On the Construction of *Śivasūtra*-Alphabets

Wiebke Petersen

Institute of Language and Information
University of Düsseldorf, Germany
petersew@uni-duesseldorf.de

3rd Int. Sanskrit Computational Linguistics Symposium, 15.-17. January 2009

अइउण्। ऋऌक्। एओङ्। ऐऔच्। हयवरट्।
लण्। ञमङणनम्। झभञ्। घढधष्। जबगडदश्।
खफछठथचटतव्। कपय्। शषसर्। हल्।

Phonological Rules

modern notation

A is replaced by B if preceded by C and succeeded by D.

$$A \rightarrow B / C\_\_D$$

example: final devoicing

$$\begin{bmatrix} + & \text{consonantal} \\ - & \text{nasal} \\ + & \text{voiced} \end{bmatrix} \rightarrow \begin{bmatrix} + & \text{consonantal} \\ - & \text{nasal} \\ - & \text{voiced} \end{bmatrix} / \_\_\sharp$$

Phonological Rules

modern notation

A is replaced by B if preceded by C and succeeded by D.

$$A \rightarrow B / C\_\_D$$

example: final devoicing

$$\begin{bmatrix} + & \text{consonantal} \\ - & \text{nasal} \\ + & \text{voiced} \end{bmatrix} \rightarrow \begin{bmatrix} + & \text{consonantal} \\ - & \text{nasal} \\ - & \text{voiced} \end{bmatrix} / \_\_\sharp$$

Phonological Rules

modern notation

A is replaced by B if preceded by C and succeeded by D.

$$A \rightarrow B / C\_D$$

Pāṇini's linear Coding

$A +$ genitive, $B +$ nominative, $C +$ ablative, $D +$ locative.

example

- *sūtra* 6.1.77: *iko yaṇaci* (इको यणचि)
- analysis: $[\text{ik}]_{\text{gen}}[\text{yaṇ}]_{\text{nom}}[\text{ac}]_{\text{loc}}$
- modern notation: [iK] → [yṆ]/_[aC]

Phonological Rules

modern notation

A is replaced by B if preceded by C and succeeded by D.

$$A \rightarrow B / C\_\_D$$

Pāṇini's linear Coding

A + genitive, B + nominative, C + ablative, D + locative.

example

- *sūtra* 6.1.77: *iko yaṇaci* (इको यणचि)
- analysis: $[\text{ik}]_{\text{gen}}[\text{yaṇ}]_{\text{nom}}[\text{ac}]_{\text{loc}}$
- modern notation: [iK] → [yṆ]/_ [aC]

Pāṇini faced the problem of giving a linear representation of the nonlinear system of sound classes.

A similar problem occurs in . . .

Libraries

Warehouses and stores

Pāṇini's solution: *Śivasūtras*

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ
6.					l	Ṇ
7.	ñ	m	ṅ	ṇ	n	M
8.	jh	bh				Ñ
9.			gh	ḍh	dh	Ṣ
10.	j	b	g	ḍ	d	Ś
11.	kh	ph	ch	ṭh	th	
			c	ṭ	t	V
12.	k	p				Y
13.		ś	ṣ	s		R
14.	h					L

अइउण्। ऋऌक्।

a·i·uṇ| ṛ·ḷk|

एओङ्। ऐऔच्।

e·oṅ| ai·auc|

हयवरट्। लण्।

hayavaraṭ| laṇ|

ञमङणनम्। झभञ्।

ñamaṅaṇanam| jhabhañ|

घढधष्। जबगडदश्।

ghaḍhadhaṣ| jabagaḍadaś|

खफछठथचटतव्।

khaphachaṭhathacaṭatav|

कपय्। शषसर्। हल्।

kapay| śaṣasar| hal|

Pāṇini's solution: *Śivasūtras*

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ
6.					l	Ṇ
7.	ñ	m	ṅ	ṇ	n	M
8.	jh	bh				Ñ
9.			gh	ḍh	dh	Ṣ
10.	j	b	g	ḍ	d	Ś
11.	kh	ph	ch	ṭh	th	
			c	ṭ	t	V
12.	k	p				Y
13.		ś	ṣ	s		R
14.	h					L

