# Decreasing proof orders
## Interpreting conversions in involutive monoids


Vincent van Oostrom

Universiteit Utrecht


IWC, Nagoya, May 29, 2012

Decreasing tiles

Involutive proofs

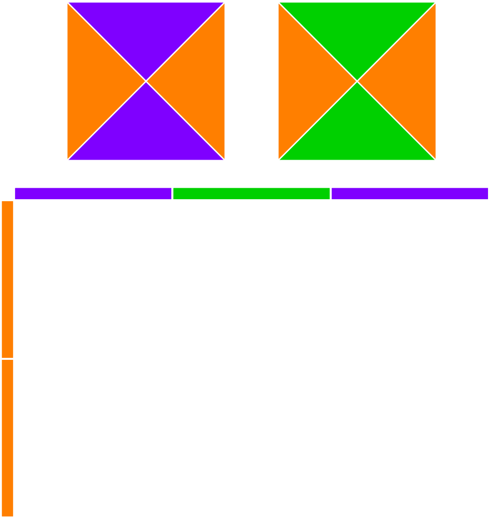
French strings

Applications

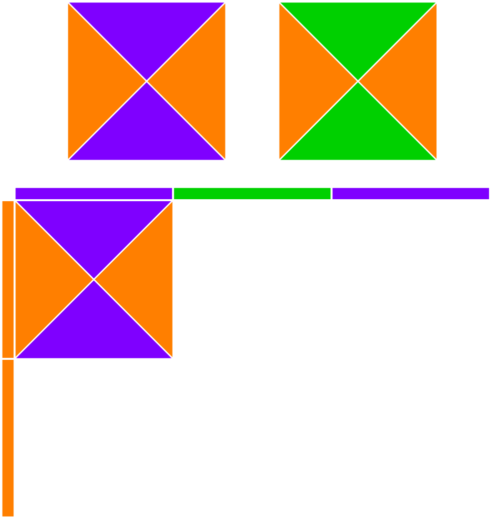
# Alhambra

# Tiling puzzles (1964–73)

# Tiling puzzles (1964–73)



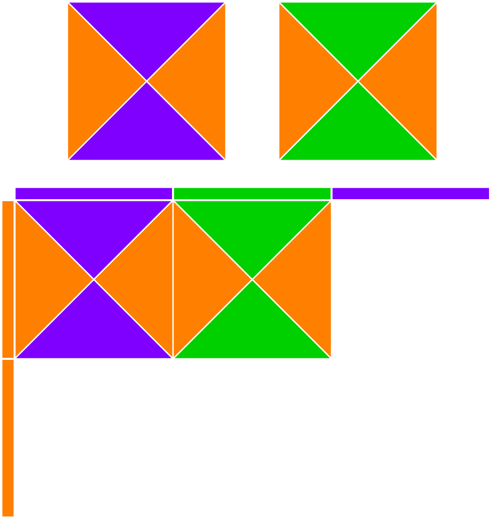
# Tiling puzzles (1964–73)



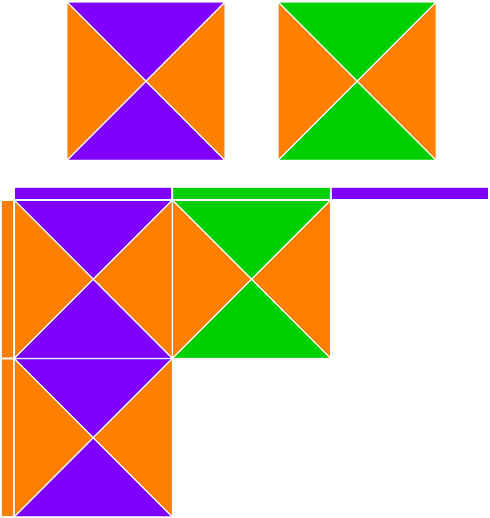


Universiteit Utrecht

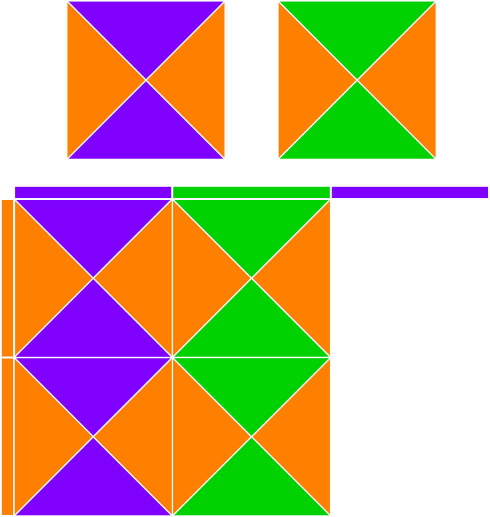
# Tiling puzzles (1964–73)



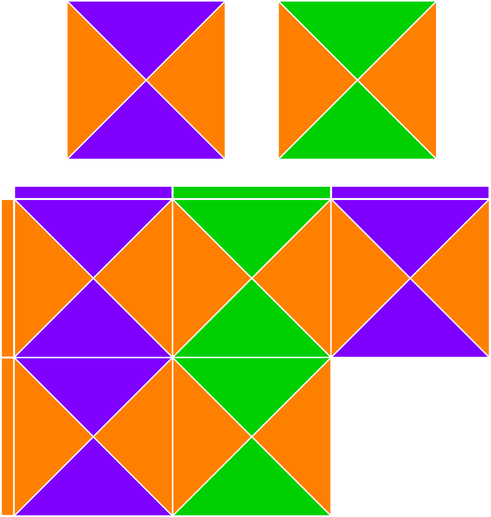
# Tiling puzzles (1964–73)



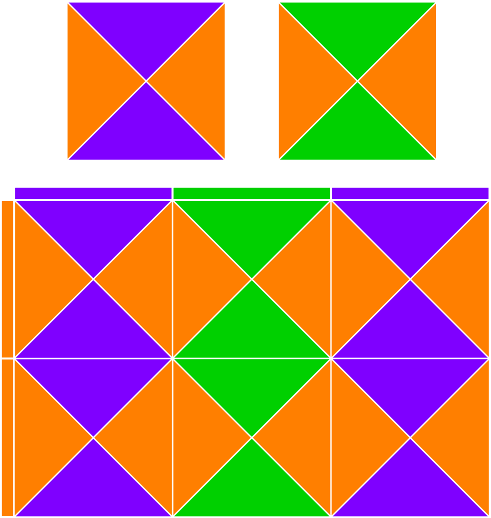
# Tiling puzzles (1964–73)



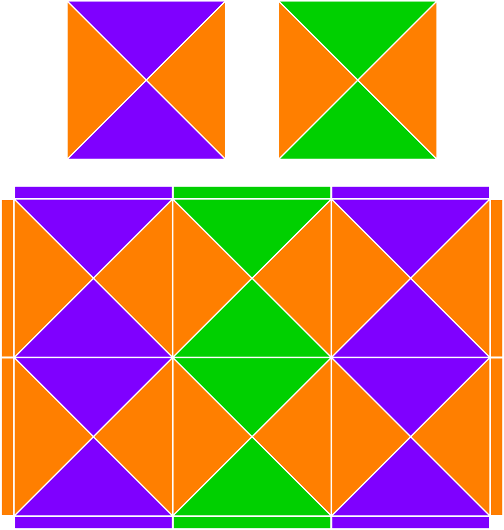
# Tiling puzzles (1964–73)



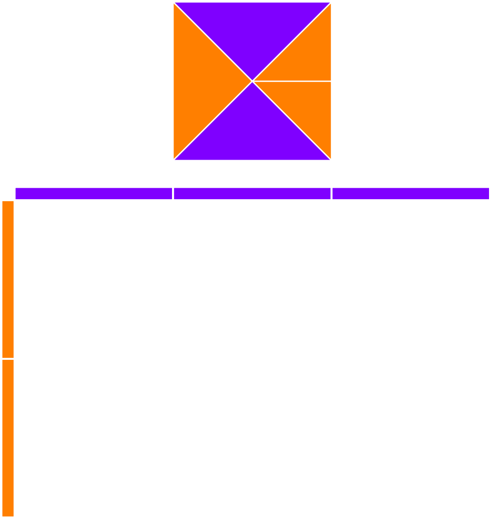
# Tiling puzzles (1964–73)



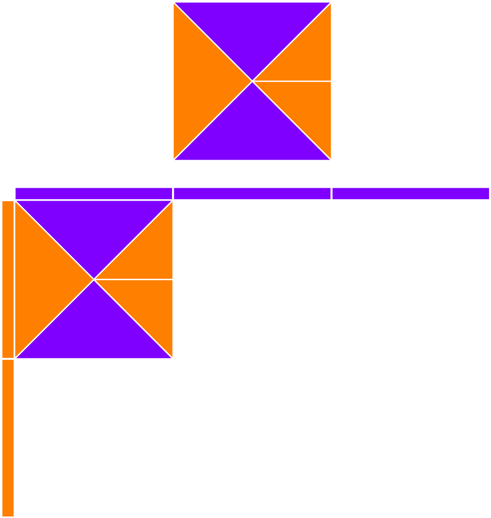
# Tiling puzzles (1964–73)



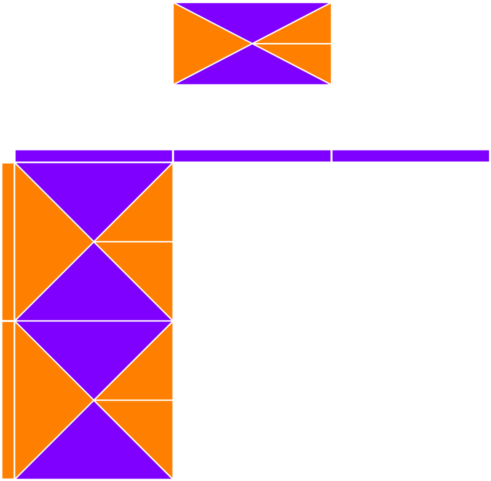
# Scalable tile puzzling (1964–78)



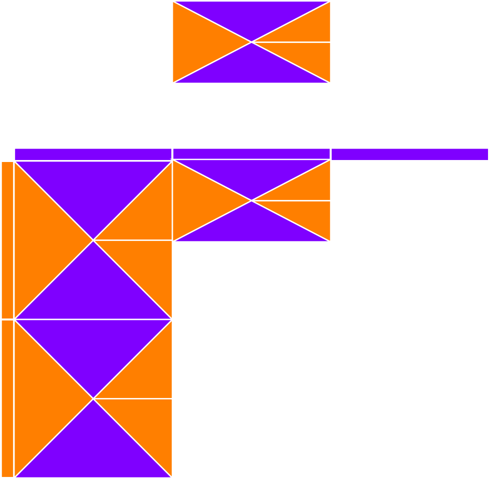
# Scalable tile puzzling (1964–78)



# Scalable tile puzzling (1964–78)



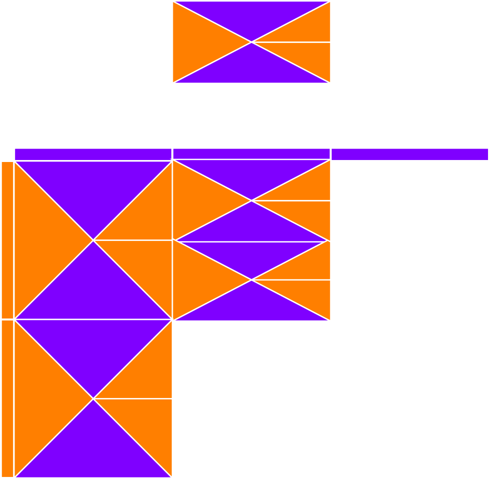
# Scalable tile puzzling (1964–78)



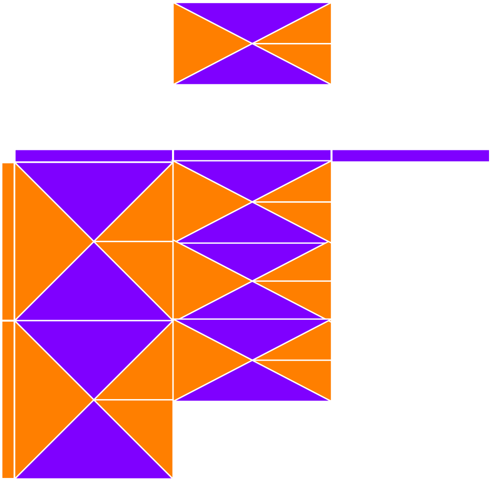


Universiteit Utrecht

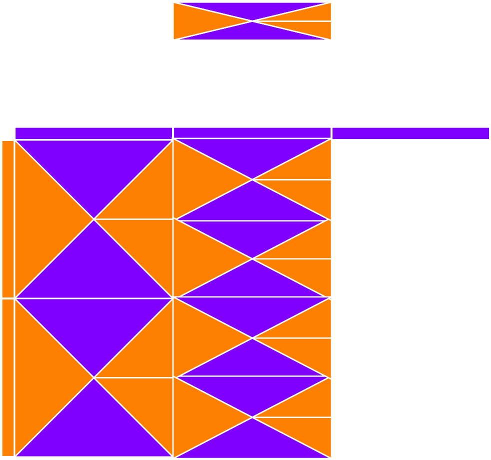
# Scalable tile puzzling (1964–78)



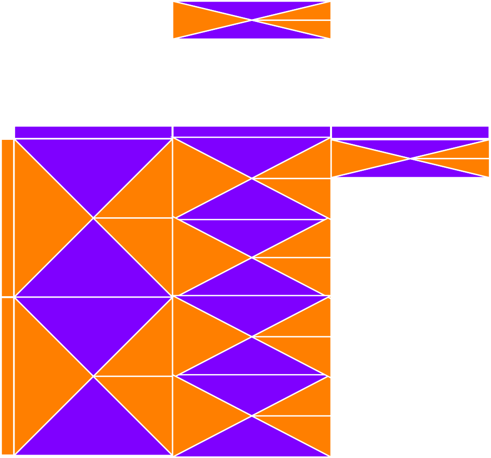
# Scalable tile puzzling (1964–78)



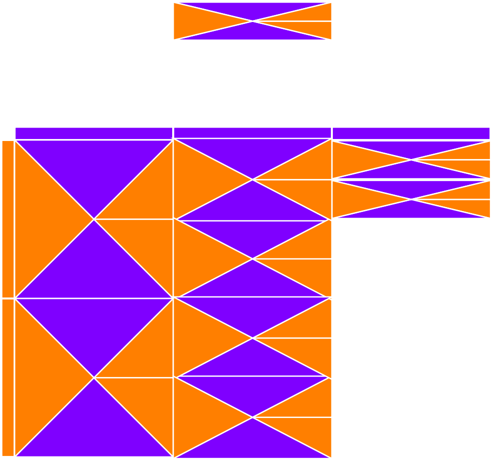
# Scalable tile puzzling (1964–78)



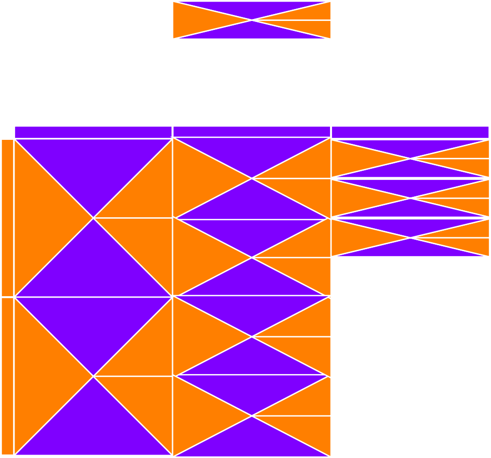
# Scalable tile puzzling (1964–78)



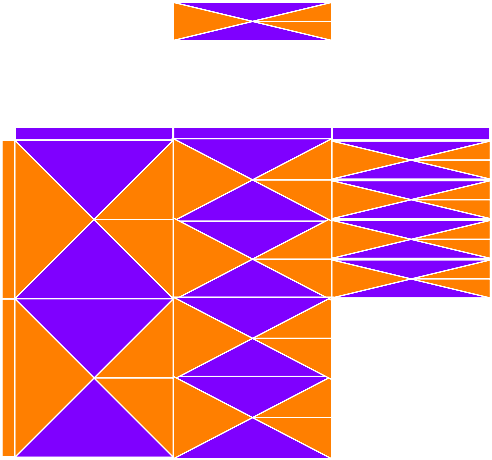
# Scalable tile puzzling (1964–78)



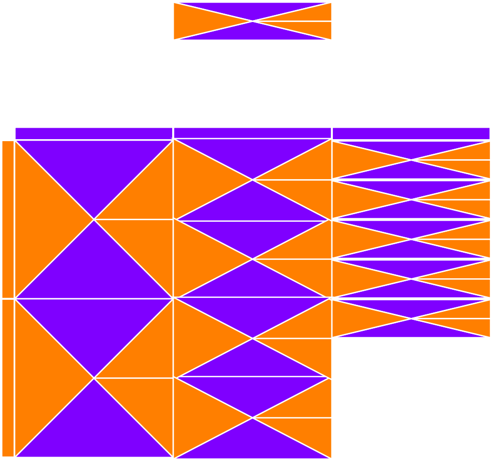
# Scalable tile puzzling (1964–78)



