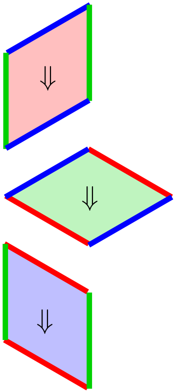


foliage

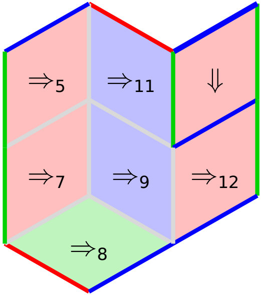


trivial zap rules

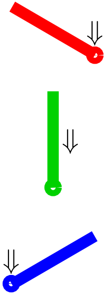
(4) local undercutting



zap rules

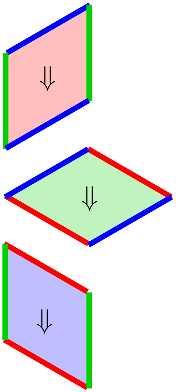


zap step

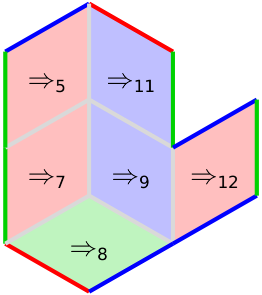


trivial zap rules

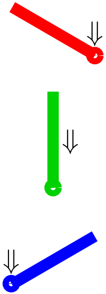
(4) local undercutting



zap rules

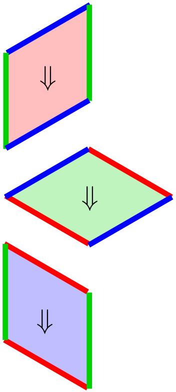


foliage

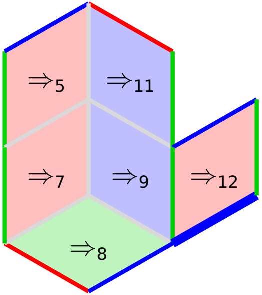


trivial zap rules

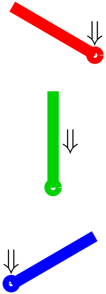
(4) local undercutting



zap rules

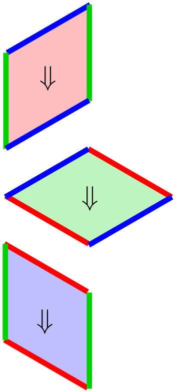


zap step

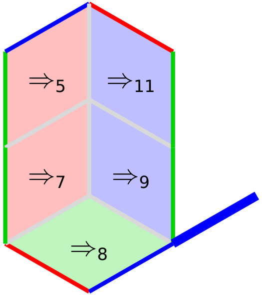


trivial zap rules

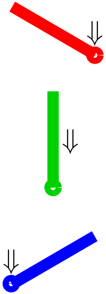
(4) local undercutting



zap rules

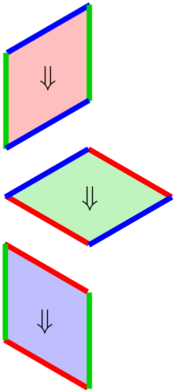


foliage

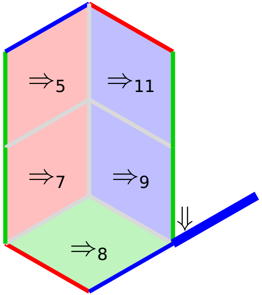


trivial zap rules

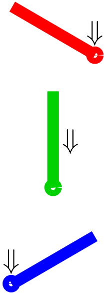
(4) local undercutting



zap rules

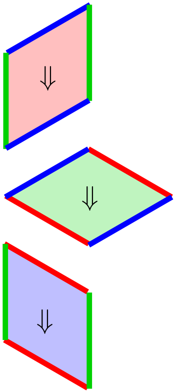


trivial zap step

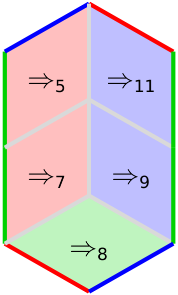


trivial zap rules

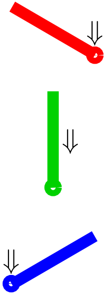
(4) local undercutting



zap rules

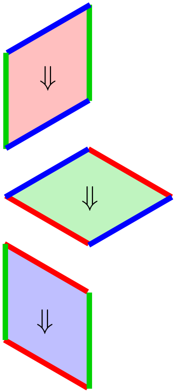


foliage

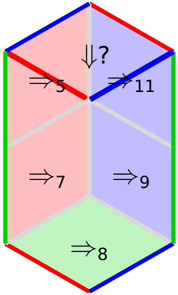


trivial zap rules

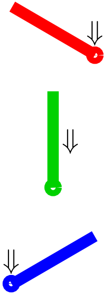
(4) local undercutting



zap rules

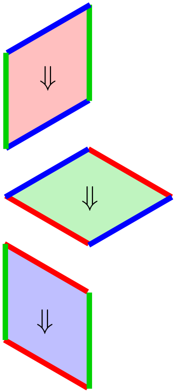


zap step incompatible with ⇒

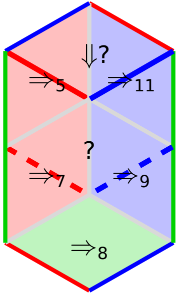


trivial zap rules

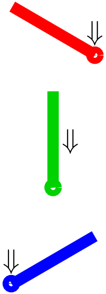
(4) local undercutting



zap rules

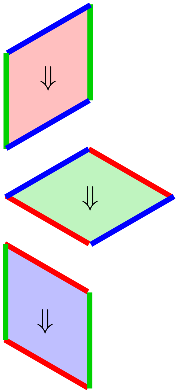


no local undercutting yet

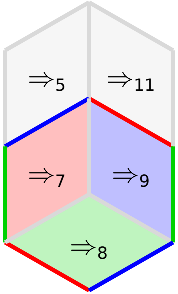


trivial zap rules

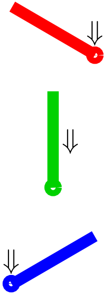
(4) local undercutting



zap rules

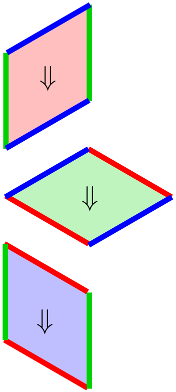


subfoliage

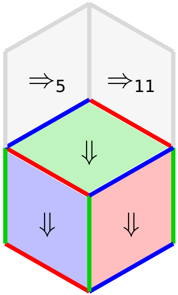