अइउण्। ऋऌक्।

a·i·uṇ| ṛ·ḷk|

एओङ्। ऐऔच्।

e·oṅ| ai·auc|

हयवरट्। लण्।

hayavaraṭ| laṇ|

ञमङणनम्। झभञ्।

ñamaṅaṇanam| jhabhañ|

घढधष्। जबगडदश्।

ghaḍhadhaṣ| jabagaḍadaś|

खफछठथचटतव्।

khaphachaṭhathacaṭatav|

कपय्। शषसर्। हल्।

kapay| śaṣasar| hal|

Pāṇini's solution: *Śivasūtras*

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ
6.					l	Ṇ
7.	ñ	m	ṅ	ṇ	n	M
8.	jh	bh				Ñ
9.			gh	ḍh	dh	Ṣ
10.	j	b	g	ḍ	d	Ś
11.	kh	ph	ch	ṭh	th	
			c	ṭ	t	V
12.	k	p				Y
13.		ś	ṣ	s		R
14.	h					L

anubandha

अइउण्। ऋऌक्।
a·i·uṇ| ṛ·ḷk|

एओङ्। ऐऔच्।
e·oṅ| ai·auc|

हयवरट्। लण्।
hayavaraṭ| laṇ|

ञमङणनम्। झभञ्।
ñamaṅaṇanam| jhabhañ|

घढधष्। जबगडदश्।
ghaḍhadhaṣ| jabagaḍadaś|

खफछठथचटतव्।
khaphachaṭhathacaṭatav|

कपय्। शषसर्। हल्।
kapay| śaṣasar| hal|

Pratyāhāras

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ

Pratyāhāras

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ

iK

Pratyāhāras

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ

iK= ⟨i, u, ṛ, ḷ⟩

Analysis of iko yaṇaci: [iK] → [yṆ]/_ [aC]

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ
6.					l	Ṇ

- [iK] → [yṆ]/_ [aC]
- ⟨i, u, ṛ, ḷ⟩ → ⟨y, v, r, l⟩/_ ⟨a, i, u, ṛ, ḷ, e, o, ai, au⟩

Analysis of iko yaṇaci: [iK] → [yṆ]/_ [aC]

1.	a	i	u			Ṇ
2.				ṛ	ḷ	K
3.		e	o			Ṅ
4.		ai	au			C
5.	h	y	v	r		Ṭ
6.					l	Ṇ

- [iK] → [yṆ]/_ [aC]
- ⟨i, u, ṛ, ḷ⟩ → ⟨y, v, r, l⟩/_ ⟨a, i, u, ṛ, ḷ, e, o, ai, au⟩

General problem of S-sortability

Given a set of classes, order the elements of the classes (without duplications) in a linear order (in a list) such that each single class forms a continuous interval with respect to that order.

- The target orders are called S-orders
- A set of classes is S-sortable if it has an S-order

Note that every S-order becomes a *Śivasūtra*-alphabet (S-alphabet) by adding a marker (*anubandha*) behind each element.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.
$a\ b\ c\ d\ e$ or $e\ d\ c\ b\ a$

Examples

S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$ is S-sortable;
one of its S-orders is
$a\ b\ c\ g\ h\ f\ i\ d\ e$