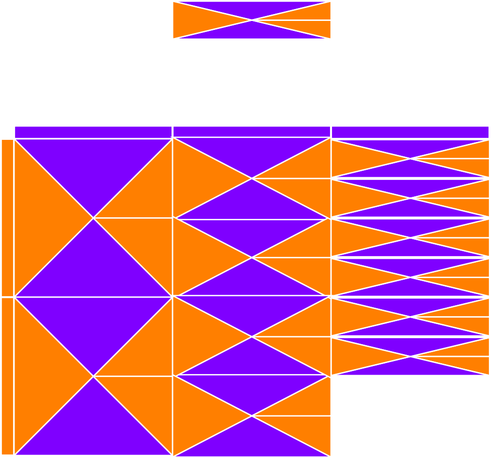
# Scalable tile puzzling (1964–78)



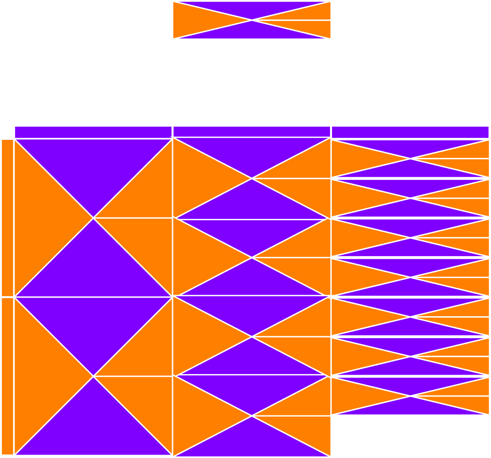
# Scalable tile puzzling (1964–78)



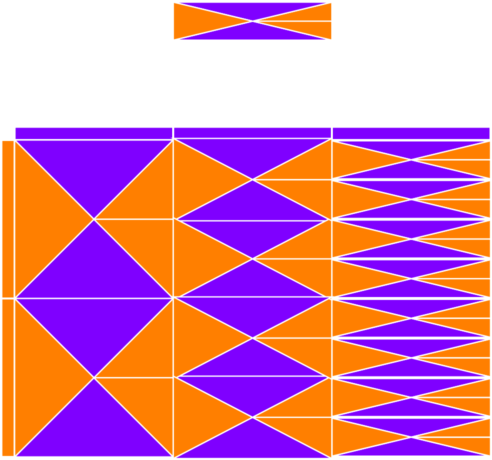
# Scalable tile puzzling (1964–78)



# Scalable tile puzzling (1964–78)



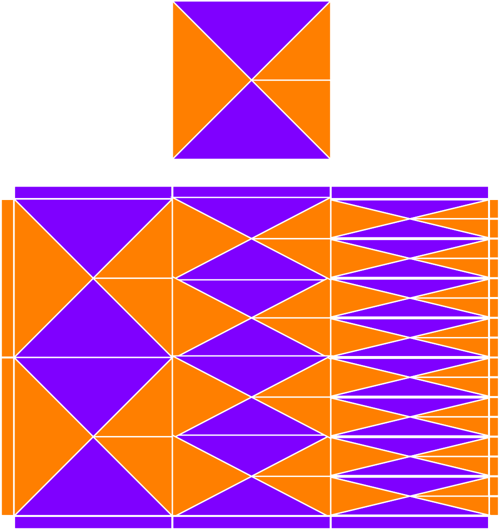
# Scalable tile puzzling (1964–78)



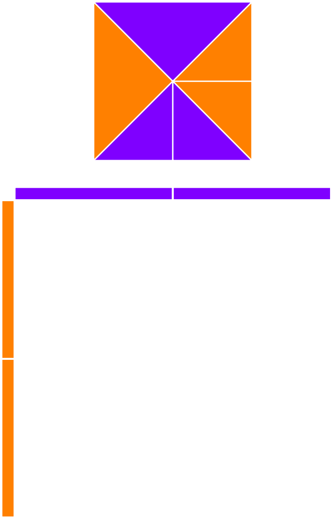


Universiteit Utrecht

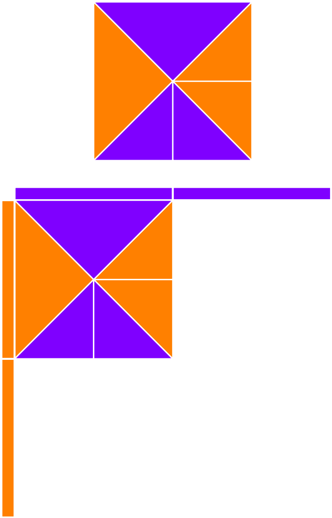
# Scalable tile puzzling (1964–78)



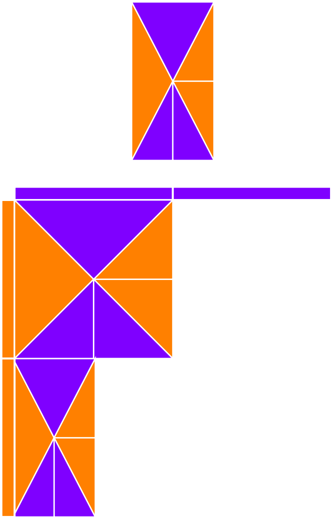
# Puzzling tiles. . . (1942–60)



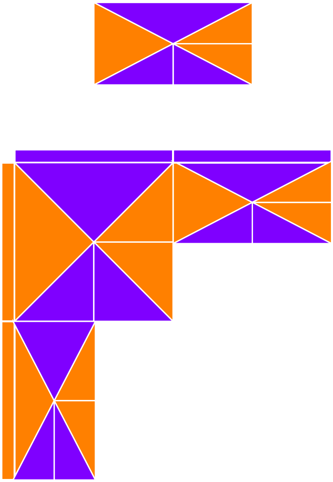
# Puzzling tiles. . . (1942–60)



# Puzzling tiles. . . (1942–60)



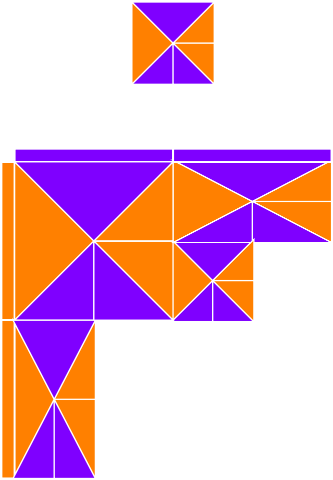
# Puzzling tiles. . . (1942–60)



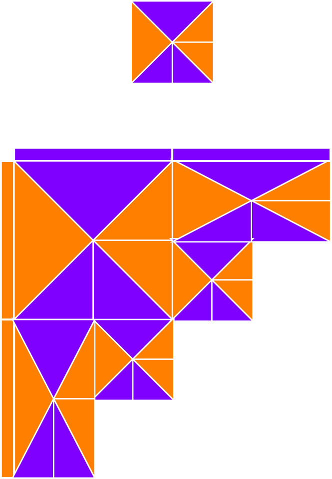


Universiteit Utrecht

# Puzzling tiles. . . (1942–60)



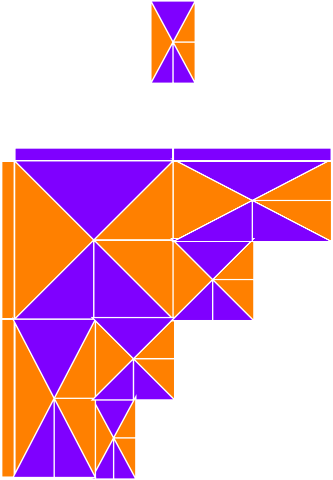
# Puzzling tiles. . . (1942–60)



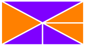


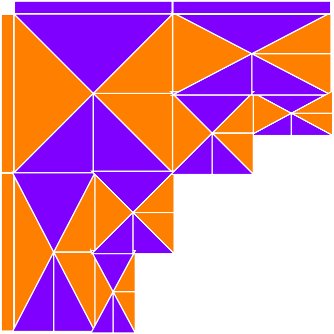
Universiteit Utrecht

# Puzzling tiles. . . (1942–60)

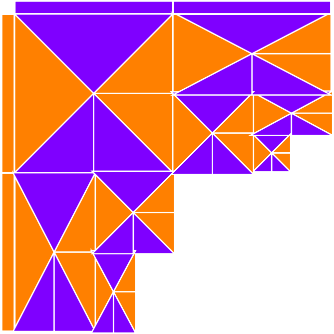


# Puzzling tiles. . . (1942–60)





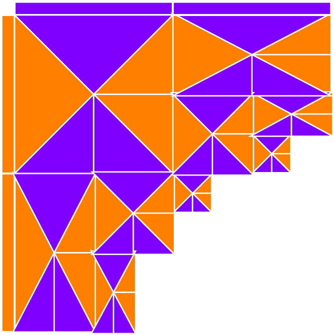
# Puzzling tiles. . . (1942–60)



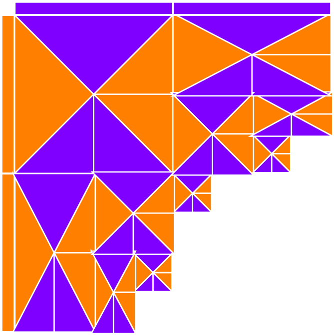


Universiteit Utrecht

# Puzzling tiles. . . (1942–60)



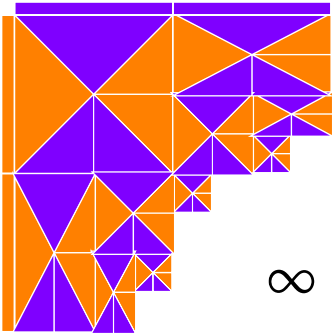
# Puzzling tiles. . . (1942–60)





Universiteit Utrecht

# Puzzling tiles. . . (1942–60)



# Puzzling tiling questions



Universiteit Utrecht

# Puzzling tiling questions

Given a set of tiles:

# Puzzling tiling questions

Given a set of tiles:

- For any situation, is there at least one fitting tile?

# Puzzling tiling questions



Given a set of tiles:

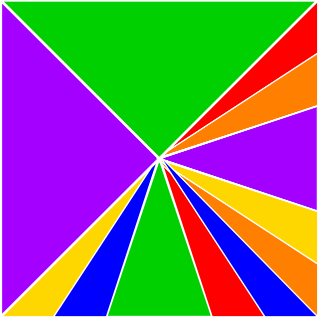
- For any situation, is there at least one fitting tile?
- Does a tiling strategy exist that terminates?

# Puzzling tiling questions

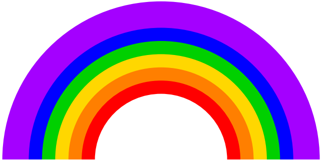
Given a set of tiles:

- For any situation, is there at least one fitting tile?
- Does a tiling strategy exist that terminates?
- Do all tiling strategies terminate?
- How many tiles are needed?

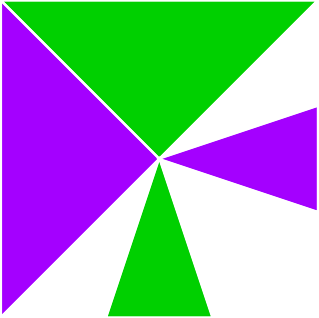
# Decreasing tiles (1978–94)



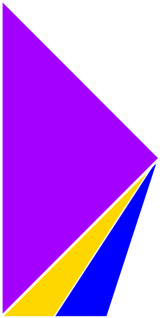
# Decreasing tiles (1978–94)



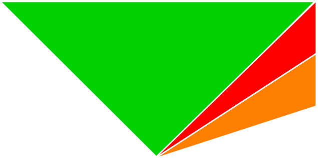
# Decreasing tiles (1978–94)



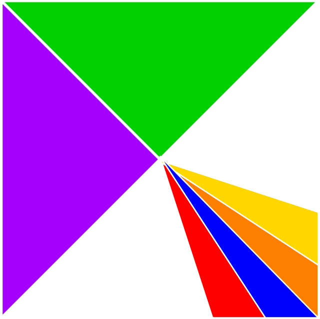
# Decreasing tiles (1978–94)



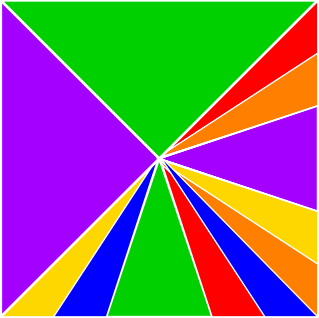
# Decreasing tiles (1978–94)



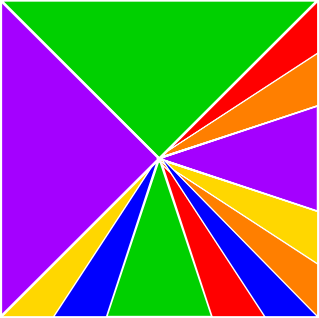
# Decreasing tiles (1978–94)



# Decreasing tiles (1978–94)



# Decreasing tiles (1978–94)

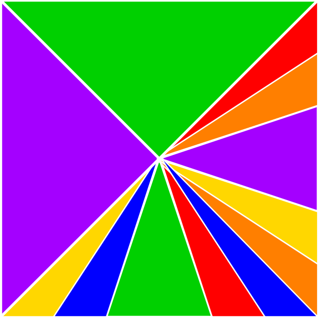




## Definition

set of such tiles decreasing if used colours well-ordered

# Decreasing tiles (1978–94)



## Definition

set of such tiles decreasing if used colours well-founded

# Terminating tiling strategy for decreasing tiles

Memorandum 78-08.

Issued August 1978.

A note on weak diamond properties.

by

N.G. de Bruijn.

# Terminating tiling strategy for decreasing tiles

Memorandum 78-08.

Issued August 1978.

A note on weak diamond properties.

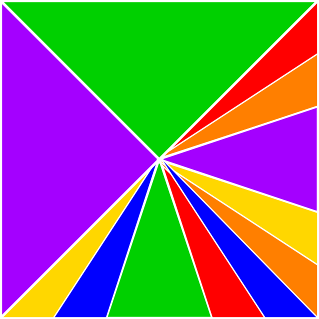
by

N.G. de Bruijn.

**Theorem**

*if tiles are decreasing, a tiling strategy exists that terminates*

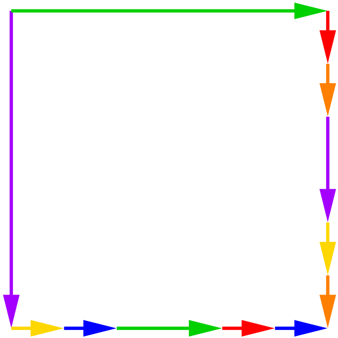
# Decreasing rewrite systems





Universiteit Utrecht

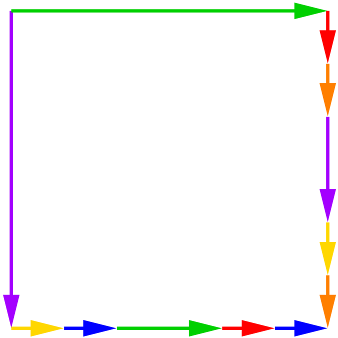
# Decreasing rewrite systems





Universiteit Utrecht

# Decreasing rewrite systems

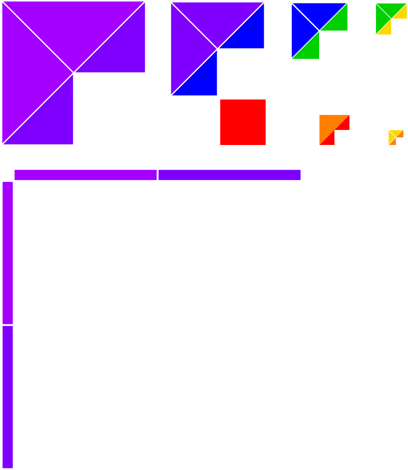




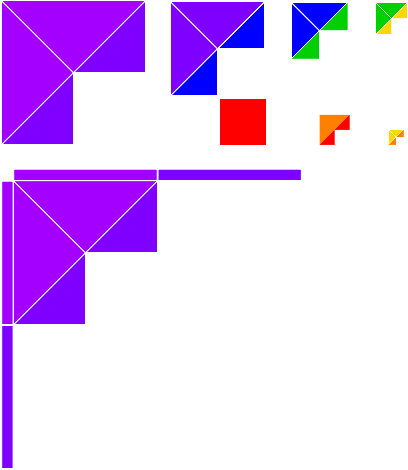
## Theorem

*if rewrite system decreasing, then confluent*

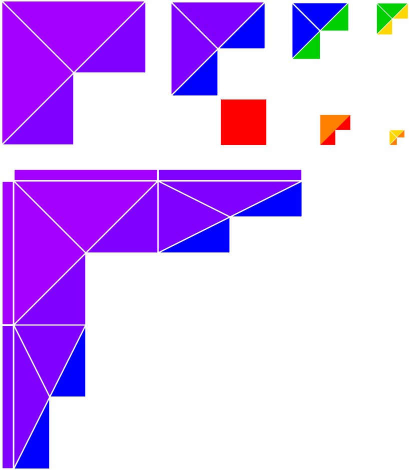
# Tiles with bite (2008–)