trivial zap rules

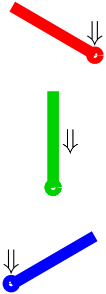
(4) local undercutting



zap rules

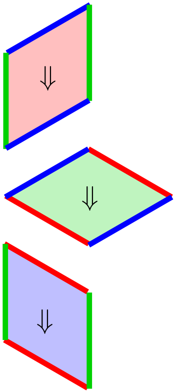


zap steps by IH for subfoliage

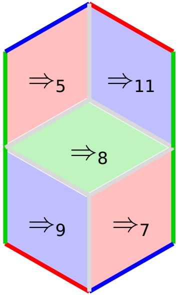


trivial zap rules

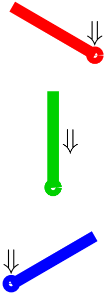
(4) local undercutting



zap rules

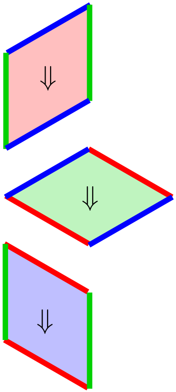


foliage

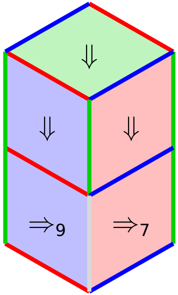


trivial zap rules

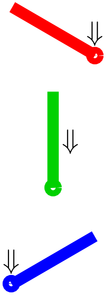
(4) local undercutting



zap rules

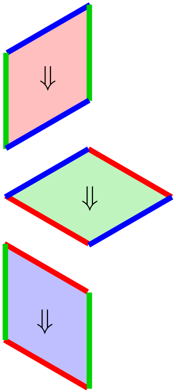


$\Downarrow$ locally undercuts $\Rightarrow$

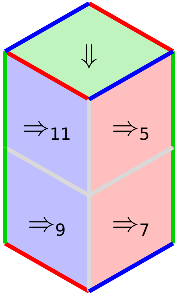


trivial zap rules

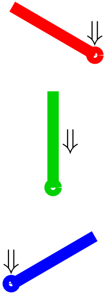
(4) local undercutting



zap rules

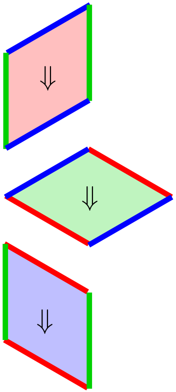


zap step compatible with $\Rightarrow$

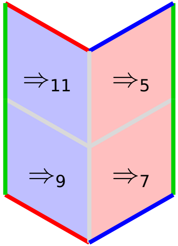


trivial zap rules

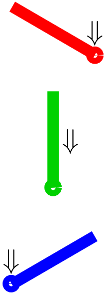
(4) local undercutting



zap rules

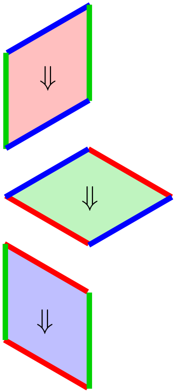


foliage

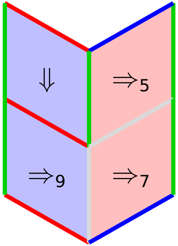


trivial zap rules

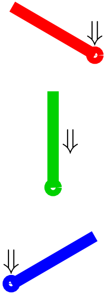
(4) local undercutting



zap rules

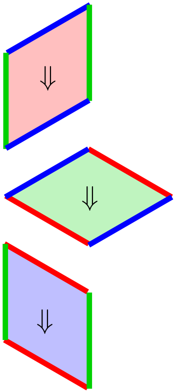


zap step

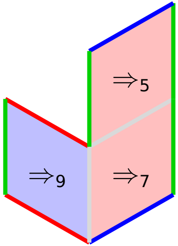


trivial zap rules

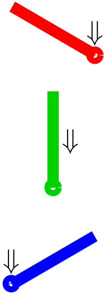
(4) local undercutting



zap rules

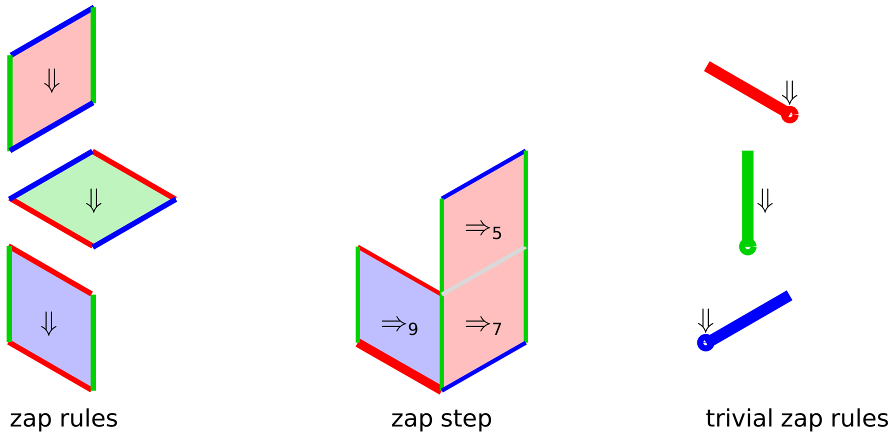


foliage

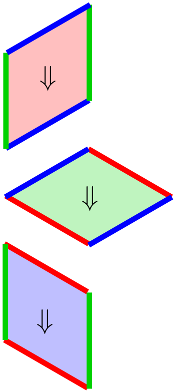


trivial zap rules

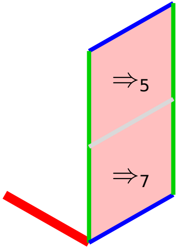
(4) local undercutting



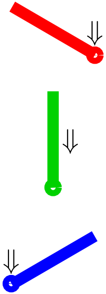
(4) local undercutting



zap rules

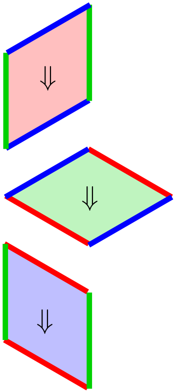


foliage

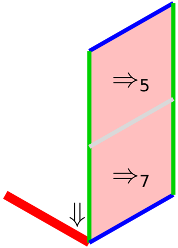


trivial zap rules

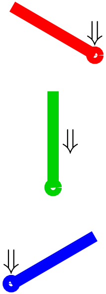
(4) local undercutting



zap rules

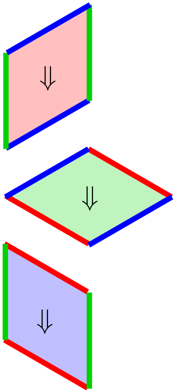


trivial zap step