non-S-sortable example

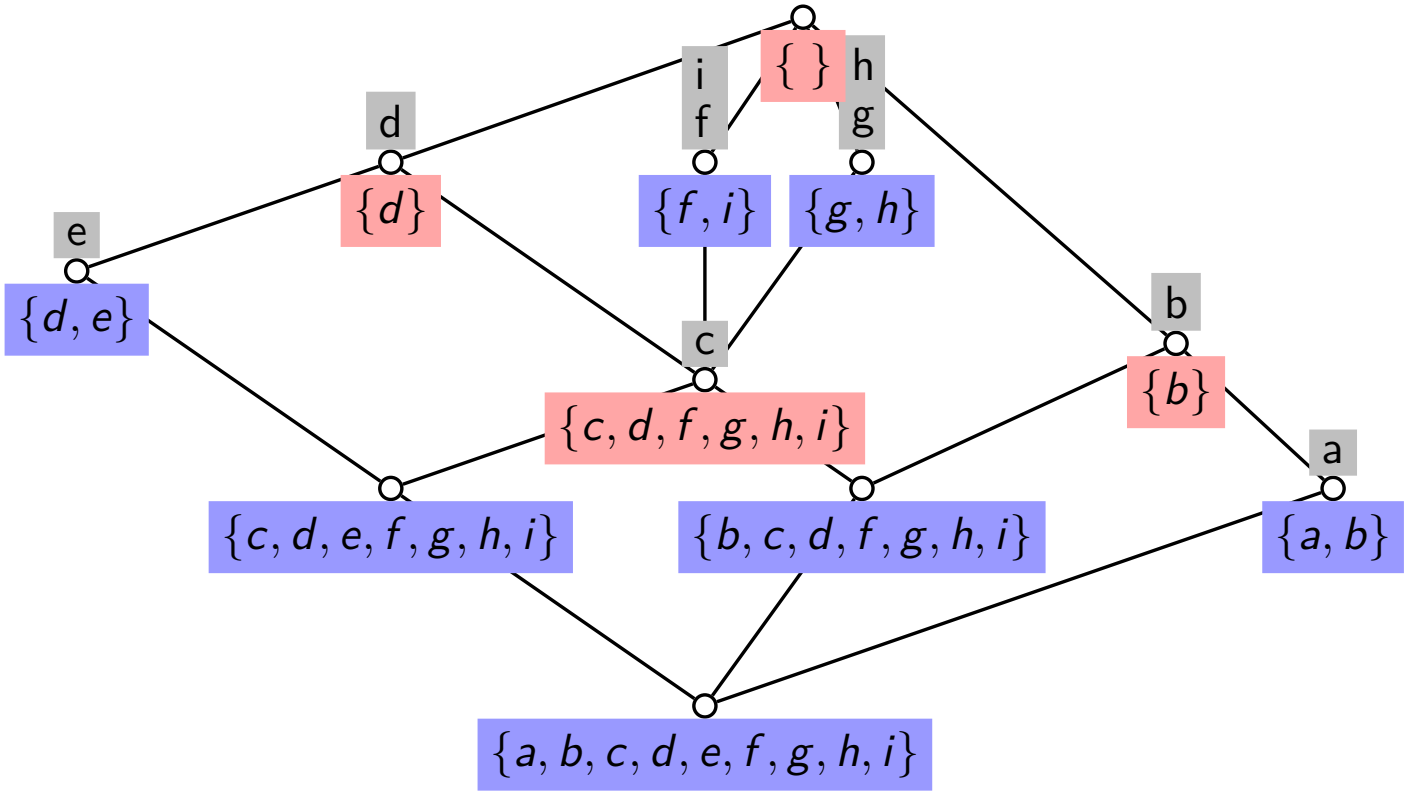
The set of classes:
$\{\{a, b\}, \{b, c\}, \{a, c\}\}$ is not S-sortable.

non-S-sortable example

The set of classes:
$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$ is not S-sortable.
$a\ b\ c\ d\ e$ or $e\ d\ c\ b\ a$