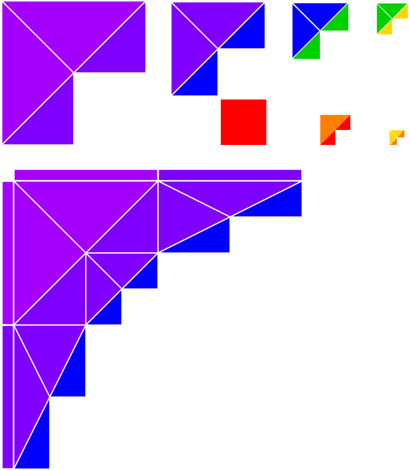
# Tiles with bite (2008–)



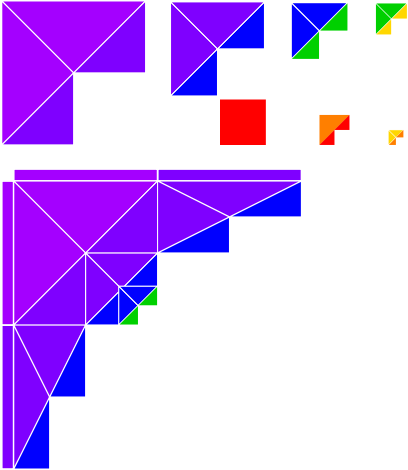
# Tiles with bite (2008–)



# Tiles with bite (2008–)



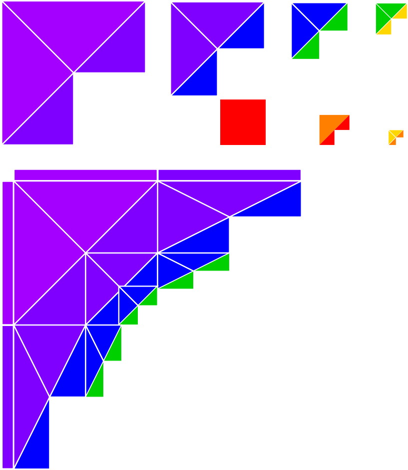
# Tiles with bite (2008–)



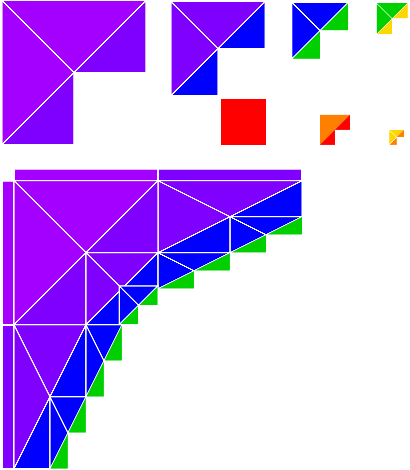


Universiteit Utrecht

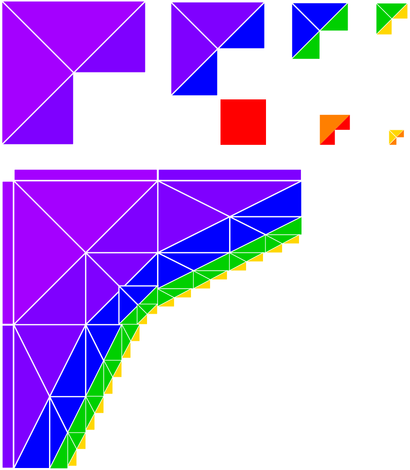
# Tiles with bite (2008–)



# Tiles with bite (2008–)



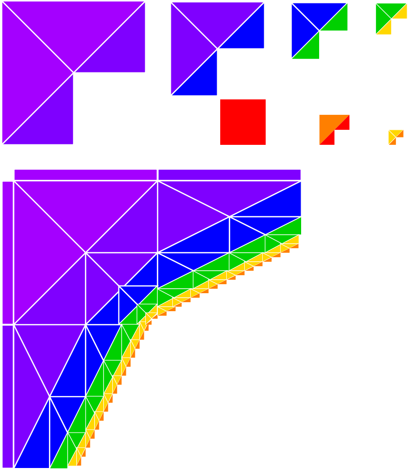
# Tiles with bite (2008–)



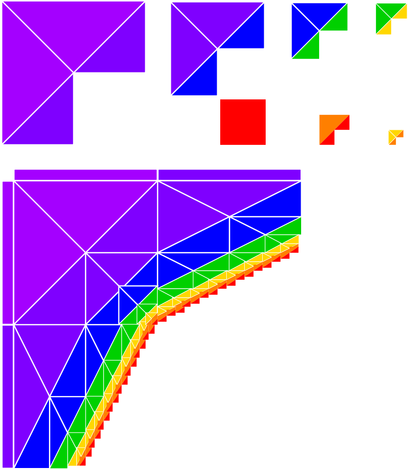


Universiteit Utrecht

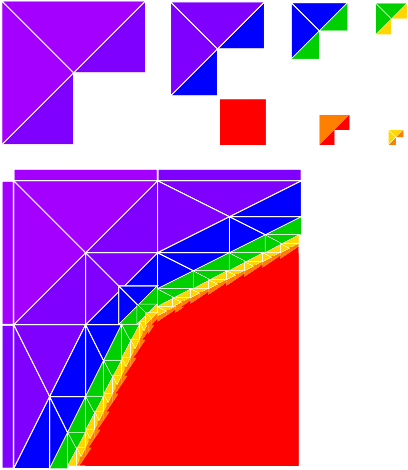
# Tiles with bite (2008–)



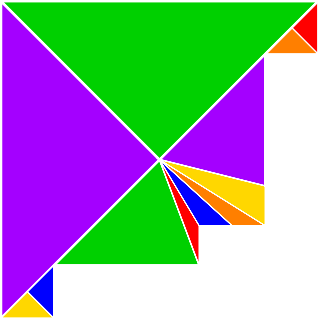
# Tiles with bite (2008–)



# Tiles with bite (2008–)



# Decreasing converted rewrite systems

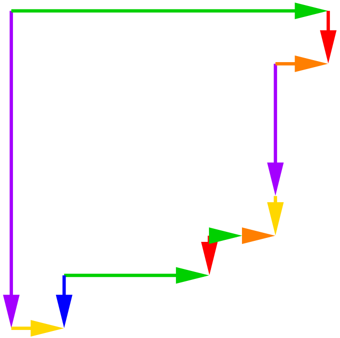




**Theorem**

*if tiles decreasing converted, a tiling strategy exists that terminates*

# Decreasing converted rewrite systems



Theorem

*if rewrite system decreasing converted, then confluent*

## $\exists \rightarrow \forall$



Given set of decreasing tiles:

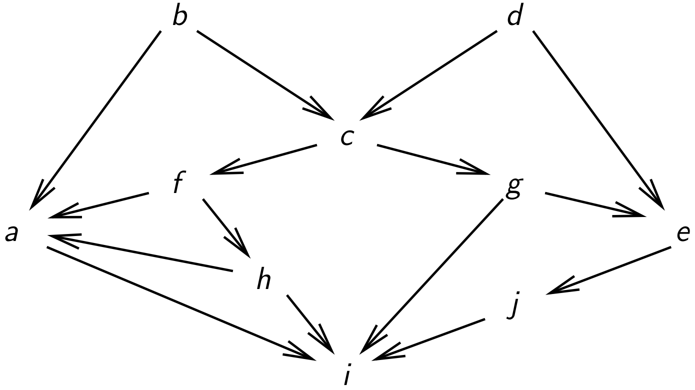
- Previous work: terminating tiling strategy exist

# $\exists \to \forall$

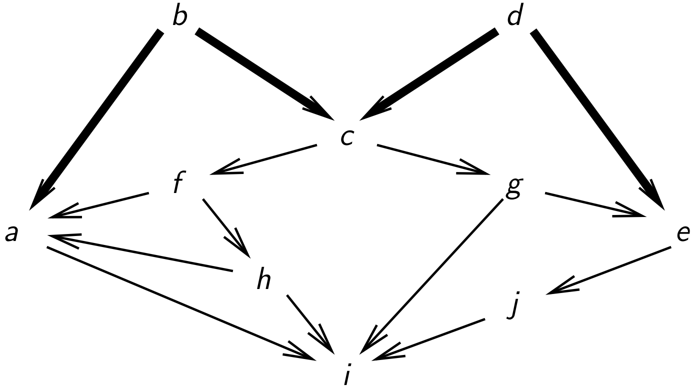
Given set of decreasing tiles:

- Previous work: terminating tiling strategy exist
- This talk: all tiling strategies terminate

# Transforming conversions

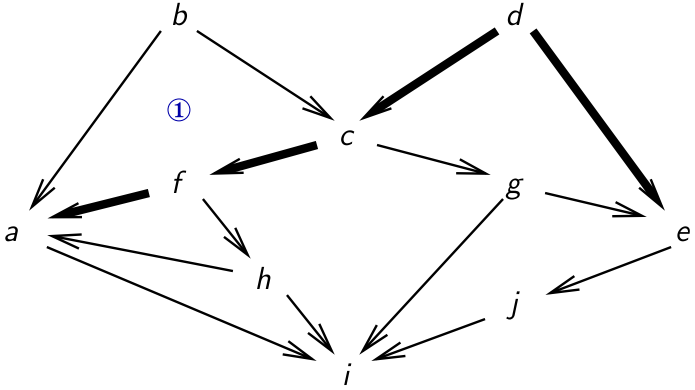


# Transforming conversions




$a$ convertible to $e$

# Transforming conversions

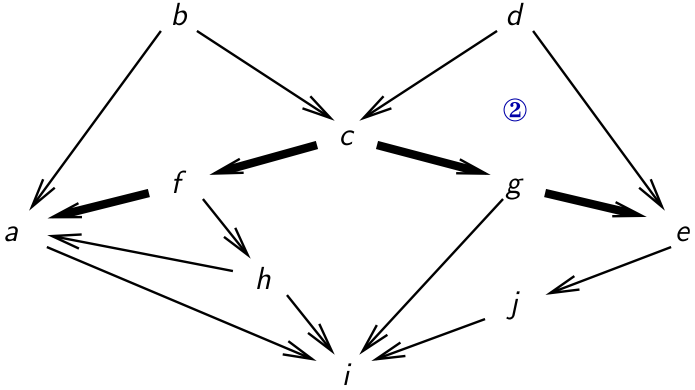



$a$ convertible to $e$



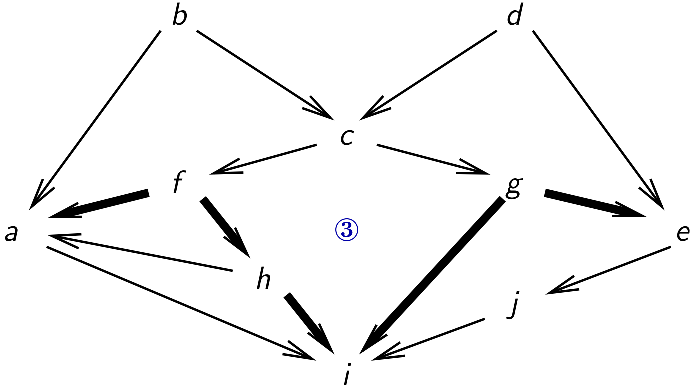
Universiteit Utrecht

# Transforming conversions



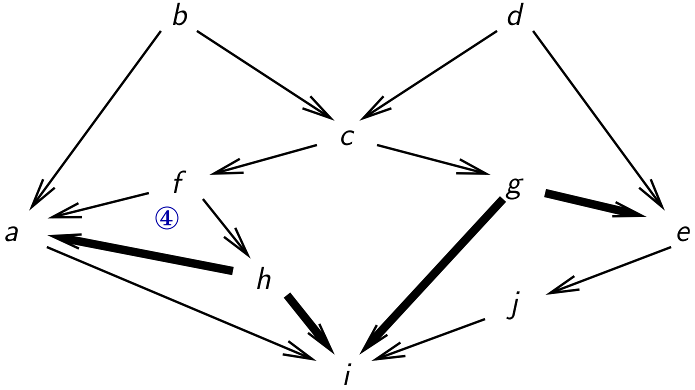

$a$ convertible to $e$

# Transforming conversions



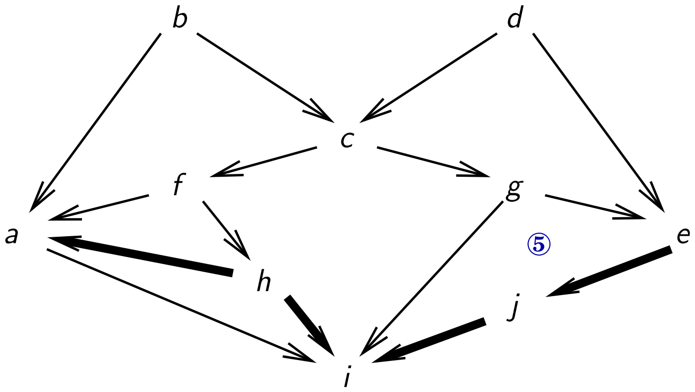

$a$ convertible to $e$

# Transforming conversions



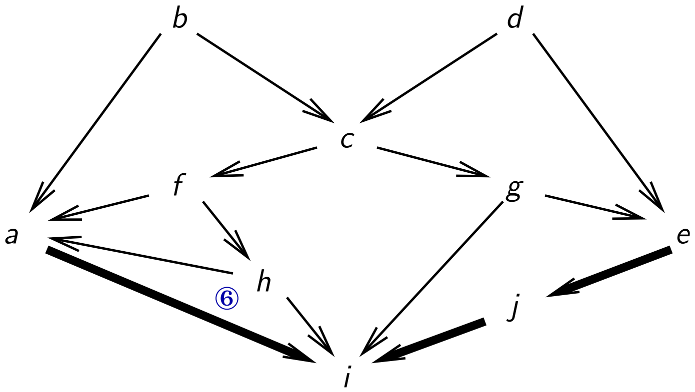

$a$ convertible to $e$

# Transforming conversions



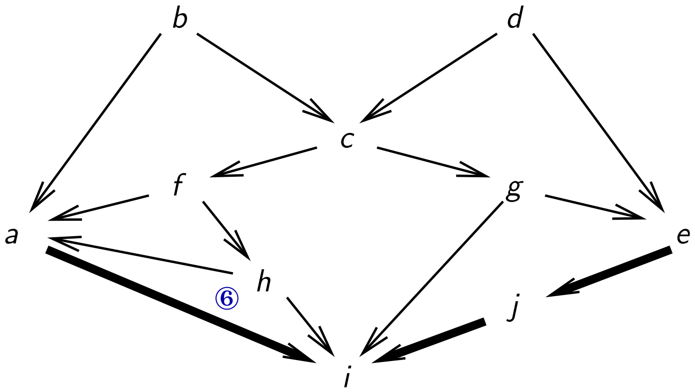

$a$ convertible to $e$

# Transforming conversions




$a$ convertible to $e$ by rewrite proof

# Transforming conversions




why do these transformations terminate?