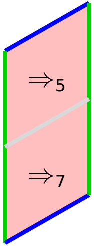


trivial zap rules

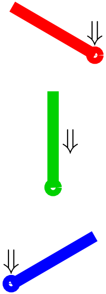
(4) local undercutting



zap rules

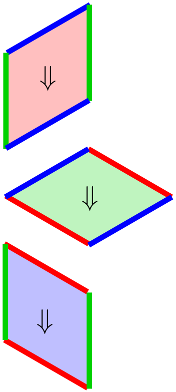


foliage

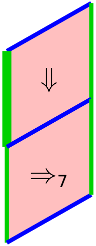


trivial zap rules

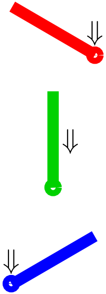
(4) local undercutting



zap rules

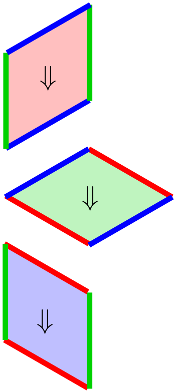


zap step

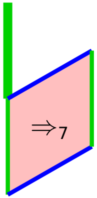


trivial zap rules

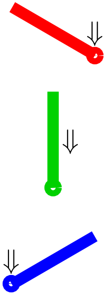
(4) local undercutting



zap rules

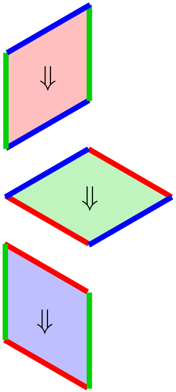


foliage

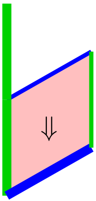


trivial zap rules

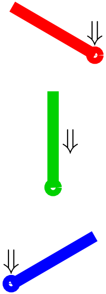
(4) local undercutting



zap rules

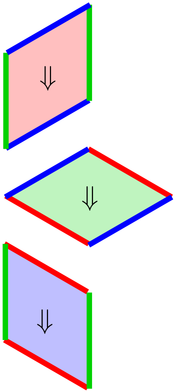


zap step

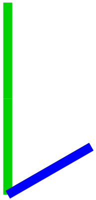


trivial zap rules

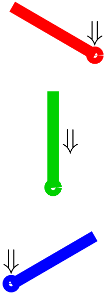
(4) local undercutting



zap rules

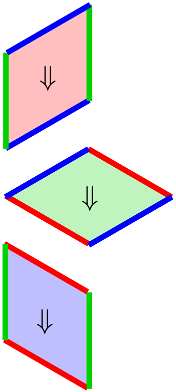


foliage

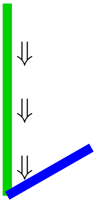


trivial zap rules

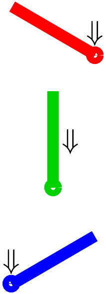
(4) local undercutting



zap rules

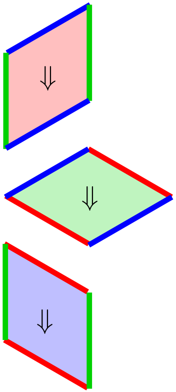


3 trivial zap steps



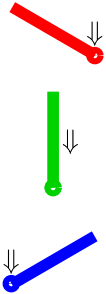
trivial zap rules

(4) local undercutting



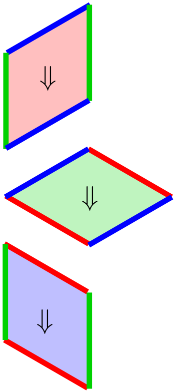
zap rules

empty foliage

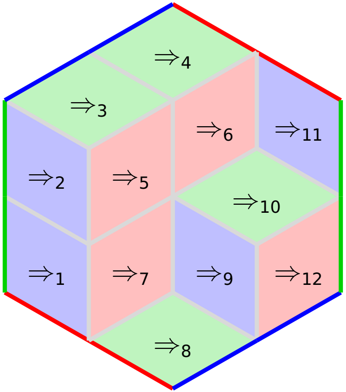


trivial zap rules

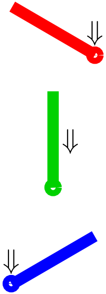
(4) local undercutting; from $\Rightarrow$-filling to $\Downarrow$-projection



zap rules

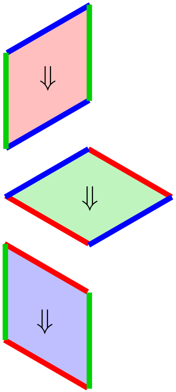


from $\Rightarrow$-filling

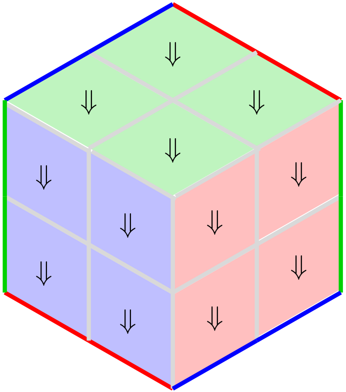


trivial zap rules

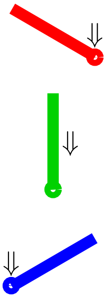
(4) local undercutting; from $\Rightarrow$-filling to $\Downarrow$-projection



zap rules



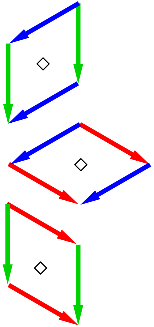
to $\Downarrow$-projection with same spectrum