Visualize relations

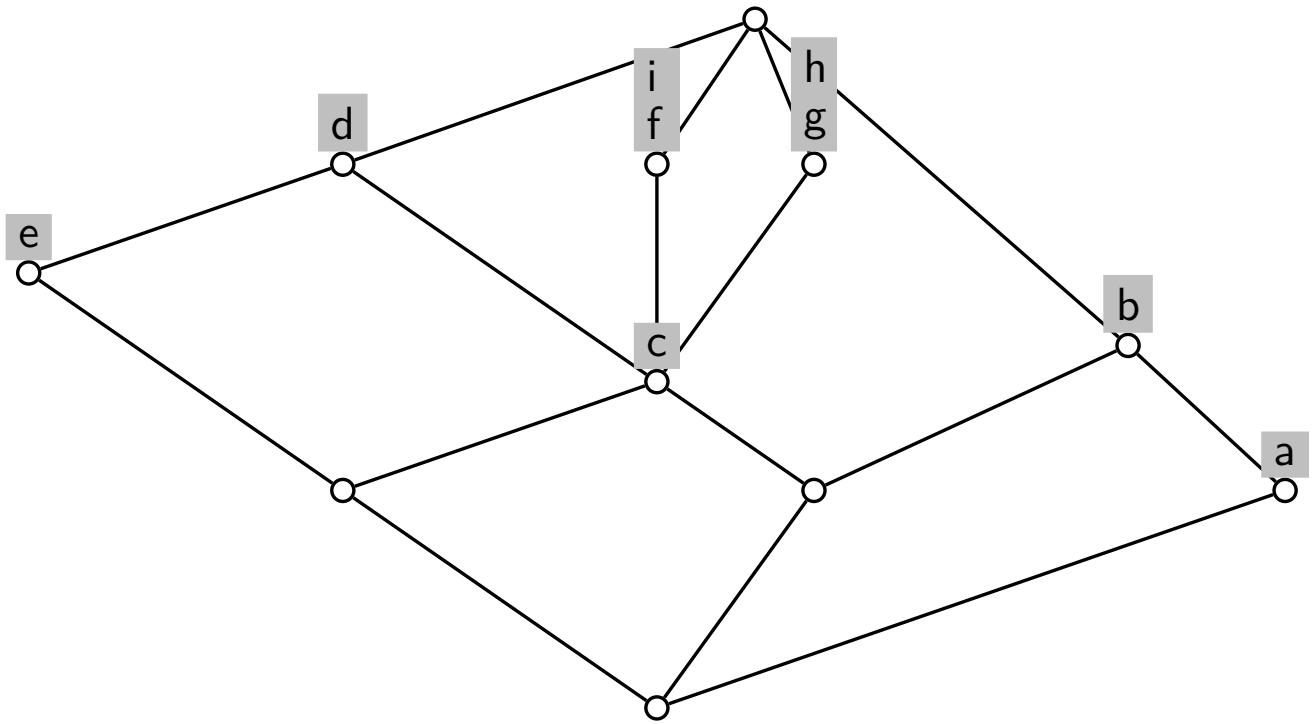
$$\{\{d,e\},\{a,b\},\{b,c,d,f,g,h,i\},\{f,i\},\{c,d,e,f,g,h,i\},\{g,h\}\}$$



concept lattice

Visualize relations

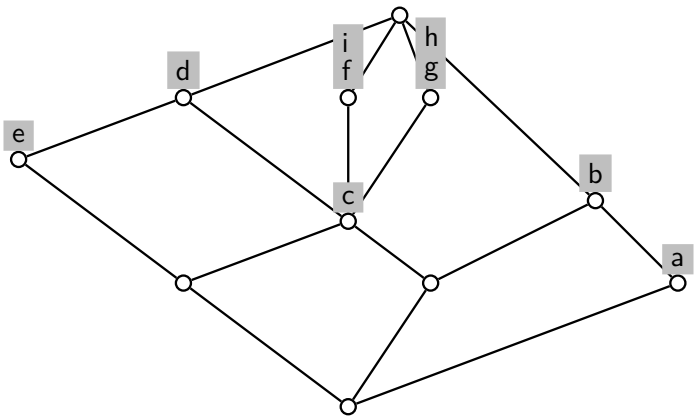
$$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$$



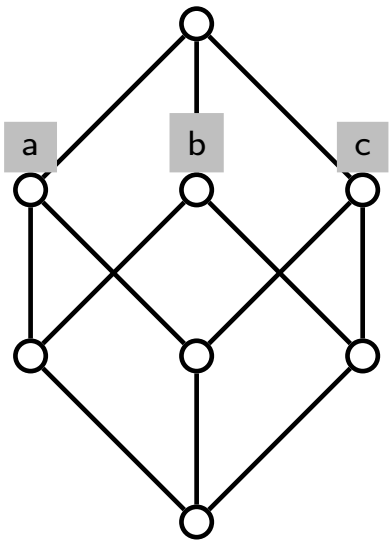
concept lattice

Visualize relations

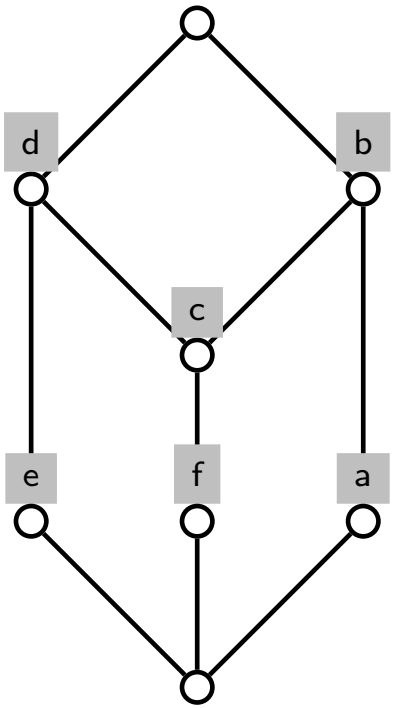
$\{\{d, e\}, \{a, b\}, \{b, c, d, f, g, h, i\}, \{f, i\}, \{c, d, e, f, g, h, i\}, \{g, h\}\}$