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a}= \mathsf{b}}\,(\mathit{step}) \qquad \frac{}{\mathsf{a}= \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a}= \mathsf{b}}{\mathsf{b}= \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a}= \mathsf{b} \quad \mathsf{b}= \mathsf{c}}{\mathsf{a}= \mathsf{c}}\,(\cdot)$$

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a} = \mathsf{b}}\,(step) \qquad \frac{}{\mathsf{a} = \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a} = \mathsf{b}}{\mathsf{b} = \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a} = \mathsf{b} \quad \mathsf{b} = \mathsf{c}}{\mathsf{a} = \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a}= \mathsf{b}}\,(\mathit{step}) \qquad \frac{}{\mathsf{a}= \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a}= \mathsf{b}}{\mathsf{b}= \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a}= \mathsf{b} \quad \mathsf{b}= \mathsf{c}}{\mathsf{a}= \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is, = coincides with* $\leftrightarrow^*$

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a}= \mathsf{b}}\,(\mathit{step}) \qquad \frac{}{\mathsf{a}= \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a}= \mathsf{b}}{\mathsf{b}= \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a}= \mathsf{b} \quad \mathsf{b}= \mathsf{c}}{\mathsf{a}= \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is, = coincides with* $\leftrightarrow^*$

Methodology to show transformation of conversions terminates:

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a}= \mathsf{b}}\,(step) \qquad \frac{}{\mathsf{a}= \mathsf{a}}\,(\mathrm{e}) \qquad \frac{\mathsf{a}= \mathsf{b}}{\mathsf{b}= \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a}= \mathsf{b} \quad \mathsf{b}= \mathsf{c}}{\mathsf{a}= \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is, = coincides with* $\leftrightarrow^*$

Methodology to show transformation of conversions terminates:

- conversion is proof (in equational logic)



Universiteit Utrecht

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a} = \mathsf{b}}\,(\mathit{step}) \qquad \frac{}{\mathsf{a} = \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a} = \mathsf{b}}{\mathsf{b} = \mathsf{a}}\,(\text{-1}) \qquad \frac{\mathsf{a} = \mathsf{b} \quad \mathsf{b} = \mathsf{c}}{\mathsf{a} = \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

## Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is, = coincides with* $\leftrightarrow^*$

Methodology to show transformation of conversions terminates:

- conversion is proof (in equational logic)
- represent proof as proof term (term over $\{\mathrm{step}, -1, \cdot, e\}$)

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a}= \mathsf{b}}\,(\mathit{step}) \qquad \frac{}{\mathsf{a}= \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a}= \mathsf{b}}{\mathsf{b}= \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a}= \mathsf{b} \quad \mathsf{b}= \mathsf{c}}{\mathsf{a}= \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

## Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is,* $=$ *coincides with* $\leftrightarrow^*$

Methodology to show transformation of conversions terminates:

- conversion is proof (in equational logic)
- represent proof as proof term (term over $\{\mathrm{step}, -1, \cdot, e\}$)
- example: proof term $m^{-1} \cdot (\ell \cdot (k^{-1} \cdot m))$ represents conversion $a \leftarrow_m b \to_\ell c \leftarrow_k a \to_m b$

# Equational logic on nullary symbols (constants)

$$\frac{a \to b}{\mathsf{a} = \mathsf{b}}\,(step) \qquad \frac{}{\mathsf{a} = \mathsf{a}}\,(\mathsf{e}) \qquad \frac{\mathsf{a} = \mathsf{b}}{\mathsf{b} = \mathsf{a}}\,(\text{-}1) \qquad \frac{\mathsf{a} = \mathsf{b} \quad \mathsf{b} = \mathsf{c}}{\mathsf{a} = \mathsf{c}}\,(\cdot)$$

no derivation rules for congruence or substitution

## Theorem ((sub)Birkhoff)

*abstract rewriting is logical, that is, = coincides with* $\leftrightarrow^*$

Methodology to show transformation of conversions terminates:

- conversion is proof (in equational logic)
- represent proof as proof term (term over $\{\text{step}, -1, \cdot, e\}$)
- example: proof term $m^{-1} \cdot (\ell \cdot (k^{-1} \cdot m))$ represents conversion $a \leftarrow_m b \rightarrow_\ell c \leftarrow_k a \rightarrow_m b$
- equip proof terms with terminating rewrite relation compatible with decreasingness

# Conversions → proof terms → involutive monoid

## Definition

set with

- associative binary operation $\cdot$
- identity element $e$
- involutive anti-automorphism $^{-1}$

$$
\begin{array}{rcll}
(a \cdot b) \cdot c & = & a \cdot (b \cdot c) & \text{(associative)} \\
a \cdot e & = & a & \text{(right identity)} \\
e \cdot a & = & a & \text{(left identity)} \\
(a^{-1})^{-1} & = & a & \text{(involutive)} \\
(a \cdot b)^{-1} & = & b^{-1} \cdot a^{-1} & \text{(anti-automorphic)} \\
\\
\varepsilon^{-1} & = & \varepsilon & \text{(derived)}
\end{array}
$$

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- integers with addition, zero, unary minus

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- positive rationals with multiplication, one, inverse

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^+, \cdot, 1, \ ^{-1})$)
- natural numbers with addition, zero, identity map

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^+, \cdot, 1, {}^{-1})$)
- multisets with multiset sum, empty multiset, identity map

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^{+}, \cdot, 1, \ ^{-1})$)
- commutative monoid with identity map

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^{+}, \cdot, 1, \ {}^{-1}))$
- commutative monoid (examples $(\mathbb{N}, +, 0)$, $([L], \uplus, [\ ]))$
- diagrams of $\diagdown$ with gluing, point, mirroring in vertical axis

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^+, \cdot, 1, \ ^{-1})$)
- commutative monoid (examples $(\mathbb{N}, +, 0)$, $([L], \uplus, [\ ])$)
- diagrams of $\diagdown$ with gluing, point, mirroring in vertical axis
- number pairs with pointwise addition, $(0, 0)$, swapping

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^+, \cdot, 1, \ ^{-1}))$
- commutative monoid (examples $(\mathbb{N}, +, 0)$, $([L], \uplus, [\ ]))$
- diagrams of $\setminus$ with gluing, point, mirroring in vertical axis
- number triples with composition given by $(n_1, m_1, k_1) \cdot (n_2, m_2, k_2) = (n_1 + n_2, m_1 + k_1 \cdot n_2 + m_2, k_1 + k_2)$, zero $(0, 0, 0)$, involution $(n, m, k)^{-1} = (k, m, n)$

# Involutive monoid examples

- $\{*\}$ with binary, nullary, unary constant-$*$ map
- group (examples $(\mathbb{Z}, +, 0, -)$, $(\mathbb{Q}^+, \cdot, 1, \ ^{-1})$)
- commutative monoid (examples $(\mathbb{N}, +, 0)$, $([L], \uplus, [\ ])$)
- diagrams of $\diagdown$ with gluing, point, mirroring in vertical axis
- number triples with composition given by $(n_1, m_1, k_1) \cdot (n_2, m_2, k_2) = (n_1 + n_2, m_1 + k_1 \cdot n_2 + m_2, k_1 + k_2)$, zero $(0, 0, 0)$, involution $(n, m, k)^{-1} = (k, m, n)$

$$
\begin{aligned}
& ((n_1, m_1, k_1) \cdot (n_2, m_2, k_2)) \cdot (n_3, m_3, k_3) \\
= \quad & (n_1 + n_2, m_1 + k_1 \cdot n_2 + m_2, k_1 + k_2) \cdot (n_3, m_3, k_3) \\
= \quad & (n_1 + n_2 + n_3, m_1 + k_1 \cdot n_2 + m_2 + (k_1 + k_2) \cdot n_3 + m_3, k_1 + k_2 + k_3) \\
= \quad & (n_1 + n_2 + n_3, m_1 + k_1 \cdot (n_2 + n_3) + m_2 + k_2 \cdot n_3 + m_3, k_1 + k_2 + k_3) \\
= \quad & (n_1, m_1, k_1) \cdot (n_2 + n_3, m_2 + k_2 \cdot n_3 + m_3, k_2 + k_3) \\
= \quad & (n_1, m_1, k_1) \cdot ((n_2, m_2, k_2) \cdot (n_3, m_3, k_3))
\end{aligned}
$$



Universiteit Utrecht

# Involutive monoid of French strings

## Definition

- French letter is an accented (acute or grave) letter

# Involutive monoid of French strings



## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition ␣  èvèn juxtaposed to ḱńíḱté gives èvèǹḱńíḱté

# Involutive monoid of French strings



## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition $\llcorner\!\lrcorner$
- empty string $\varepsilon$

# Involutive monoid of French strings

## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition ␣
- empty string $\varepsilon$
- mirroring $^{-1}$   t̀èl̀k̀èǹs̀ mirrors śńékĺét́

# Involutive monoid of French strings

## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition $\_$
- empty string $\varepsilon$
- mirroring $\phantom{x}^{-1}$
- $\widehat{L}$ set of French Strings on $L$ ($\hat{a}$ for either $\grave{a}$ or $\acute{a}$)

# Involutive monoid of French strings

## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition ␣
- empty string $\varepsilon$
- mirroring $\;^{-1}$
- $\widehat{L}$ set of French Strings on $L$

# Involutive monoid of French strings



## Definition

- French letter is an accented (acute or grave) letter
- juxtaposition $\_$
- empty string $\varepsilon$
- mirroring $\;^{-1}$
- $\widehat{L}$ set of French Strings on $L$

letter markup (representation preserves length,prefix,suffix)

# Boustrophedon

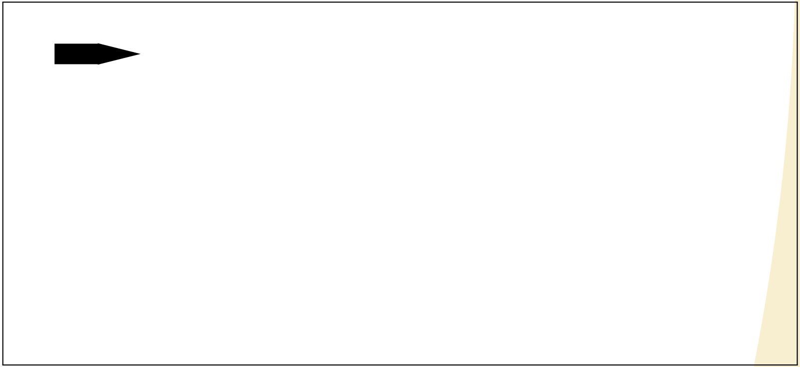
Gortyn code, Crete, 5th century B.C. (wikipedia)

# Boustrophedon



Universiteit Utrecht

# Boustrophedon

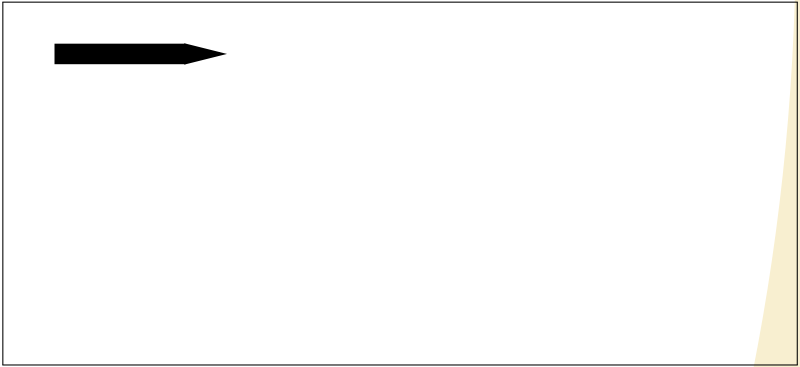


how the cow ploughs



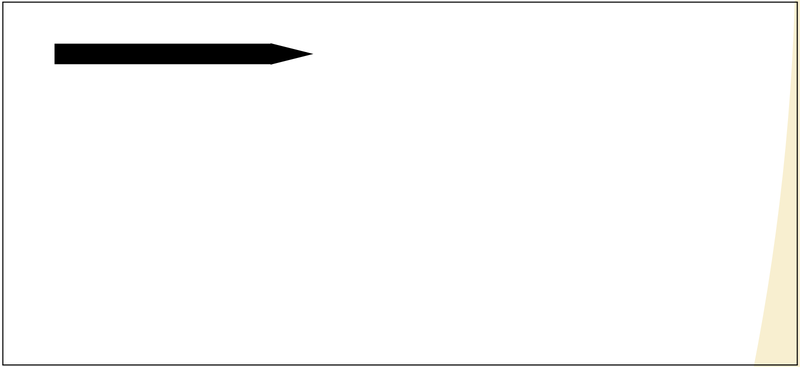
Universiteit Utrecht

# Boustrophedon



how the cow ploughs

# Boustrophedon



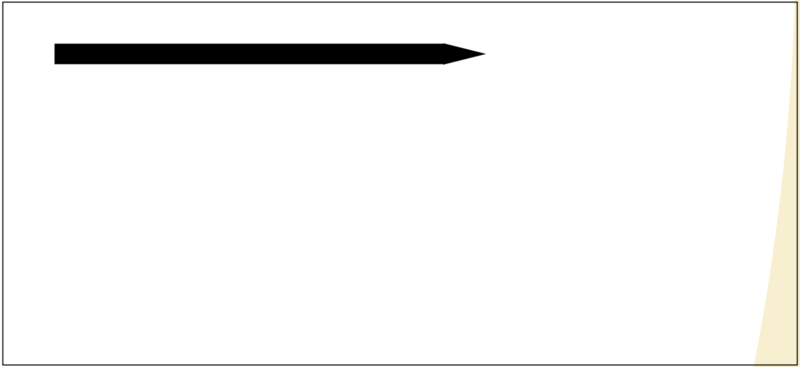
how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