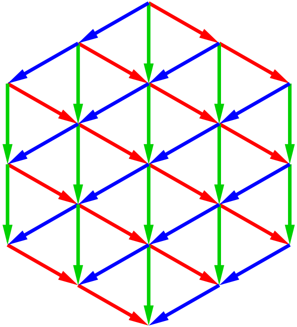
trivial zap rules

(4) local undercutting

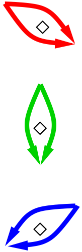
- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\rightarrow$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions



diamonds



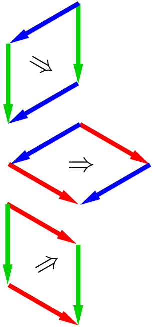
grid rewrite system $\rightarrow$



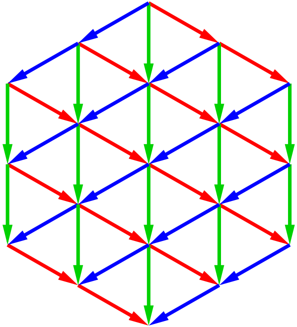
trivial diamonds

(4) local undercutting

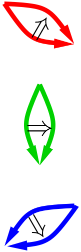
- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\to$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions



filling rules



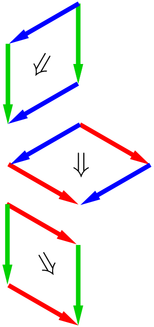
grid rewrite system $\to$



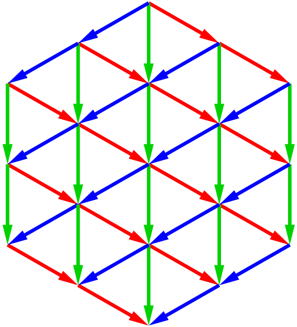
trivial filling rules

(4) local undercutting

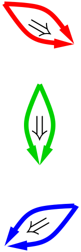
- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\rightarrow$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions



projection rules



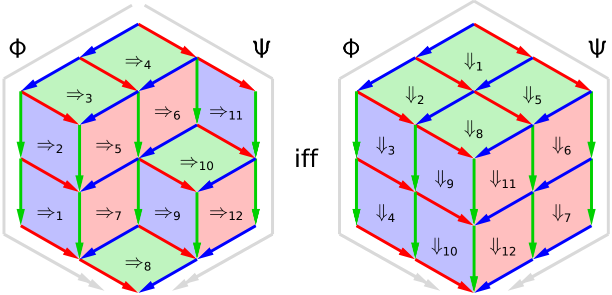
grid rewrite system $\rightarrow$



trivial projection rules

(4) local undercutting; from $\Rightarrow$-filling to $\Downarrow$-projection

- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\to$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rrightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Ddownarrow \varepsilon$ for reductions Φ, Ψ (Lévy 78, ⩔ & Klop & de Vrijer 98)

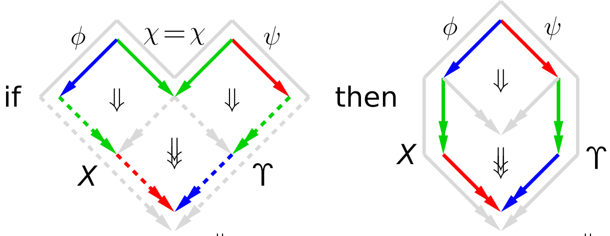


filling $\Phi \Rrightarrow \Psi$ projection $\Phi^{-1} \cdot \Psi \Ddownarrow \varepsilon$

(4) local undercutting; from $\Rightarrow$-filling to $\Downarrow$-projection

- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\rightarrow$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rrightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Downarrow\!\!\Downarrow \varepsilon$ for reductions Φ, Ψ

  if $\rightarrow$ terminating and projection $\Downarrow$ locally undercutting (LUC; more later)



$\phi^{-1} \cdot \chi \cdot \chi^{-1} \cdot \psi \Downarrow\!\!\Downarrow X \cdot \Upsilon^{-1}$ $\qquad$ $(\phi \cdot X)^{-1} \cdot \psi \cdot \Upsilon \Downarrow\!\!\Downarrow \varepsilon$

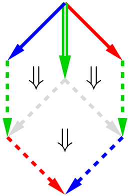
(4) local undercutting

- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\rightarrow$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rrightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Downarrow\!\!\Downarrow \varepsilon$ for reductions Φ, Ψ
  if $\rightarrow$ terminating and projection $\Downarrow$ locally undercutting (LUC; more later)
- grid rewrite system $\rightarrow$ is terminating (trivial; $\rightarrow$ is a dag)

(4) local undercutting

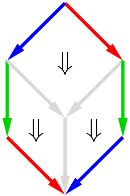
- calissons as diamonds ${}^{\phi}_{\chi}\!\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\!\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\to$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rightarrow\!\!\!\Rightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Downarrow\!\!\Downarrow \varepsilon$ for reductions Φ, Ψ
  if $\to$ terminating and projection $\Downarrow$ locally undercutting (LUC; more later)
- grid rewrite system $\to$ is terminating
- projection $\Downarrow$ is locally undercutting

if



local cut

then



undercut

(4) local undercutting

- calissons as diamonds ${}^{\phi}_{\chi}\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\to$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rightarrow\!\!\!\!\Rightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Downarrow\!\!\!\!\Downarrow \varepsilon$ for reductions Φ, Ψ
  if $\to$ terminating and projection $\Downarrow$ locally undercutting (LUC; more later)
- grid rewrite system $\to$ is terminating
- projection $\Downarrow$ is locally undercutting
- undercutting preserves spectrum so spectrum of filling and projection same (spectrum of projection is unique by random descent; 07)