$\{\{a, b\}, \{b, c\}, \{a, c\}\}$



$\{\{d, e\}, \{a, b\}, \{b, c, d\}, \{b, c, d, f\}\}$

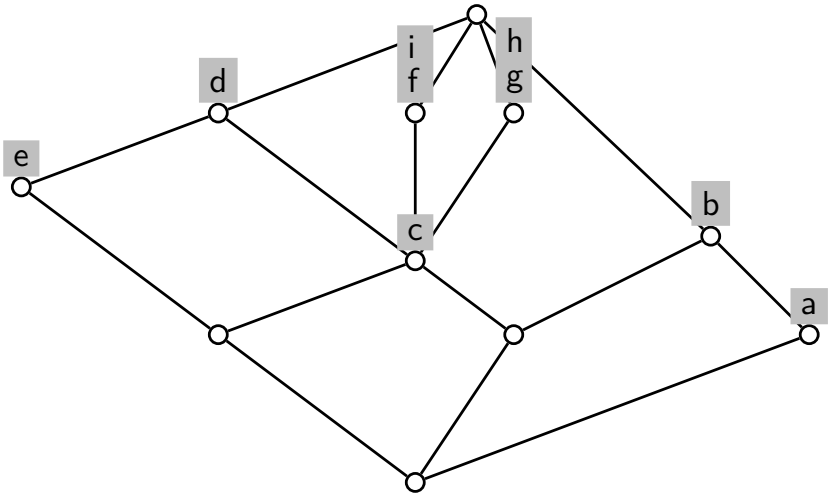


Main theorem of S-sortability

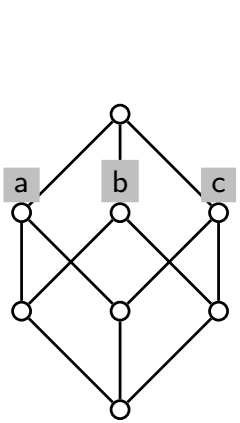
A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

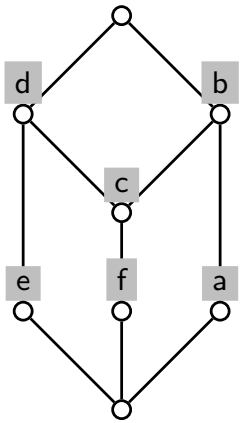
1. Its concept lattice is Hasse-planar and for any element a there is a node labeled a in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