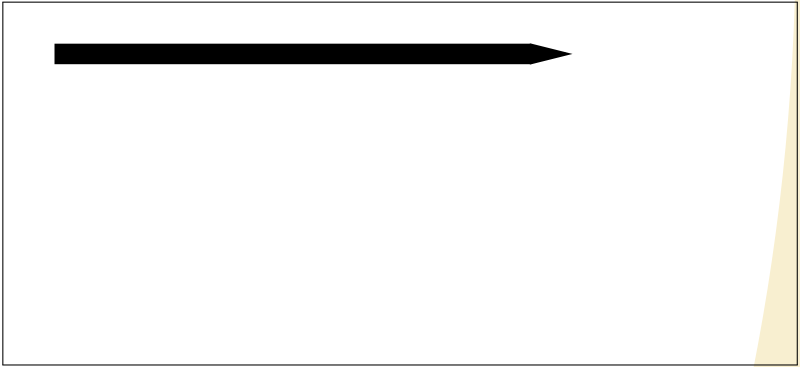


how the cow ploughs



Universiteit Utrecht

# Boustrophedon



how the cow ploughs



Universiteit Utrecht

# Boustrophedon



how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon

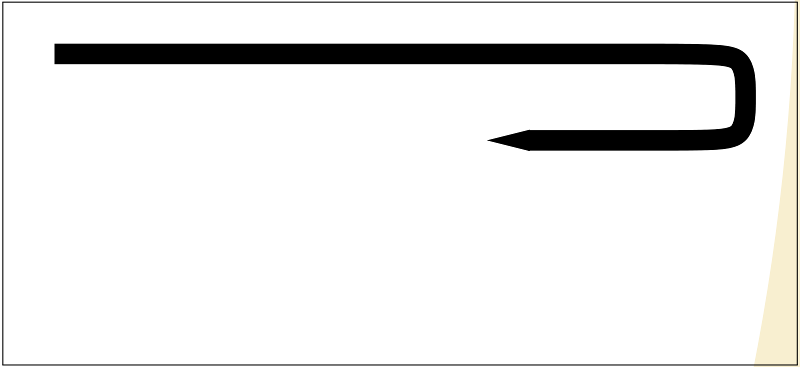


how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon

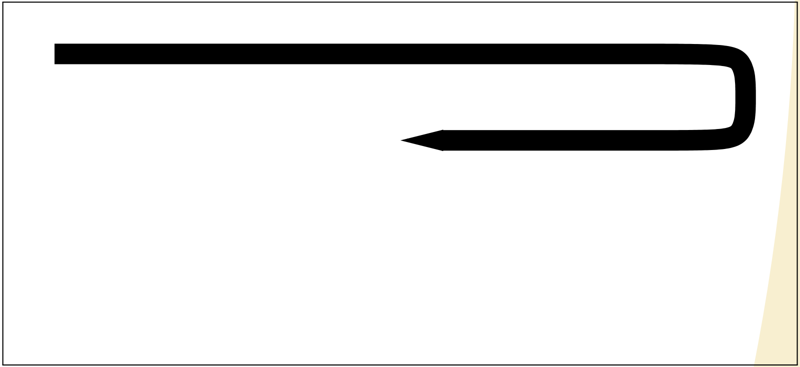


how the cow ploughs



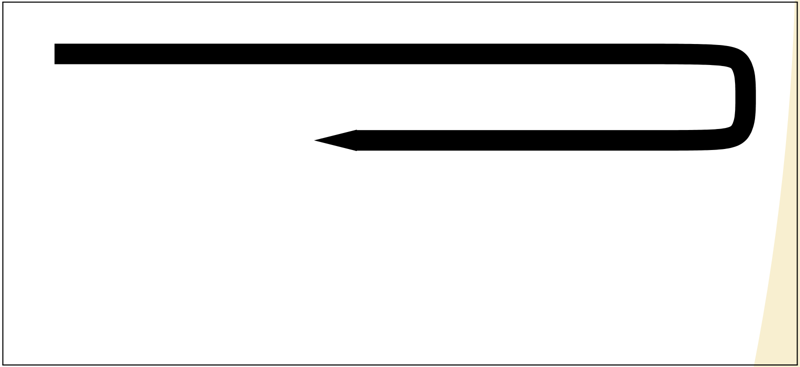
Universiteit Utrecht

# Boustrophedon



how the cow ploughs

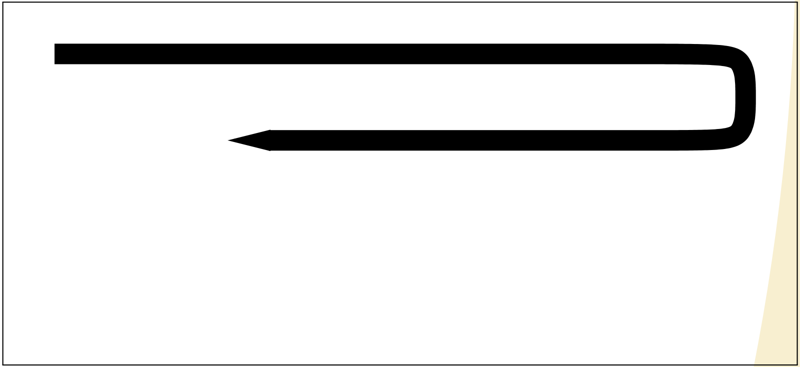
# Boustrophedon



how the cow ploughs

# Boustrophedon





how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon

how the cow ploughs



Universiteit Utrecht

# Boustrophedon

how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon

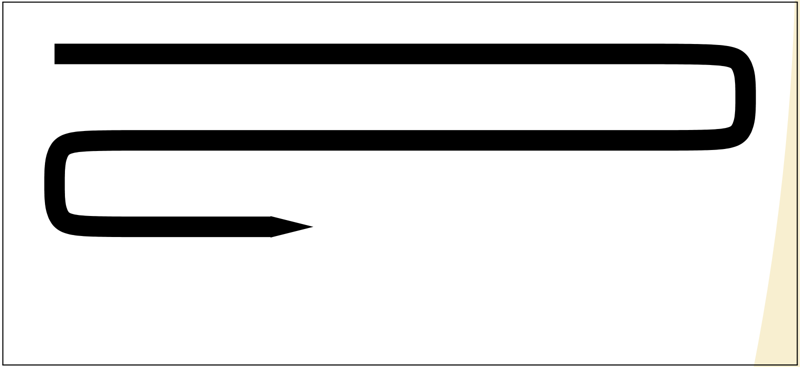


how the cow ploughs

# Boustrophedon

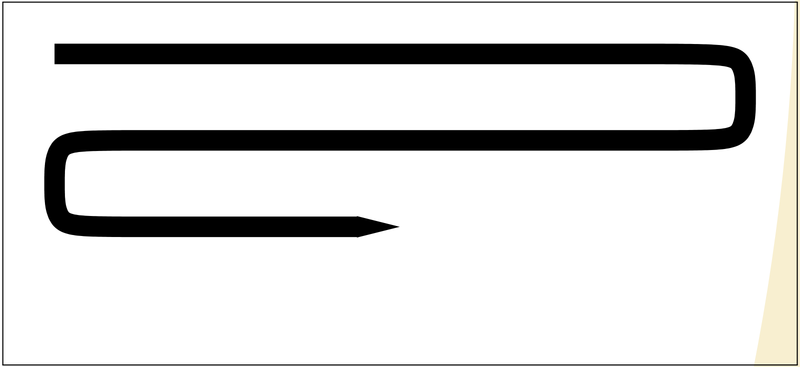
how the cow ploughs

# Boustrophedon



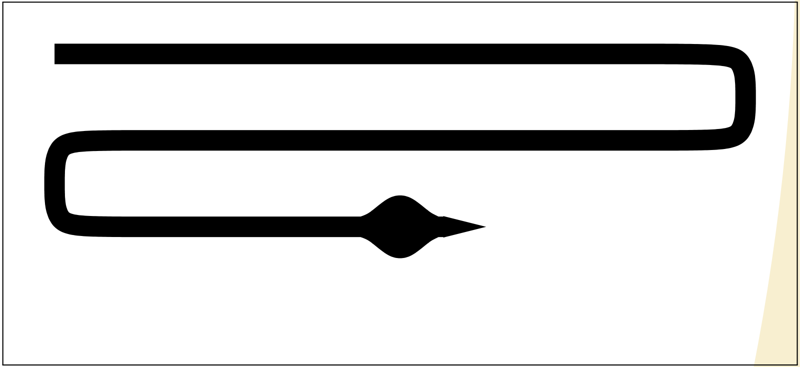
how the cow ploughs

# Boustrophedon



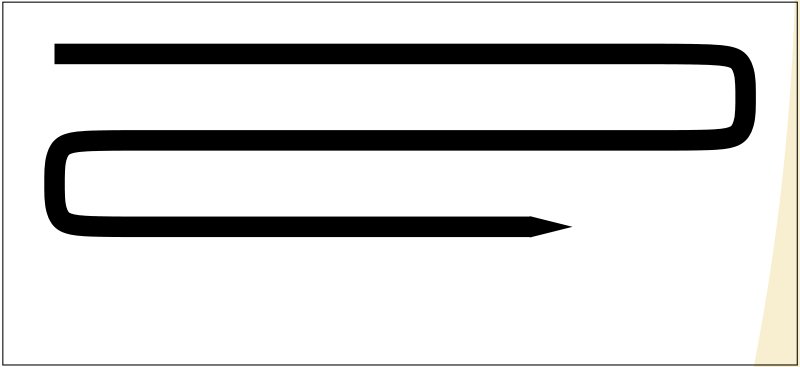
how the cow ploughs

# Boustrophedon



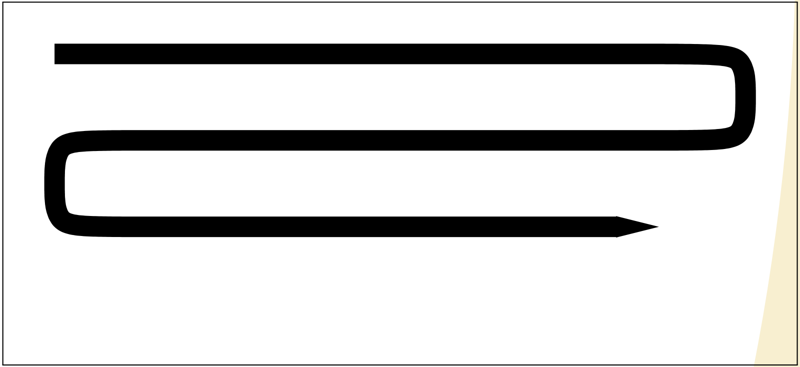
how the plow coughs

# Boustrophedon



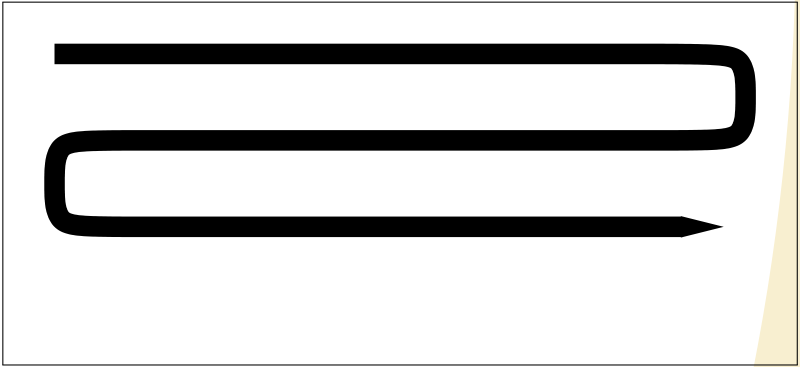
how the cow ploughs

# Boustrophedon



how the cow ploughs

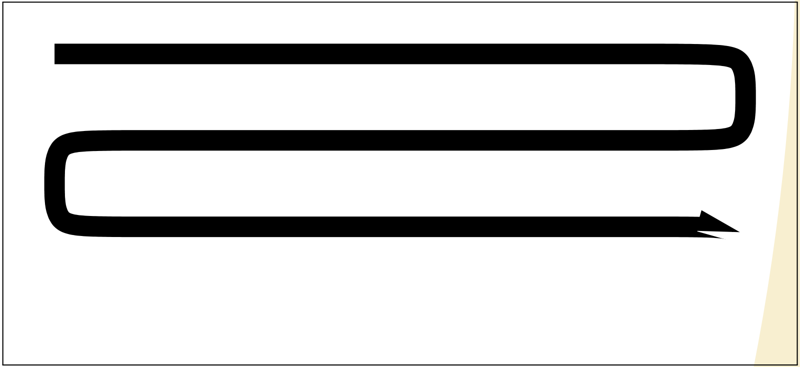
# Boustrophedon



how the cow ploughs

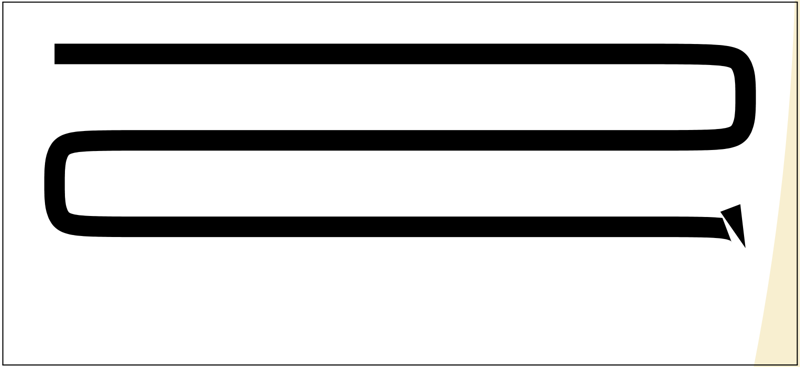
# Boustrophedon





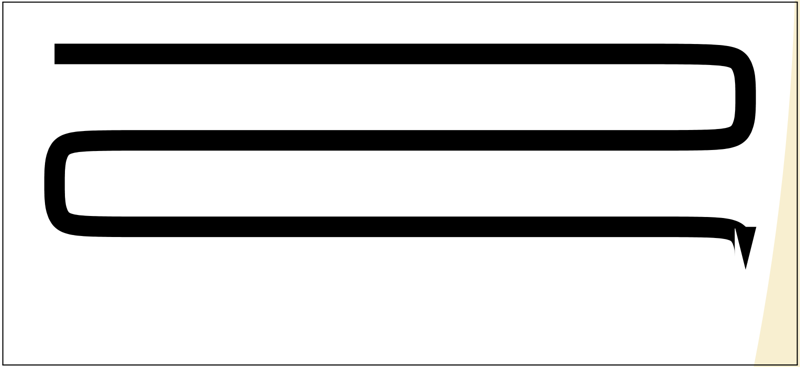
how the cow ploughs

# Boustrophedon



how the cow ploughs

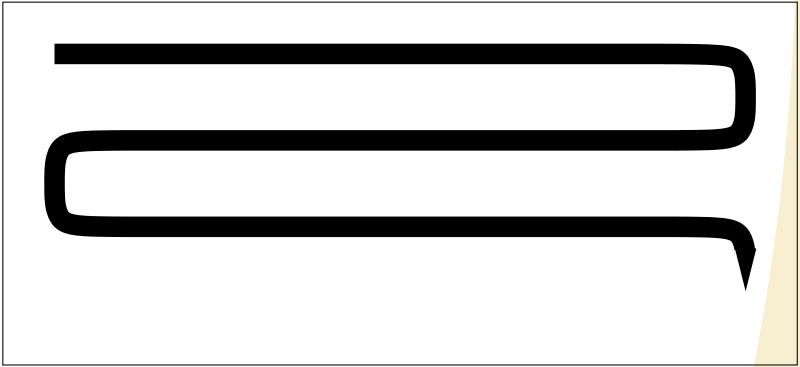
# Boustrophedon



how the cow ploughs

# Boustrophedon

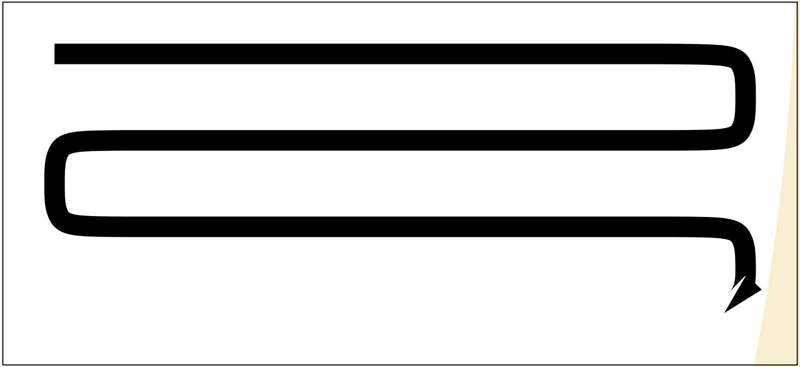




how the cow ploughs

# Boustrophedon

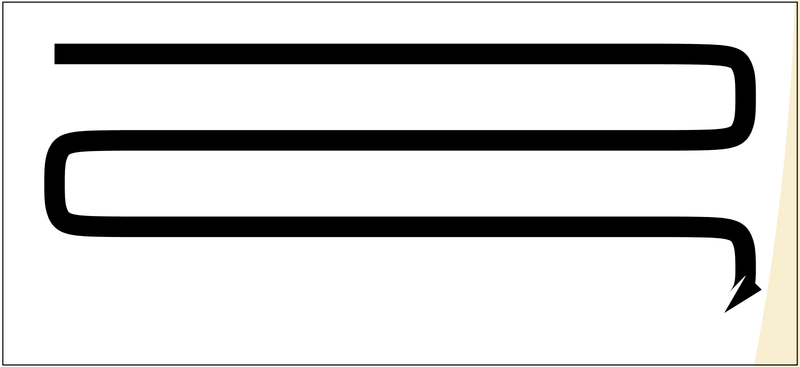




how the cow ploughs

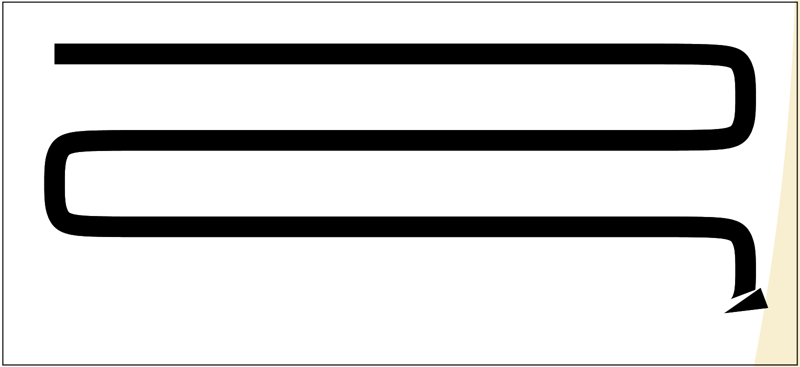
# Boustrophedon





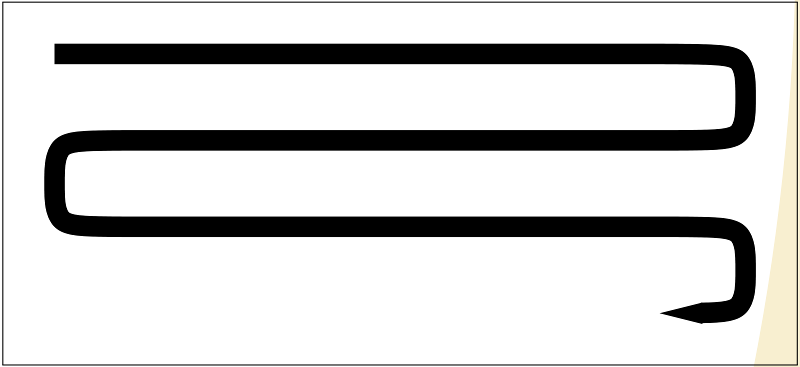
how the cow ploughs

# Boustrophedon



how the cow ploughs

# Boustrophedon



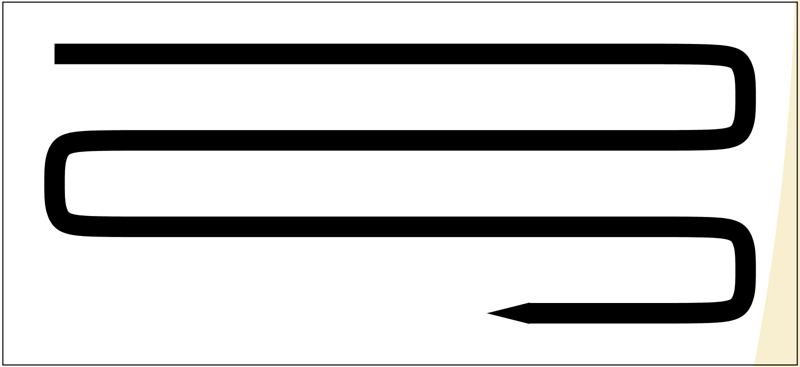
how the cow ploughs

# Boustrophedon

how the cow ploughs

# Boustrophedon



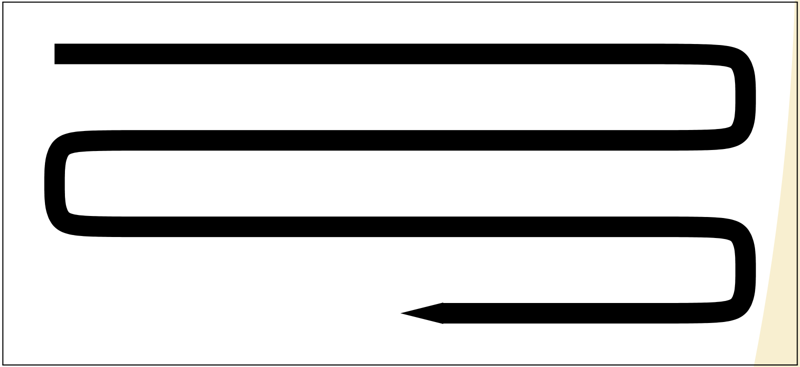


how the cow ploughs

Universiteit Utrecht

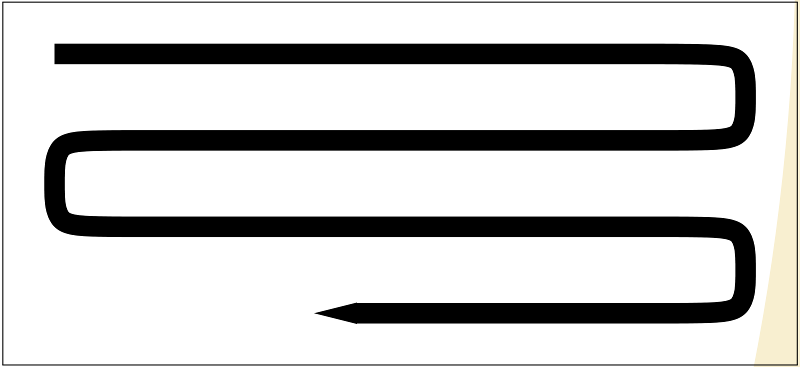
# Boustrophedon





how the cow ploughs

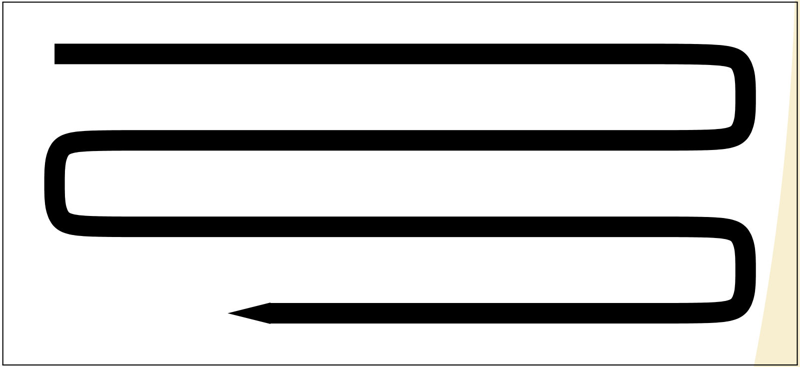
# Boustrophedon



how the cow ploughs

# Boustrophedon





how the cow ploughs

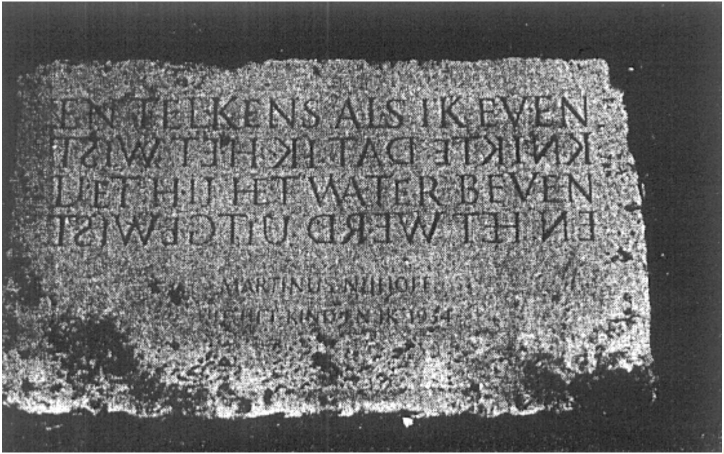
# Boustrophedon



how the cow ploughs

# Boustrophedon






Martinus Nijhoff, Het kind en ik, Nieuwe Gedichten, 1934
(Hortus Botanicus, Universiteitsmuseum Utrecht, next to pond)