(4) local undercutting

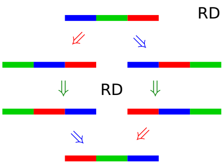
- calissons as diamonds ${}^{\phi}_{\chi}\!\diamond^{\psi}_{\upsilon}$ and ${}^{\phi}_{\varepsilon}\!\diamond^{\phi}_{\varepsilon}$ of grid rewrite system $\to$ for hexagon filling $\phi \cdot \chi \Rightarrow \psi \cdot \upsilon$ on reductions, projection $\phi^{-1} \cdot \psi \Downarrow \chi \cdot \upsilon^{-1}$ on conversions
- $\Phi \Rrightarrow \Psi$ iff $\Phi^{-1} \cdot \Psi \Downarrow\!\!\Downarrow \varepsilon$ for reductions Φ, Ψ
  if $\to$ terminating and projection $\Downarrow$ locally undercutting (LUC; more later)
- grid rewrite system $\to$ is terminating
- projection $\Downarrow$ is locally undercutting
- undercutting preserves spectrum so spectrum of filling and projection same

remark

zapping, contracting conversion cycles to a loop, goes back to Newman 42
it is a basic tool for e.g. Finite Derivation Types (Squier 87), Garside theory (Dehornoy et al. 15), homotopy type theory (Kraus & von Raumer 23), and polygraphs (the ‘polybook’, Ara et al. 25)

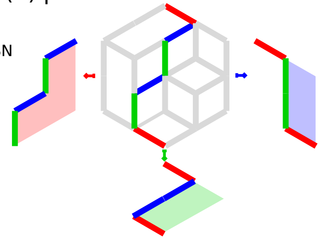
(1) random descent

RD & WN $\Longleftrightarrow$ WCR & SN



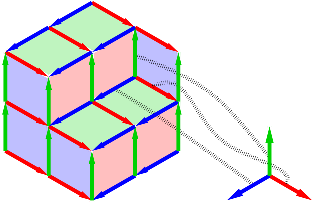
quantitative commutation

(2) proof order



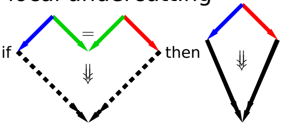
typed involutive monoids for conversions

(3) bricklaying



big brick as unique normal form of beds

(4) local undercutting



common multiple $\Longrightarrow$ least common multiple

upperbound $\Longrightarrow$ least upperbound / emph

confluent $\Longrightarrow$ orthogonal

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid if

- $\langle M, \perp, + \rangle$ a monoid;

idea: measure reduction (derivation) as sum of the weights of its (non-$\perp$) steps

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid if

- $\langle M, \perp, + \rangle$ a monoid;
- $\leq$ well-founded order with $\perp$ least;

idea: measure reduction (derivation) as sum of the weights of its (non-$\perp$) steps

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid if

- $\langle M, \perp, + \rangle$ a monoid;
- $\leq$ well-founded order with $\perp$ least;
- $+$ is $\leq$-monotonic in both arguments; strictly in 2^{nd}.

idea: measure reduction (derivation) as sum of the weights of its (non-$\perp$) steps

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid if

- $\langle M, \perp, + \rangle$ a monoid;
- $\leq$ well-founded order with $\perp$ least;
- $+$ is $\leq$-monotonic in both arguments; strictly in 2$^{\text{nd}}$.

main example: ordinals with zero, addition, less–than–or–equal

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid

- measure on $\rightarrow$ maps steps to $M - \{\perp\}$;

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid

- measure on $\rightarrow$ maps steps to $M - \{\perp\}$;
- measure of finite reduction is sum ($+$; tail to head) of steps (starting with $\perp$);

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid

- measure on $\to$ maps steps to $M - \{\perp\}$;
- measure of finite reduction is sum of steps ;
- measure of infinite reduction is $\top$ (fresh top greater than all $m \in M$).

Characterising random descent

Definition

$\langle M, \perp, +, \leq \rangle$ derivation monoid

- measure on $\to$ maps steps to $M - \{\perp\}$;
- measure of finite reduction is sum of steps ;
- measure of infinite reduction is $\top$ (fresh top greater than all $m \in M$).

Theorem (RD)

local peaks completable to same weight $\iff$ *peak random descent (PR):*

$\mathrm{NF} \ni a \; {}_{n}^{*}\!\!\leftarrow \cdot \to_{\mu}^{\circ} b \implies \exists k. a \; {}_{k}^{*}\!\!\leftarrow b \; \& \; k + \mu = n$

(peak of reductions to nf $\implies$ reductions same weight)

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

λI is the original non-erasing λ-calculus where $x \in FV(M)$ for all $\lambda x.M$

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

intuition: because β is non-erasing in λI it makes terms larger
but does not quite work, e.g. $(\lambda x.x\,x)\,(\lambda x.x\,x)$ β-reduces to itself

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

Proof.

- introduce (unary) edge symbol e; measure terms by number of e's

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

Proof.

- introduce (unary) edge symbol e; measure terms by number of e's
- lift β-rule: $e^n(\lambda x.M)\,N \to e^{n+1}(M[x{:=}N])$; weigh step by difference

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

Proof.

- introduce (unary) edge symbol e; measure terms by number of e's
- lift β-rule: $e^n(\lambda x.M)\,N \to e^{n+1}(M[x{:=}N])$; weigh step by difference
- local peaks completable into same weight by orthogonality of lifted system

Applying random descent

Theorem (Church 41)

if M is a λI-term, then M is normalising (WN) iff M is terminating (SN)

Proof.