Example: S-sortable



Examples: not S-sortable



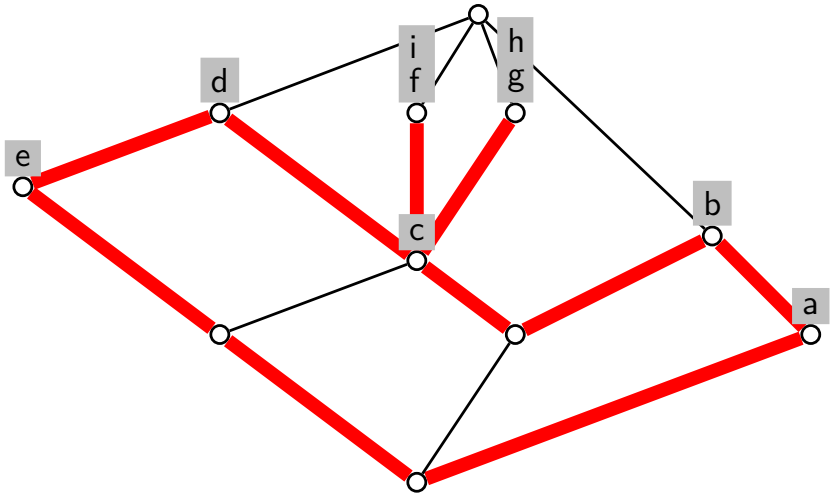


Main theorem of S-sortability

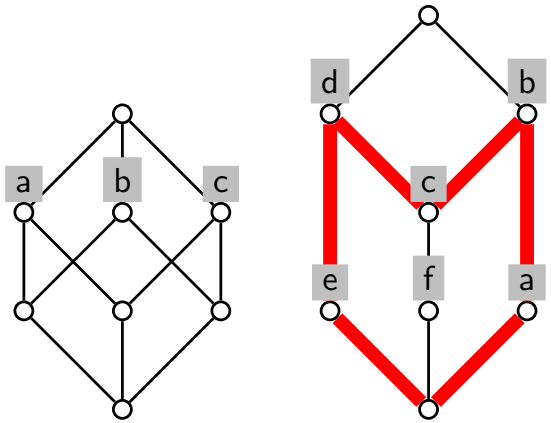
A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

1. Its concept lattice is Hasse-planar and for any element *a* there is a node labeled *a* in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

Example: S-sortable



Examples: not S-sortable

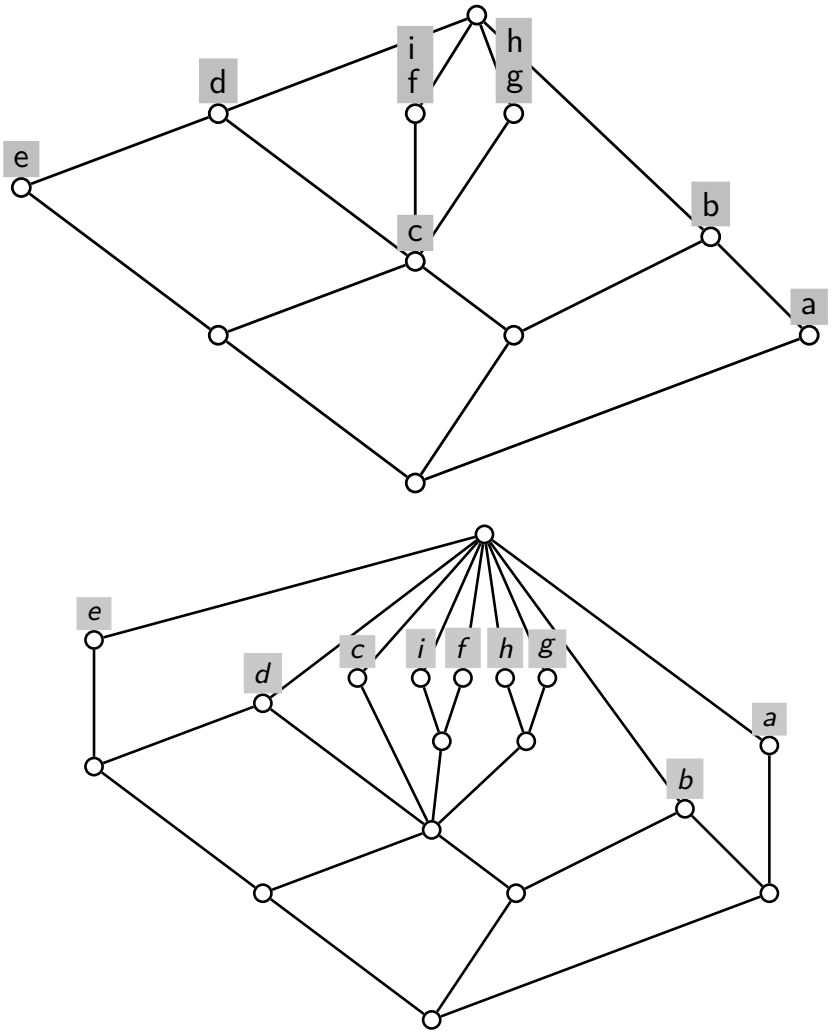


Main theorem of S-sortability

A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

1. Its concept lattice is Hasse-planar and for any element *a* there is a node labeled *a* in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