# Boustrophedon



EN TELKENS ALS IK EVEN
KNIKTE DAT IK HET WIST
LIET HIJ HET WATER BEVEN
EN HET WERD UITGEWIST

# Involutive monoid homomorphisms



Definition

homomorphism is map preserving operations

Examples

- involutive monoid to itself (identity)

# Involutive monoid homomorphisms



## Definition

homomorphism is map preserving operations

## Examples

- involutive monoid to itself (identity)
- French strings → number pairs (grave,acute)
  ćèńàr̀ $\mapsto (3, 2)$

# Involutive monoid homomorphisms



### Definition

homomorphism is map preserving operations

### Examples

- involutive monoid to itself (identity)
- number pairs → natural numbers (sum)
  $(3, 2) \mapsto 5$

Universiteit Utrecht

# Involutive monoid homomorphisms



## Definition

homomorphism is map preserving operations

## Examples

- involutive monoid to itself (identity)
- French strings → natural numbers (length)
  composition of previous two

# Involutive monoid homomorphisms



### Definition

homomorphism is map preserving operations

### Examples

- involutive monoid to itself (identity)
- French strings → natural numbers (length)
- French strings → multisets (letters)
  $\acute{b}\acute{a}\acute{r}\grave{b}\grave{a}\acute{r}\acute{o} \mapsto [a, a, b, b, o, r, r]$

# Involutive monoid homomorphisms

## Definition

homomorphism is map preserving operations

## Examples

- involutive monoid to itself (identity)
- French strings → natural numbers (length)
- French strings → multisets (letters)
- French strings → diagrams

ćèńàr̀ ↦ 

# Involutive monoid homomorphisms

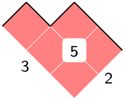


## Definition

homomorphism is map preserving operations

## Examples

- involutive monoid to itself (identity)
- French strings → natural numbers (length)
- French strings → multisets (letters)
- diagrams → triples

 $\mapsto (3, 5, 2)$ cf. 

# Involutive monoid homomorphisms



## Definition

homomorphism is map preserving operations

## Examples

- involutive monoid to itself (identity)
- French strings → natural numbers (length)
- French strings → multisets (letters)
- French strings → triples (area)
  composition of previous two

# Free involutive monoid on generators



**Theorem**

*French strings on $L$ give free involutive monoid on $L$*

# Free involutive monoid on generators



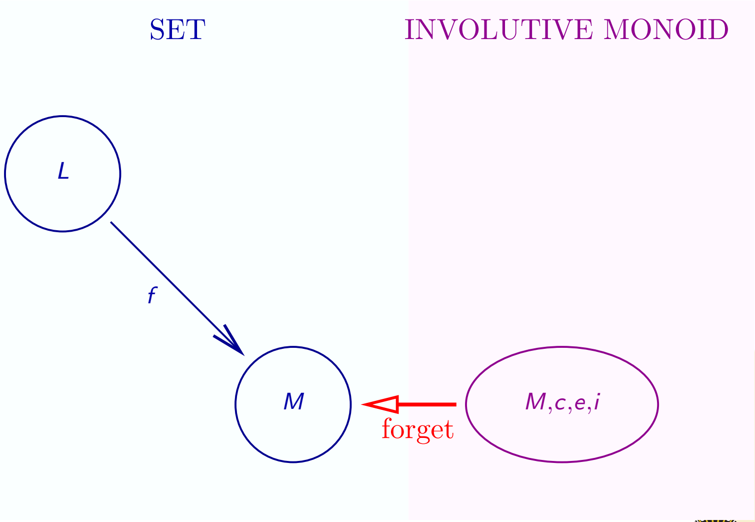
**Theorem**
*French strings on $L$ give free involutive monoid on $L$*

French string : conversion = string : reduction

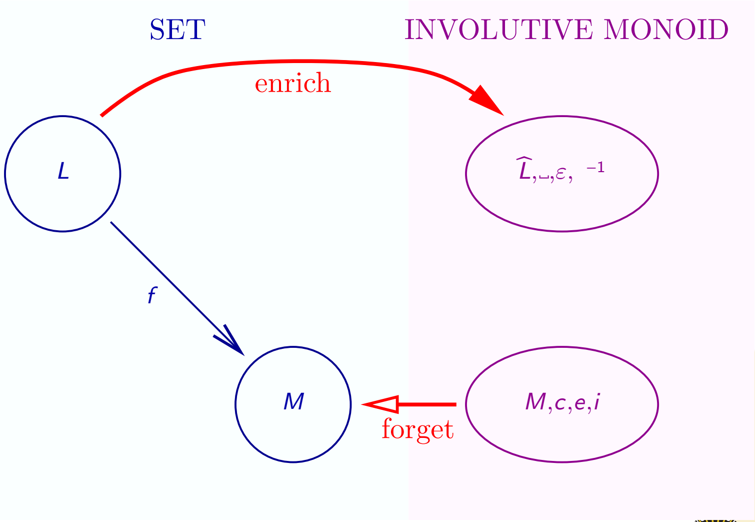
# Freeness of involutive monoid of French Strings

$$\begin{tikzcd} L \arrow[dr, "f"'] & \\ & M \end{tikzcd}$$

# Freeness of involutive monoid of French Strings



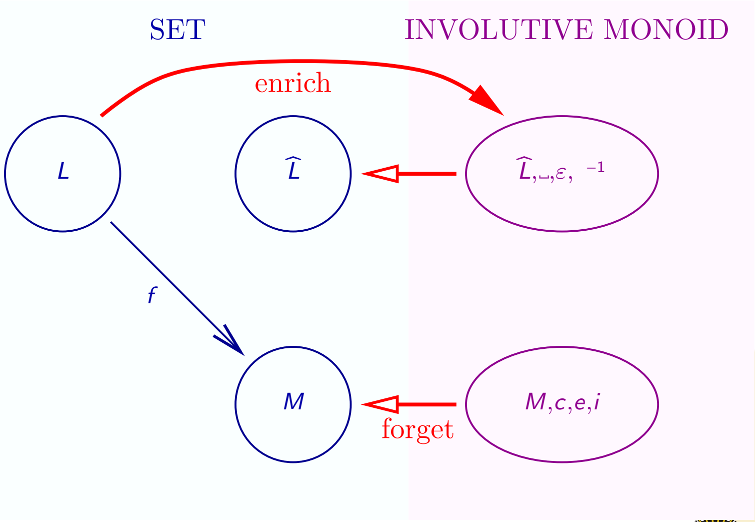
# Freeness of involutive monoid of French Strings



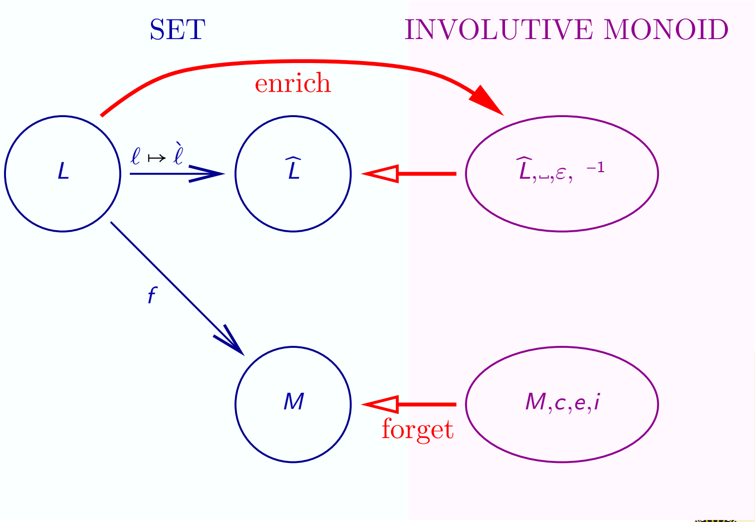



Universiteit Utrecht

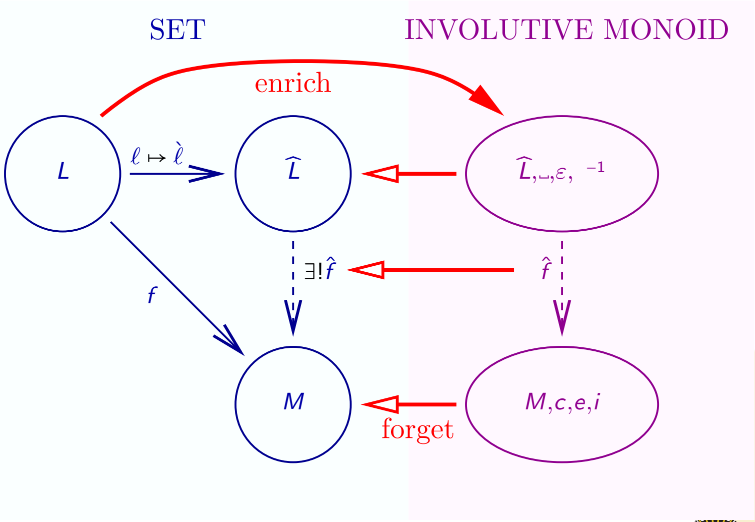
# Freeness of involutive monoid of French Strings



# Freeness of involutive monoid of French Strings



# Freeness of involutive monoid of French Strings






Universiteit Utrecht

# Free involutive monoid on generators

## Theorem

*French strings on $L$ give free involutive monoid on $L$*



Universiteit Utrecht

# Free involutive monoid on generators

## Theorem

*French strings on $L$ give free involutive monoid on $L$*

## Proof.

$\widehat{L}$ in bijection via $\grave{\ell} \mapsto \ell$, with union of $\{e\}$ and

$$N ::= \ell \mid i(\ell) \mid c(\ell, N) \mid c(i(\ell), N)$$

# Free involutive monoid on generators

Theorem

*French strings on $L$ give free involutive monoid on $L$*

Proof.

$\widehat{L}$ in bijection via $\grave{\ell} \mapsto \ell$, with union of $\{e\}$ and

$$N ::= \ell \mid i(\ell) \mid c(\ell, N) \mid c(i(\ell), N)$$

$N$ set of normal forms on $L$ for TRS completing axioms

$$\begin{aligned} c(c(x,y),z) &\to c(x,c(y,z)) \\ c(x,e) &\to x \\ c(e,x) &\to x \\ i(i(x)) &\to x \\ i(c(x,y)) &\to c(i(y),i(x)) \\ i(e) &\to e \end{aligned}$$

# Involutive monoid on French terms $L^\sharp$

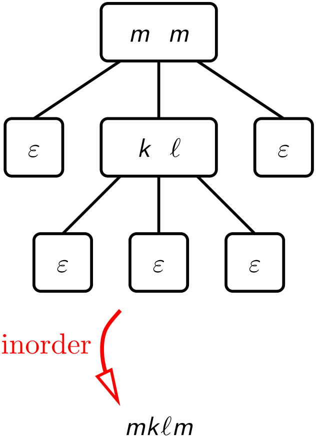
## Definition

certain terms on certain French strings

# Involutive monoid on French terms $L^\sharp$

## Definition

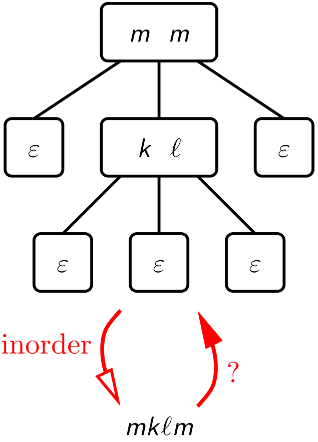
terms on strings



# Involutive monoid on French terms $L^\sharp$

## Definition

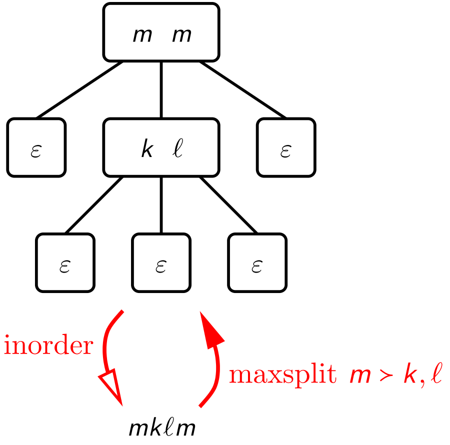
terms on strings



# Involutive monoid on French terms $L^\sharp$

## Definition

terms on strings on $>$-ordered letters



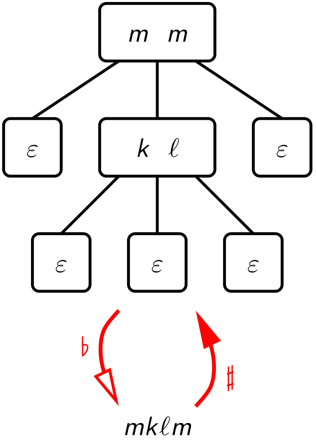



Universiteit Utrecht

# Involutive monoid on French terms $L^\sharp$

## Definition

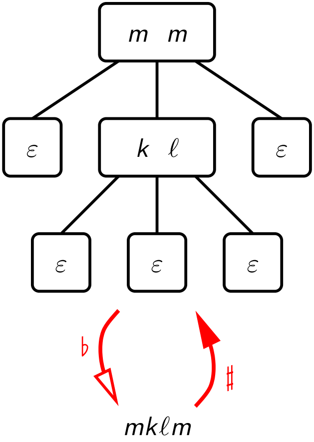
terms on strings on $>$-ordered letters



# Involutive monoid on French terms $L^\sharp$

## Definition

terms on strings on $\succ$-ordered letters where $\flat \circ \sharp$ identity



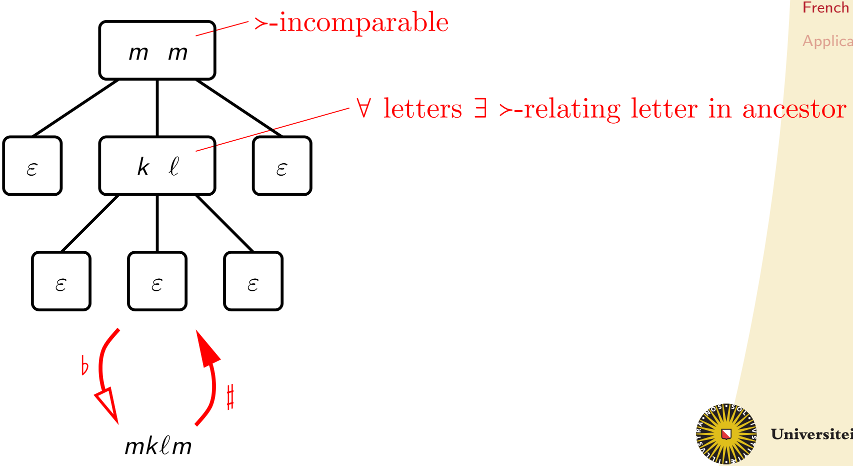



Universiteit Utrecht

# Involutive monoid on French terms $L^\sharp$

## Definition

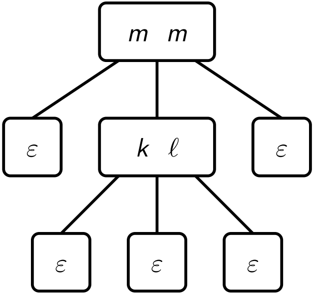
terms on strings on $\succ$-ordered letters where $\flat \circ \sharp$ identity