- introduce (unary) edge symbol e; measure terms by number of e's
- lift β-rule: $e^n(\lambda x.M)\,N \to e^{n+1}(M[x{:=}N])$; weigh step by difference
- local peaks completable into same weight by orthogonality of lifted system
- conclude SN of M from WN of M by RD theorem
  (weight of no reduction exceeds that of one to normal form, so SN) $\square$

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

for every object there exists (SN) a unique (CR) normal form; $\to$ models function

Completeness of random descent for completeness

Definition

rewrite system $\rightarrow$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

confluent & terminating (SN) $\iff$ *random descent & normalising (WN)*

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

if-direction above; idea for only only–if-direction:
assign weights by topological sorting starting from normal forms $\square$

Completeness of random descent for completeness

Definition

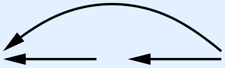
rewrite system $\rightarrow$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \rightarrow b\}$; step $a \rightarrow b$ by dif a and b



$\square$

Completeness of random descent for completeness

Definition

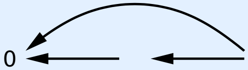
rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \to b\}$; step $a \to b$ by dif a and b



$\square$

Completeness of random descent for completeness

Definition

rewrite system $\rightarrow$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \rightarrow b\}$; step $a \rightarrow b$ by dif a and b

$$\begin{tikzcd} 0 & 1 \arrow[l] & {} \arrow[l] \arrow[ll, bend right] \end{tikzcd}$$

$\square$

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \to b\}$; step $a \to b$ by dif a and b

$$\begin{tikzcd} 0 & 1 \arrow[l, "1"'] & {} \arrow[l] \arrow[ll, bend right] \end{tikzcd}$$

$\square$

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \to b\}$; step $a \to b$ by dif a and b

$$\begin{tikzcd} 0 & 1 \arrow[l, "1"'] & 2 \arrow[l] \arrow[ll, bend right] \end{tikzcd}$$

$\square$

Completeness of random descent for completeness

Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

measure a by supremum$\{(\text{measure of } b) + 1 \mid a \to b\}$; step $a \to b$ by dif a and b

$$\begin{tikzcd}
0 & 1 \arrow[l, "1"'] & 2 \arrow[l, "1"'] \arrow[ll, bend right, "2"']
\end{tikzcd}$$

$\square$

Completeness of random descent for completeness

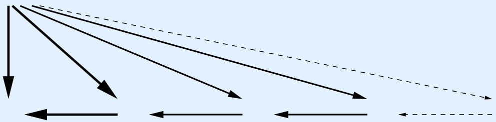
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

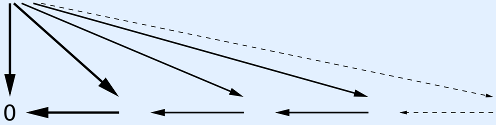
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

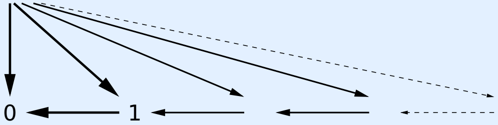
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

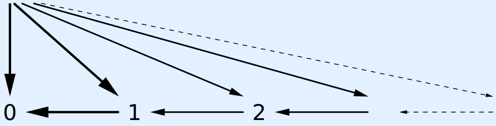
Definition

rewrite system $\rightarrow$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

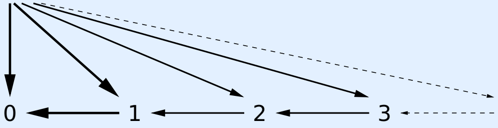
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

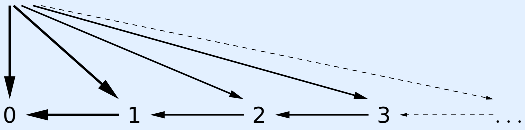
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Completeness of random descent for completeness

Definition

rewrite system $\rightarrow$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.

$$\begin{tikzcd}
\omega \arrow[d] \arrow[dr] \arrow[drr] \arrow[drrr] \arrow[drrrr, dashed] & & & & \\
0 & 1 \arrow[l] & 2 \arrow[l] & 3 \arrow[l] & \cdots \arrow[l, dashed]
\end{tikzcd}$$

$\square$

Completeness of random descent for completeness

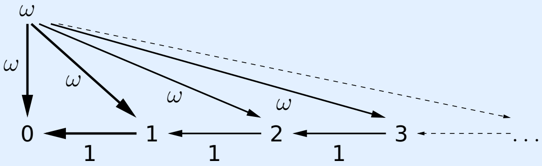
Definition

rewrite system $\to$ is complete if it is confluent (CR) and terminating (SN)

Theorem (as mentioned in first half)

complete $\iff$ *random descent & normalising (WN)*

Proof by example.



$\square$

Applying completeness of random descent

Definition (Nederpelt & Klop)

$\to$ is increasing (INC) if map from objects to $\mathbb{N}$ that increases by rewriting

Applying completeness of random descent

Definition (Nederpelt & Klop)

$\to$ is increasing (INC) if map from objects to $\mathbb{N}$ that increases by rewriting

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ *complete (CR & SN)*

Proof.

WCR & INC $\implies$ random descent (RD) by weighing with difference $\square$

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ complete (CR & SN)

Theorem

simply typed $\lambda\beta$-calculus is complete

Proof.

WCR by orthogonality, WN by strategy (Turing), but INC?? (erasing!)

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\Longrightarrow$ *complete (CR & SN)*

Theorem

simply typed $\lambda\beta$*-calculus is complete*

Proof.

- memorise (Nederpelt,Klop,Khasidashvili,de Groote,Wells,...)

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ *complete (CR & SN)*

Theorem

simply typed $\lambda\beta$*-calculus is complete*

Proof.

- memorise (Nederpelt,Klop,Khasidashvili,de Groote,Wells,. . .)
- lift β as $\langle(\lambda x.M),\vec{K}\rangle N \to \langle M[x{:=}N], N\vec{K}\rangle$ where $\langle L,\vec{K}\rangle$ is L with memory $\vec{K}$

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ complete (CR & SN)

Theorem

simply typed $\lambda\beta$-calculus is complete

Proof.

- memorise (Nederpelt,Klop,Khasidashvili,de Groote,Wells,...)
- lift β as $\langle(\lambda x.M),\vec{K}\rangle N \to \langle M[x{:=}N], N\vec{K}\rangle$ where $\langle L,\vec{K}\rangle$ is L with memory $\vec{K}$
- lifting preserves WCR, WN, typing; creates INC

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ complete (CR & SN)

Theorem

simply typed $\lambda\beta$-calculus is complete

Proof.

- memorise (Nederpelt,Klop,Khasidashvili,de Groote,Wells,...)
- lift β as $\langle(\lambda x.M),\vec{K}\rangle N \to \langle M[x{:=}N], N\vec{K}\rangle$ where $\langle L,\vec{K}\rangle$ is L with memory $\vec{K}$
- lifting preserves WCR, WN, typing; creates INC
- conclude to completeness (CR & SN) by corollary $\square$

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\implies$ *complete (CR & SN)*

Theorem

simply typed $\lambda\beta$-calculus is complete

Barendregt–Geuvers–Klop conjecture

proofs of WN lift to proofs of SN for typed λ-calculi (PTSs)

Applying completeness of random descent

Corollary (result by Nederpelt & Klop)

normalisation WN & local confluence (WCR) & INC $\Longrightarrow$ complete (CR & SN)

Theorem

simply typed $\lambda\beta$-calculus is complete

Barendregt–Geuvers–Klop conjecture

proofs of WN lift to proofs of SN for typed λ-calculi (PTSs)
my take: λ-calculi confluent, so the same as proving RD; which measure??

Finitely branching systems

Observation

for finitely branching (FB) systems, measures in completeness proof in $\mathbb{N}$

Finitely branching systems

Observation

for finitely branching systems, measures in completeness proof in $\mathbb{N}$

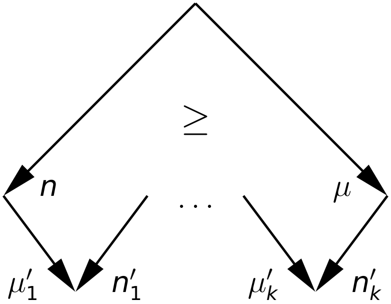
$+$ commutative, cancellative; then RD $\iff$ locally Dyck (Toyama, 2016)

Finitely branching systems

Observation

for finitely branching systems, measures in completeness proof in $\mathbb{N}$

locally Dyck if



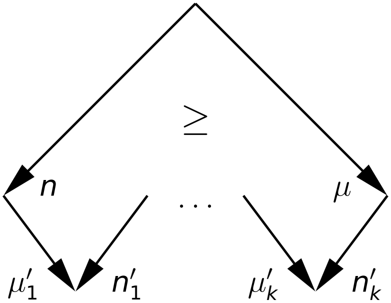
and forward weights $>$ backward weights: $\forall\, i. n + \sum \mu'_i > \sum n'_i$

Finitely branching systems

Corollary

(WN systems) complete iff locally Dyck for some measure

locally Dyck if

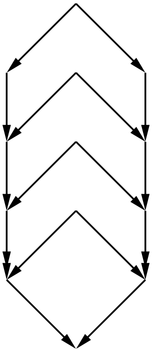


and forward weights $>$ backward weights: $\forall i. n + \sum \mu'_i > \sum n'_i$

Example: deep valleys with shallow conversions

Example (⩔ 2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

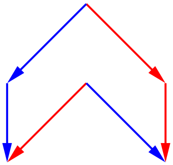


Example: deep valleys with shallow conversions

Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete:



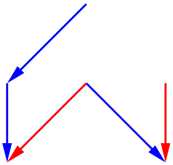
forward $3 \geq 3$ backward

Example: deep valleys with shallow conversions

Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete:



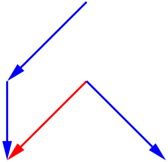
forward 3 $>$ 2 backward

Example: deep valleys with shallow conversions

Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete:



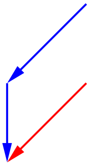
forward 3 $>$ 1 backward

Example: deep valleys with shallow conversions

Example (2008)

$\rightarrow$ with $b_i \leftarrow a_i \rightarrow c_i$, $b_i \rightarrow b_{i+1}$, and $c_i \rightarrow c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\rightarrow$ locally Dyck for length measure, hence uniformly complete:



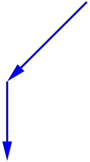
forward 2 $>$ 1 backward

Example: deep valleys with shallow conversions

Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete:



forward $2 > 0$ backward

Example: deep valleys with shallow conversions

Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete:

forward $1 > 0$ backward

Example: deep valleys with shallow conversions

Example (♡ 2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \le i \le n$, with $b_{n+1} = c_{n+1}$

- $\to$ locally Dyck for length measure, hence uniformly complete
- $\to$ is trivially WN

Example: deep valleys with shallow conversions

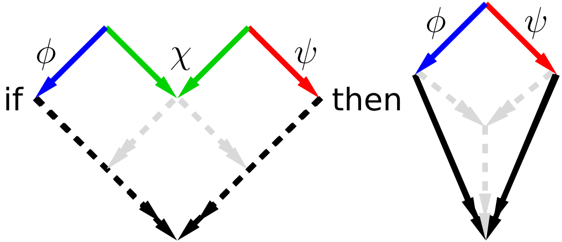
Example (2008)

$\to$ with $b_i \leftarrow a_i \to c_i$, $b_i \to b_{i+1}$, and $c_i \to c_{i+1}$, for $1 \leq i \leq n$, with $b_{n+1} = c_{n+1}$

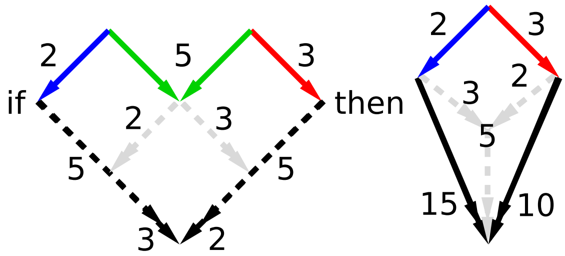
- $\to$ locally Dyck for length measure, hence uniformly complete
- $\to$ is trivially WN

hence $\to$ is complete

Local undercutting / semi-lattice



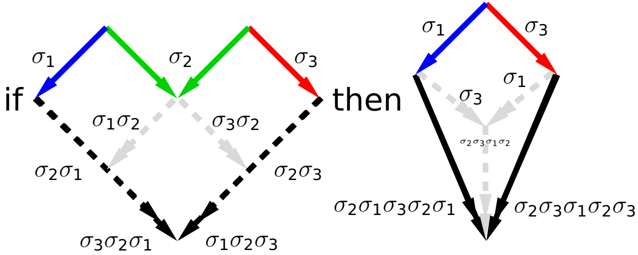
LSL for least upperbounds



Example (positive natural numbers with multiplication)