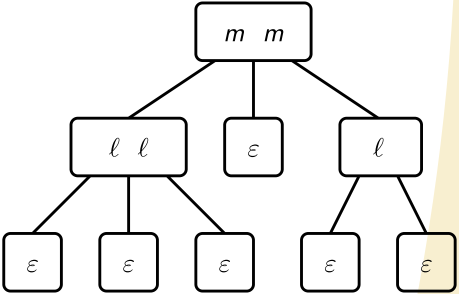


# Involutive monoid on French terms $L^\sharp$

## Definition

terms on strings on $\succ$-ordered letters where $\flat \circ \sharp$ identity



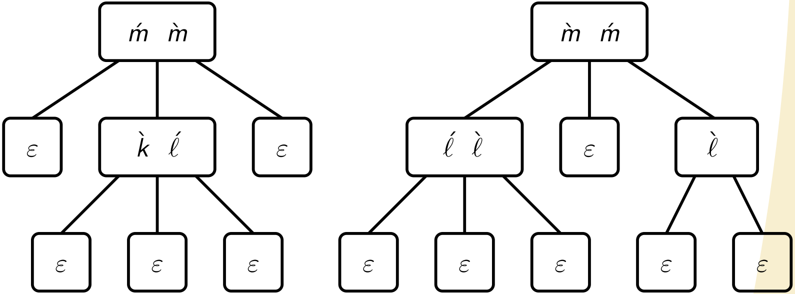




$mk\ell m$ $\qquad$ $\ell\ell mm\ell$

# Involutive monoid on French terms $L^\sharp$

## Definition

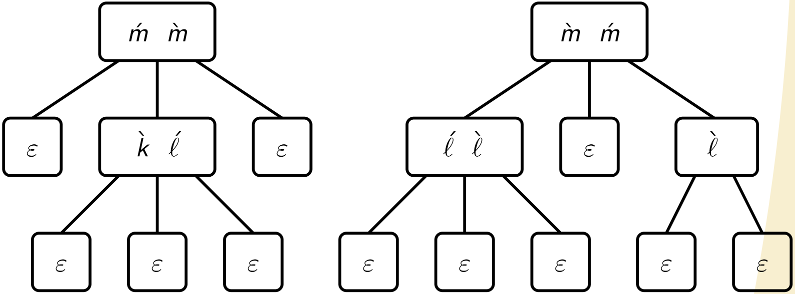
terms on French strings on $\succ$-ordered letters where $\flat \circ \sharp$ identity
operations on $L^\sharp$ defined via $\widehat{L}$, e.g. $t \cdot u = (t^\flat u^\flat)^\sharp$




$\acute{m}\grave{k}\acute{\ell}\grave{m}$ $\acute{\ell}\grave{\ell}\,\grave{m}\acute{m}\grave{\ell}$

# A well-founded order on French terms

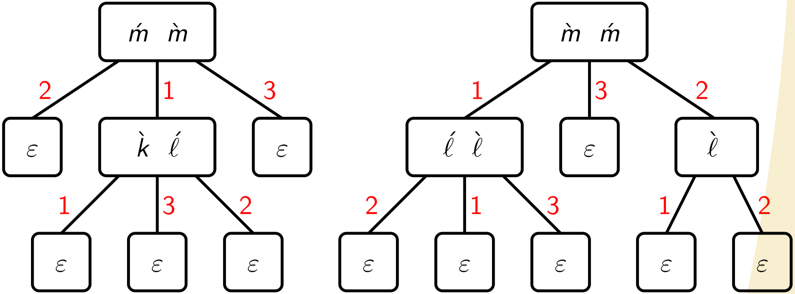
- (iterative) lexicographic path order based on $>$




$\acute{m}\grave{k}\acute{\ell}\grave{m}$ $\qquad$ $\acute{\ell}\grave{\ell}\,\grave{m}\acute{m}\grave{\ell}$

# A well-founded order on French terms

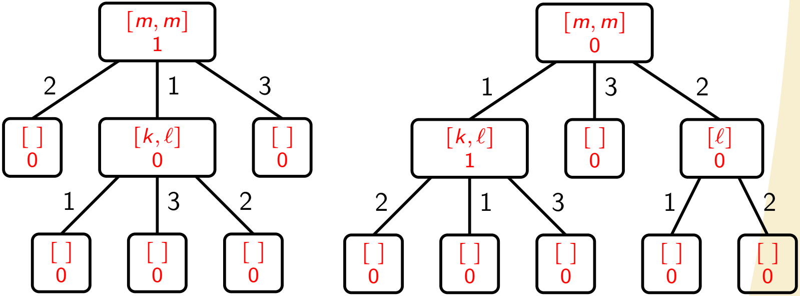
- (iterative) lexicographic path order based on $>$
- lexicographic order on argument places compatible with marks




$\acute{m}\grave{k}\acute{\ell}\grave{m}$ $\acute{\ell}\grave{\ell}\ \grave{m}\acute{m}\grave{\ell}$

# A well-founded order on French terms

- (iterative) lexicographic path order based on $>$
- lexicographic order on argument places compatible with marks
- signature ordered by $\succ = \binom{\succ_{mul}}{>}$ via $\binom{\text{multiset}}{\text{area}}$

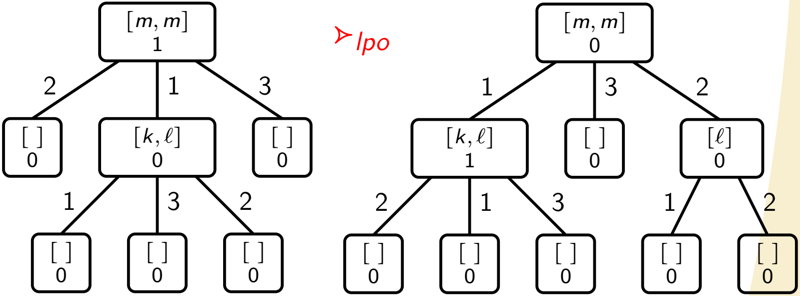



$\acute{m}\grave{k}\acute{\ell}\grave{m}$ $\acute{\ell}\grave{\ell}\,\grave{m}\acute{m}\grave{\ell}$

# A well-founded order on French strings/terms

- (iterative) lexicographic path order based on $\succ$
- lexicographic order on argument places compatible with marks
- signature ordered by $\succ = \binom{\succ_{mul}}{>}$ via $\binom{\text{multiset}}{\text{area}}$






$\acute{m}\grave{k}\acute{\ell}\grave{m} \succ_{lpo} \acute{\ell}\grave{\ell}\,\grave{m}\acute{m}\grave{\ell}$

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms
- $\succ_{lpo}$ not an ordered monoid: $\grave{k}\grave{\ell} \succ_{lpo} \acute{\ell}$ but $\grave{k}\grave{\ell}\grave{\ell} \not\succ_{lpo} \acute{\ell}\grave{\ell}$

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms
- $\succ_{lpo}$ not an ordered monoid
- $s\hat{\ell}r \succ_{lpo} s\{\ell\gt\}r$ (in EBNF { } is arbitrary repetition)

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms
- $\succ_{lpo}$ not an ordered monoid
- $s\hat{\ell}r \succ_{lpo} s\{\ell>\}r$

Proof.

induction on length $sr$, cases whether $\ell$ is $\succ$-maximal in $s\hat{\ell}r$

yes decrease in multiset of head

no induction on substring/term $\hat{\ell}$ is in □

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms
- $\succ_{lpo}$ not an ordered monoid
- $s\hat{\ell}r \succ_{lpo} s\{\ell\succ\}r$

Proof.
induction on length $sr$, cases whether $\ell$ is $\succ$-maximal in $s\hat{\ell}r$

yes decrease in multiset of head
no induction on substring/term $\hat{\ell}$ is in $\square$

- $s\acute{\ell}\grave{m}r \succ_{lpo} s\{\ell\succ\}[\grave{m}]\{\ell, m\succ\}[\acute{\ell}]\{m\succ\}r$ ([ ] is option)

# Properties of $\succ_{lpo}$

- head of term $\succ$-related to heads of all subterms
- $\succ_{lpo}$ not an ordered monoid
- $s\hat{\ell}r \succ_{lpo} s\{\ell\succ\}r$

Proof.
induction on length $sr$, cases whether $\ell$ is $\succ$-maximal in $s\hat{\ell}r$

yes decrease in multiset of head
no induction on substring/term $\hat{\ell}$ is in $\square$

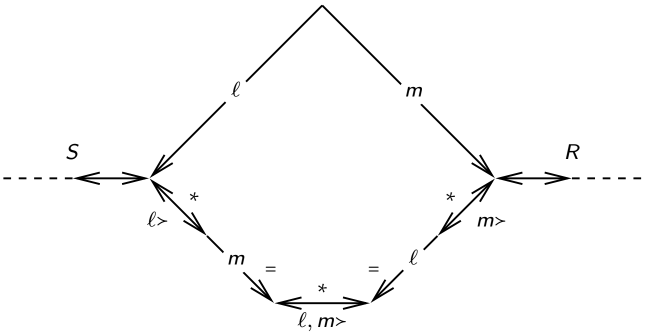
- $s\acute{\ell}\grave{m}r \succ_{lpo} s\{\ell\succ\}[\grave{m}]\{\ell, m\succ\}[\acute{\ell}]\{m\succ\}r$

Proof.
induction on length $sr$, cases whether $\ell, m$ are $\succ$-maximal in $s\acute{\ell}\grave{m}r$

both decrease in area of head
$\acute{\ell}$ decrease in the substring/term to the right of $\acute{\ell}$
$\grave{m}$ decrease in the substring/term to the left of $\grave{m}$
neither induction on substring/term $\acute{\ell}\grave{m}$ is in $\square$

# Filling in locally decreasing diagram decreases

## Theorem

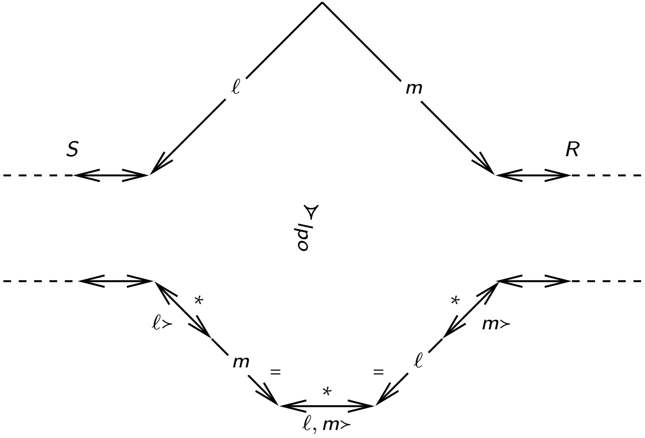





Universiteit Utrecht

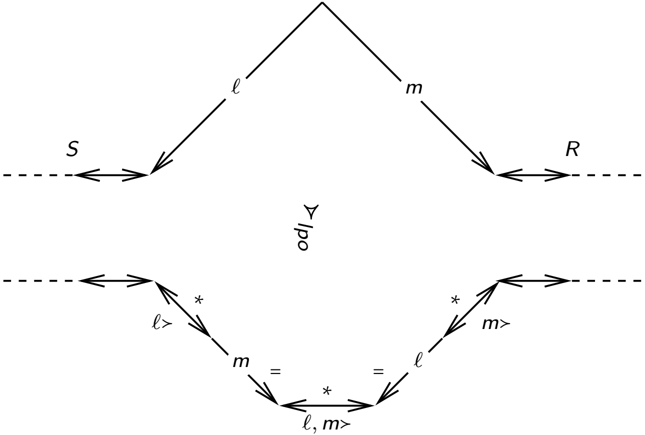
# Filling in locally decreasing diagram decreases

## Theorem



# Filling in locally decreasing diagram decreases

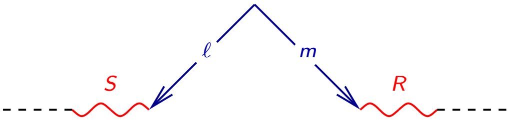
## Theorem

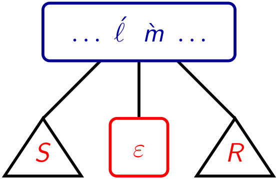



## Proof.

$s\acute{\ell}\grave{m}r \succ_{lpo} s\{\ell\succ\}[\grave{m}]\{\ell, m\succ\}[\acute{\ell}]\{m\succ\}r$

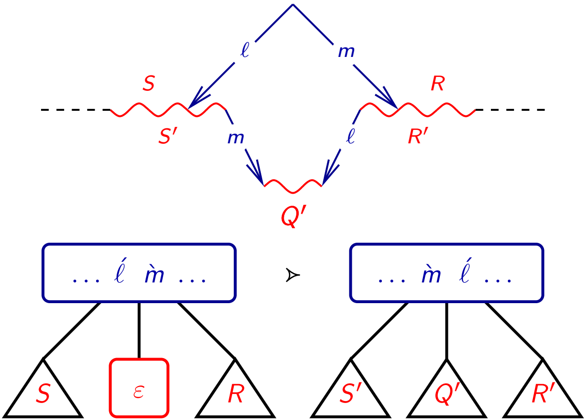
# Idea: $\succ$-maximal steps modulo non-$\succ$-maximal steps




case 1: local confluence peak of $\succ$-maximal steps

# Idea: $\succ$-maximal steps modulo non-$\succ$-maximal steps

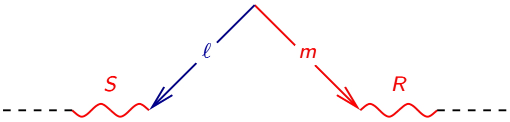


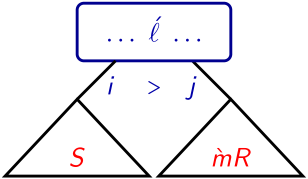

area decrease



Universiteit Utrecht

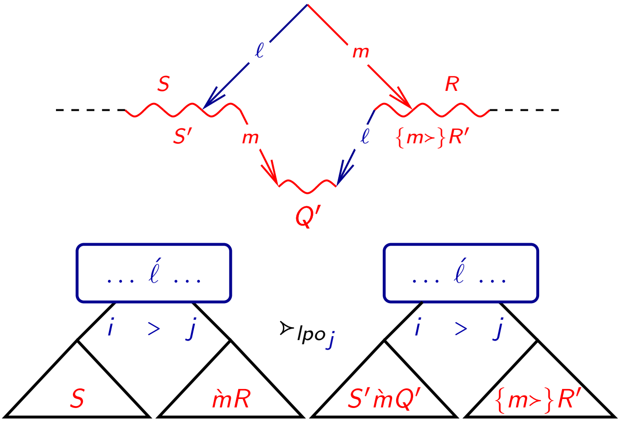
# Idea: ≻-maximal steps modulo non-≻-maximal steps



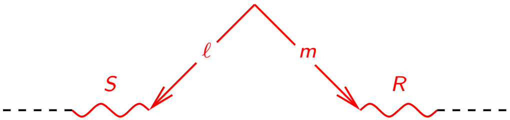

case 2: local coherence peak of ≻-maximal and non-≻-maximal step

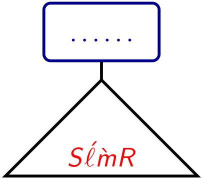
# Idea: $\succ$-maximal steps modulo non-$\succ$-maximal steps




decrease in $j$th argument, lexicographically before $i$th

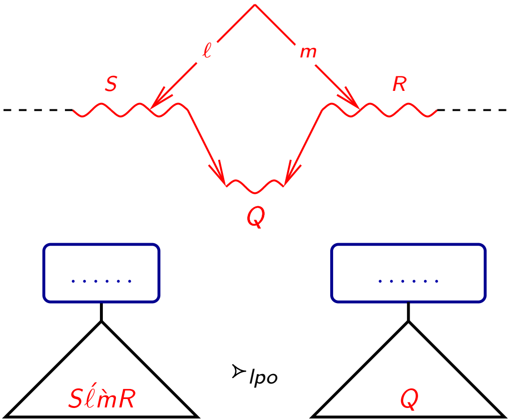
# Idea: ≻-maximal steps modulo non-≻-maximal steps