30 is an upperbound of $\mathrm{lcm}(2,5)$ and $\mathrm{lcm}(5,3)$
and is so too of $\mathrm{lcm}(2,3)$ obtained by cutting 5
($\mathrm{lcm}(2,3)$ undercuts the upperbound 30 of $\mathrm{lcm}(2,5)$ and $\mathrm{lcm}(5,3)$)

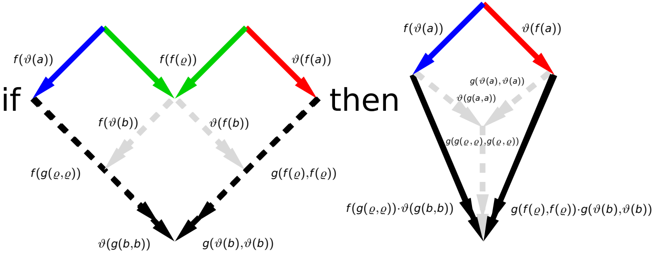
LSL for least common multiples



Example (positive braids; Dehornoy et al. 15, Example II.4.20)

$\sigma_1\sigma_2\sigma_1\sigma_3\sigma_2\sigma_1$ is a common multiple of $\mathrm{lcm}(\sigma_1, \sigma_2)$ and $\mathrm{lcm}(\sigma_2, \sigma_3)$
and is so too of $\mathrm{lcm}(\sigma_1, \sigma_3)$ obtained by cutting σ_2
(with $\sigma_1\sigma_2\sigma_1 = \sigma_2\sigma_1\sigma_2$, $\sigma_1\sigma_3 = \sigma_3\sigma_1$, $\sigma_3\sigma_2\sigma_3 = \sigma_2\sigma_3\sigma_2$ on Artin generators σ_i)

LSL for orthogonality



Example (orthogonal TRSs; Terese 03, Figure 8.53)

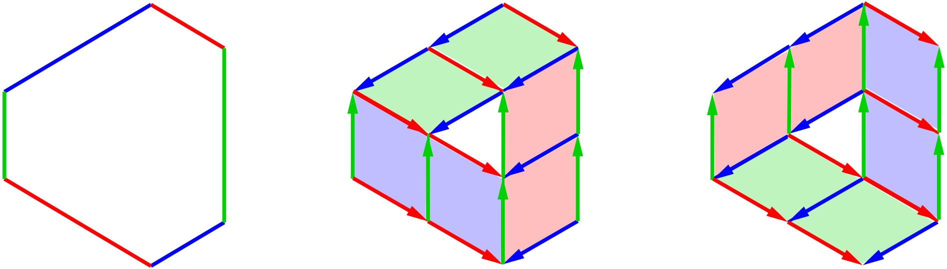
$g(g(b,b),g(b,b))$ is a common reduct of $f(\vartheta(a))^{-1}\cdot f(f(\varrho))$ and $f(f(\varrho))^{-1}\cdot\vartheta(f(a))$ is so too of $f(\vartheta(a))^{-1}\cdot\vartheta(f(a))$ obtained by cutting $f(f(\varrho))$ (for OTRS rules $\varrho : a \to b$ and $\vartheta : f(x) \to g(x)$)

Conclusions

- modern confluence techniques powerful; 4 solve problem of the calissons (for all zonogonal hexagons; non-convex boxes? Dijkstra 89)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons (for all zonogonal hexagons; non-convex boxes? Dijkstra 89)



Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{X}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{X}$) $\Longrightarrow$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids; extends Dehornoy et al. 15 (Projection Theorem: permutation iff projection equivalence (Terese 03) entails cube-property; Lévy 78)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}_{\chi}^{\phi}\diamond_{\Upsilon}^{\psi}$ iff ${}_{\Upsilon}^{\psi}\diamond_{\chi}^{\phi}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)? (coinduction instead of induction?)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{X}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{X}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)?
- quantitative tiling methods (combinatorics) for quantitative rewriting?

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{\chi}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{\chi}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)?
- quantitative tiling methods for quantitative rewriting?
- contrapositive of LSL for non-confluence? Dehornoy et al. 15; Klop 24

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{X}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{X}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\rightarrow$ for filling iff projection (given LSL)?
- quantitative tiling methods for quantitative rewriting?
- contrapositive of LSL for non-confluence?

Thanks to

Jan Willem Klop for suggesting to model calissons by rewriting as in (1),(2)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{\chi}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{\chi}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)?
- quantitative tiling methods for quantitative rewriting?
- contrapositive of LSL for non-confluence?

Thanks to

Jan Willem Klop for suggesting to model the problem of the calissons by rewriting
Nicolai Kraus, Yves Guiraud for discussion on Newman's II-Lemma (see paper)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{X}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{X}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)?
- quantitative tiling methods for quantitative rewriting?
- contrapositive of LSL for non-confluence?

Thanks to

Jan Willem Klop for suggesting to model calissons by rewriting
Nicolai Kraus, Yves Guiraud for discussion on Newman's II-Lemma
Nils for example generator and filler (app needs WebGL)

Conclusions

- modern confluence techniques powerful; solve problem of the calissons
- local semi-lattice (LSL = LUC with ${}^{\phi}_{X}\diamond^{\psi}_{\Upsilon}$ iff ${}^{\psi}_{\Upsilon}\diamond^{\phi}_{X}$) $\implies$ filling iff projection for term rewriting, $\lambda\beta$ and positive braids
- productivity instead of termination of $\to$ for filling iff projection (given LSL)?
- quantitative tiling methods for quantitative rewriting?
- contrapositive of LSL for non-confluence?

Thanks to

Jan Willem Klop for suggesting to model calissons by rewriting
Nicolai Kraus, Yves Guiraud for discussion on Newman's II-Lemma
Nils for example generator and filler
you for your interest