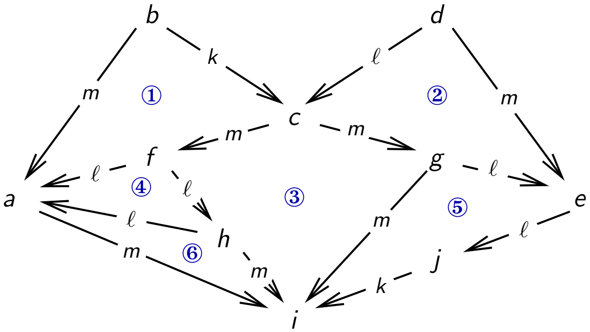

case 3: local modulo peak of non-≻-maximal steps

# Idea: $\succ$-maximal steps modulo non-$\succ$-maximal steps



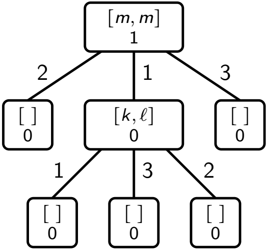

decrease in argument both steps are in

# $\succ_{lpo}$ at work

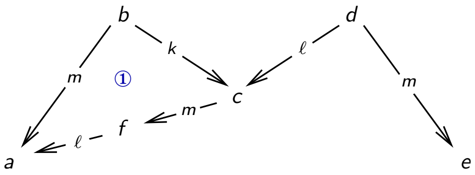


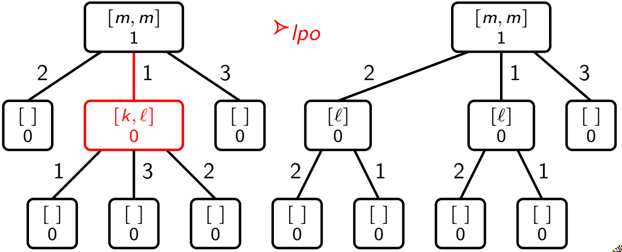
# Filling in local diagrams ①

$$\begin{tikzcd}[column sep=small]
 & b \arrow[dl, "m"'] \arrow[dr, "k"] & & d \arrow[dl, "\ell"'] \arrow[dr, "m"] & \\
a & & c & & e
\end{tikzcd}$$

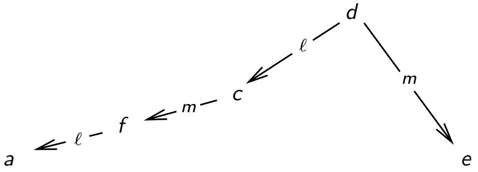


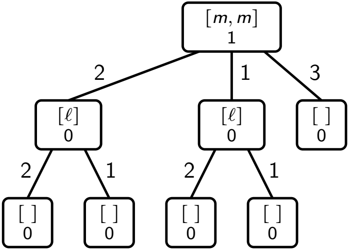
# Filling in local diagrams ①



# Filling in local diagrams ②

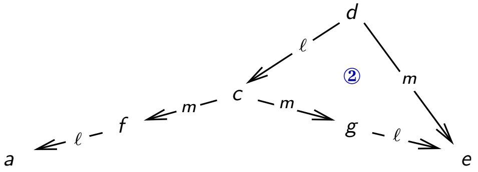


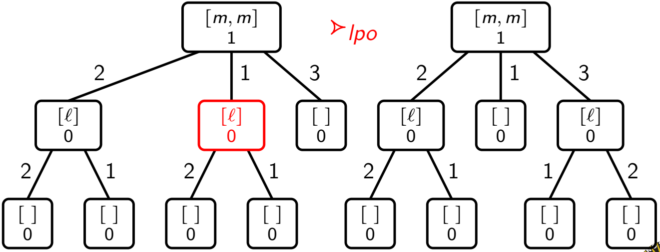






Universiteit Utrecht

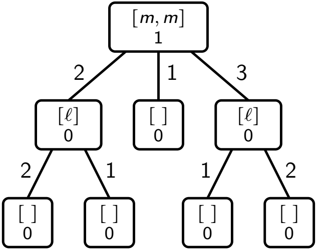
# Filling in local diagrams ②



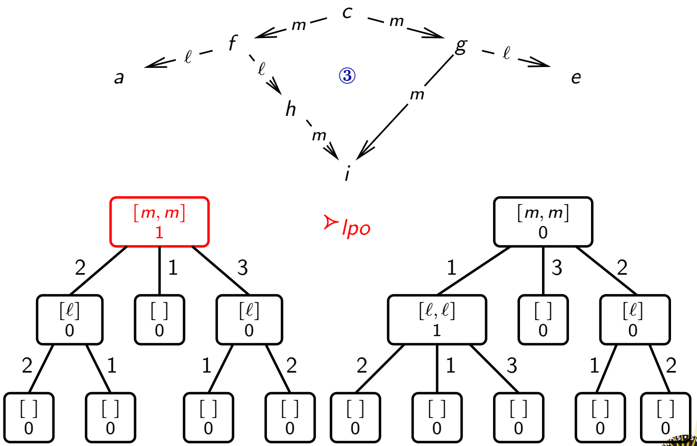



# Filling in local diagrams ③

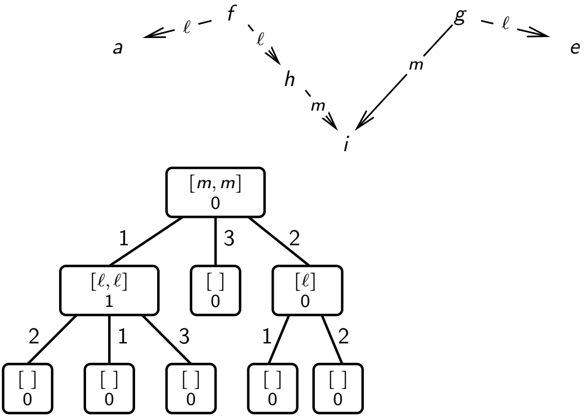
$$a \xleftarrow{\ell} f \xleftarrow{m} c \xrightarrow{m} g \xrightarrow{\ell} e$$



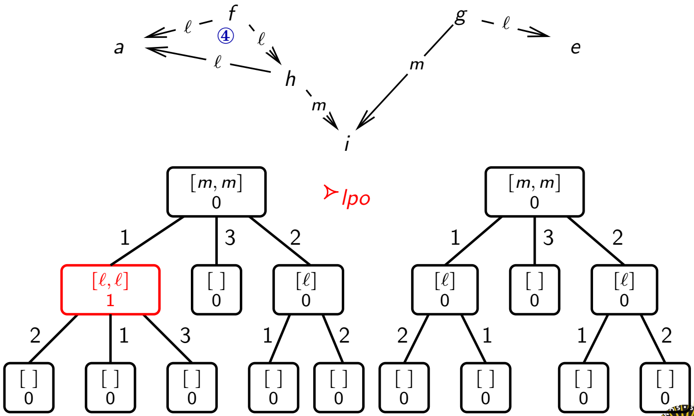
# Filling in local diagrams ③



# Filling in local diagrams ④



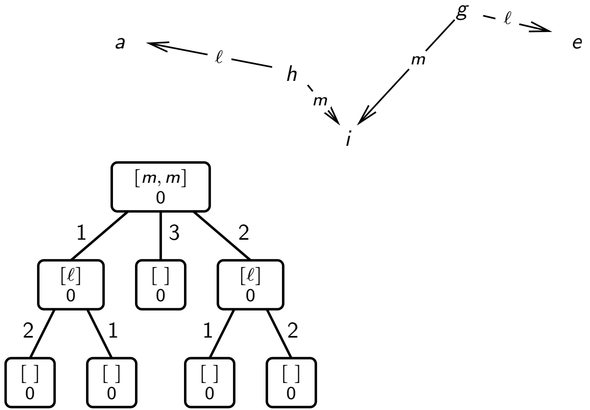
# Filling in local diagrams ④



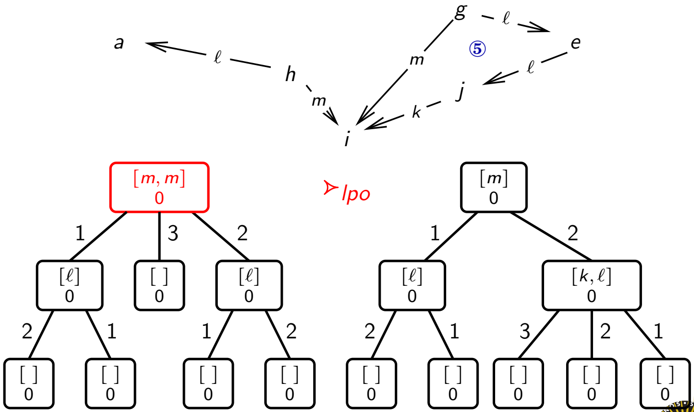



Universiteit Utrecht

# Filling in local diagrams ⑤



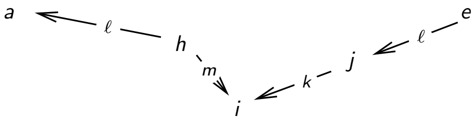
# Filling in local diagrams ⑤

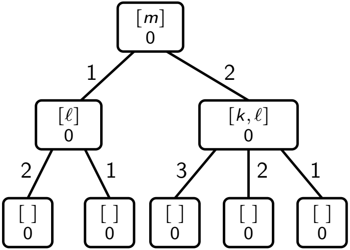




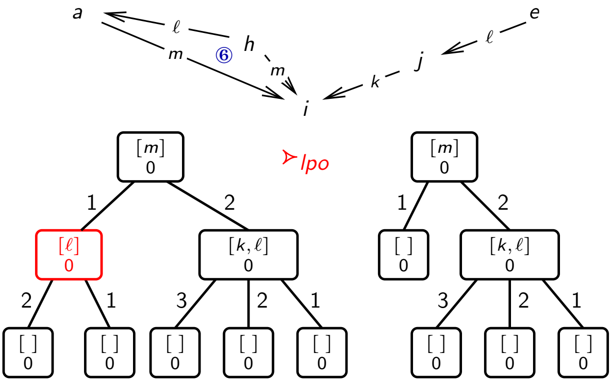

Universiteit Utrecht

# Filling in local diagrams ⑥

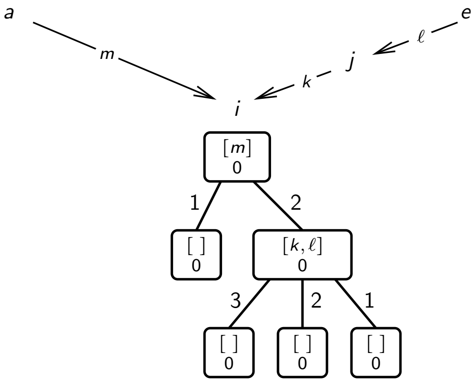


# Filling in local diagrams ⑥



# Filling in local diagrams ⑥



# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)

# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)
- decidable: by universal quantification over orders extending ($s$ bigger than $r$ if $\forall$ well-orders extending $\succ$, they are related)

# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)
- decidable: by universal quantification over orders extending ($s$ bigger than $r$ if $\forall$ well-orders extending $\succ$, they are related)
- decreasing diagrams modulo: involutive letters $\dot{\ell}$, i.e. $\dot{\ell}^{-1} = \dot{\ell}$

# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)
- decidable: by universal quantification over orders extending ($s$ bigger than $r$ if $\forall$ well-orders extending $\succ$, they are related)
- involutive rewriting ($\varrho : s \to r$ converse of $\varrho^{-1} : s^{-1} \to r^{-1}$)

# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)
- decidable: by universal quantification over orders extending ($s$ bigger than $r$ if $\forall$ well-orders extending $\succ$, they are related)
- involutive rewriting ($\varrho : s \to r$ converse of $\varrho^{-1} : s^{-1} \to r^{-1}$)
- covers all confluence modulo results in Ohlebusch (either by the previous item, or ordering modulo steps below other steps)

# Flexibility

Adaptations:

- monotonic: by universal quantification over contexts ($s$ bigger than $r$ if $\forall q_1, q_2,\ q_1 s q_2 \succ_{lpo} q_1 r q_2$)
- decidable: by universal quantification over orders extending ($s$ bigger than $r$ if $\forall$ well-orders extending $\succ$, they are related)
- involutive rewriting ($\varrho : s \to r$ converse of $\varrho^{-1} : s^{-1} \to r^{-1}$)
- covers all confluence modulo results in Ohlebusch (either by the previous item, or ordering modulo steps below other steps)
- application to factorisation theorems (factorisation is commutation with the inverse, RTA 2012, Beniamino Accattoli)

# Conclusion

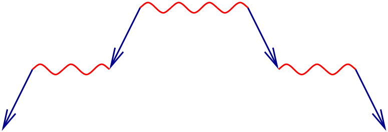
- alternative correctness proof of decreasing diagrams (De Bruijn,vO,Klop,de Vrijer,Bezem,Jouannaud)

# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps

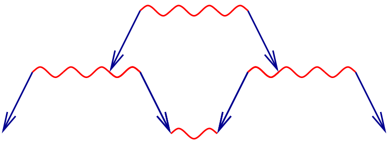
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



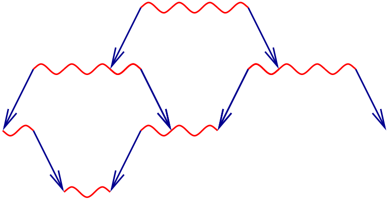
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



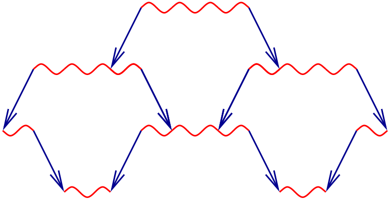
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



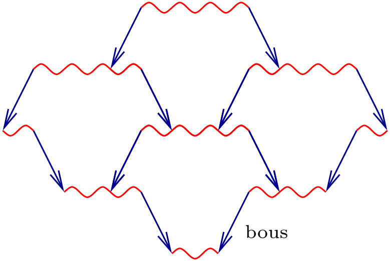
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps

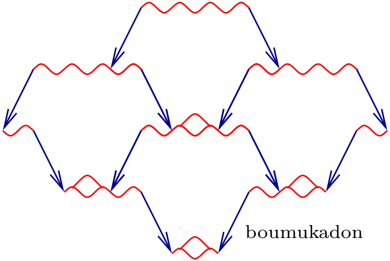





Universiteit Utrecht

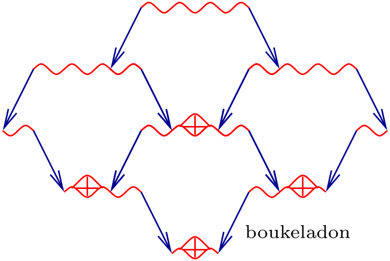
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps

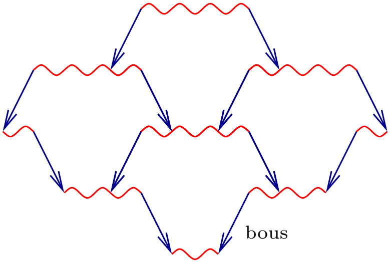




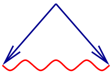

Universiteit Utrecht

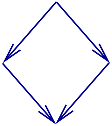
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps



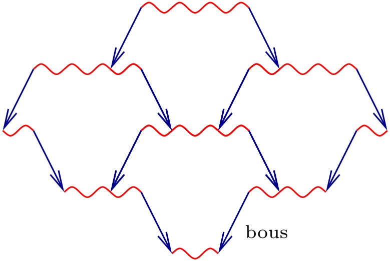

- Newman's Lemma (multiset)+Lemma of Hindley–Rosen (area)



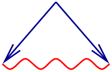


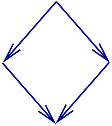
# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps




- Newman's Lemma+Lemma of Hindley–Rosen



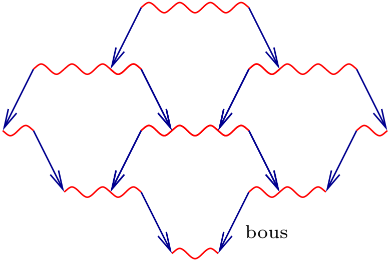




Universiteit Utrecht

# Conclusion

- alternative correctness proof of decreasing diagrams
- confluence of $\succ$-maximal steps modulo non-$\succ$-maximal steps




- Newman's Lemma+Lemma of Hindley–Rosen

- flexible



Universiteit Utrecht

# *Het kind en ik*

*Ik zou een dag uit vissen,*
*ik voelde mij moedeloos.*
*Ik maakte tussen de lissen*
*met de hand een wak in het kroos.*

*Er steeg licht op van beneden*
*uit de zwarte spiegelgrond.*
*Ik zag een tuin onbetreden*
*en een kind dat daar stond.*

*Het stond aan zijn schrijftafel*
*te schrijven op een lei.*
*Het woord onder de griffel*
*herkende ik, was van mij.*

*Maar toen heeft het geschreven,*
*zonder haast en zonder schroom,*
*al wat ik van mijn leven*
*nog ooit te schrijven droom.*

*En telkens als ik even*
*knikte dat ik het wist,*
*liet hij het water beven*
*en het werd uitgewist.*

Het kind en ik

Ik zou een dag uit vissen,
ik voelde mij moedeloos.
Ik maakte tussen de lissen
met de hand een wak in het kroos.

Er steeg licht op van beneden
uit de zwarte spiegelgrond.
Ik zag een tuin onbetreden
en een kind dat daar stond.

Het stond aan zijn schrijftafel
te schrijven op een lei.
Het woord onder de griffel
herkende ik, was van mij.

Maar toen heeft het geschreven,
zonder haast en zonder schroom,
al wat ik van mijn leven
nog ooit te schrijven droom.

En telkens als ik even
knikte dat ik het wist,
liet hij het water beven
en het werd uitgewist.