Example: S-sortable

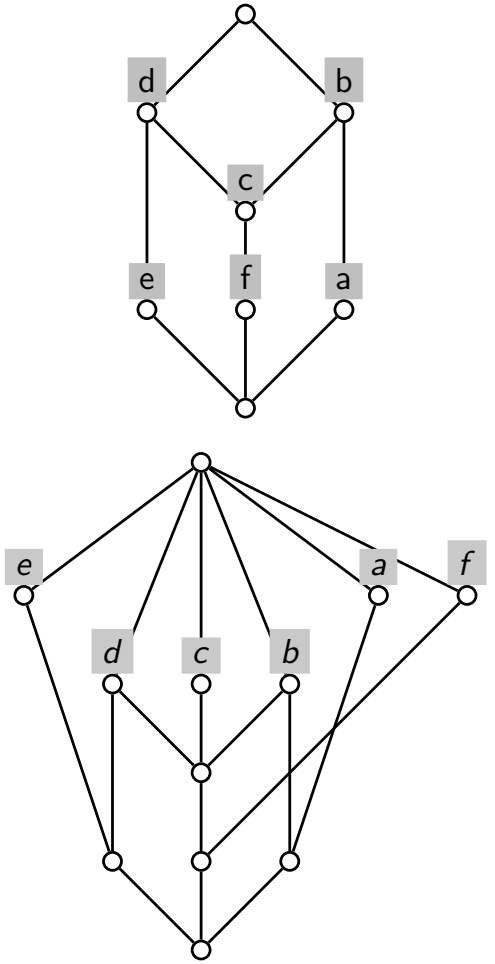


Main theorem of S-sortability

A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

1. Its concept lattice is Hasse-planar and for any element a there is a node labeled a in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

Example: not S-sortable



Main theorem of S-sortability

A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

1. Its concept lattice is Hasse-planar and for any element a there is a node labeled a in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

This condition can be checked algorithmically.

Main theorem of S-sortability

A set of classes is S-sortable without duplications if one of the following equivalent statements is true:

1. Its concept lattice is Hasse-planar and for any element a there is a node labeled a in the S-graph.
2. The concept lattice of the enlarged set of classes is Hasse-planar.
3. The Ferrers-graph of the enlarged set of classes is bipartite.

Getting back to Pāṇini's problem

$a{\cdot}i{\cdot}u\dot{n}|\ \dot{r}{\cdot}\dot{l}k|\ e{\cdot}o\dot{n}|\ ai{\cdot}auc|\ hayavara\dot{t}|$
$la\dot{n}|\ \tilde{n}ama\dot{n}a\dot{n}anam|\ jhabha\tilde{n}|\ ghad\dot{h}adhas\dot{h}|\ jabagad\dot{a}das\acute{s}|$
$khaphacha\dot{t}hathacata\dot{t}atav|\ kapay|\ \acute{s}as\dot{a}sar|\ hal|$

Q: Are the *Śivasūtras* minimal (with respect to length)?

What does minimal mean?

$a{\cdot}i{\cdot}u\dot{n}|\ \dot{r}{\cdot}\dot{l}k|\ e{\cdot}o\dot{n}|\ ai{\cdot}auc|\ hayavara\dot{t}|$
$la\dot{n}|\ \tilde{n}ama\dot{n}a\dot{n}anam|\ jhabha\tilde{n}|\ ghad\dot{h}adhas\dot{s}|\ jabagad\dot{a}das\acute{s}|$
$khaphacha\dot{t}hathaca\dot{t}atav|\ kapay|\ \acute{s}a\dot{s}asar|\ hal|$

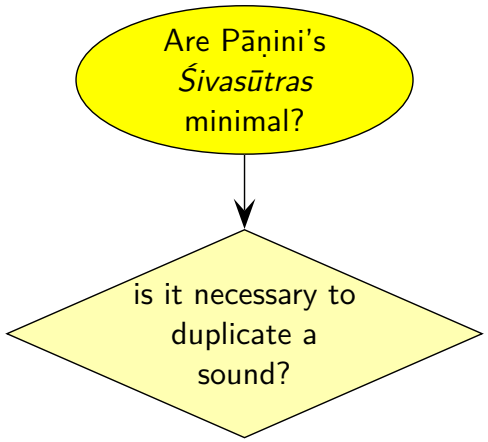
The *Śivasūtras* are minimal if it is **im**possible rearrange the Sanskrit sounds in a new list with *anubandhas* such that

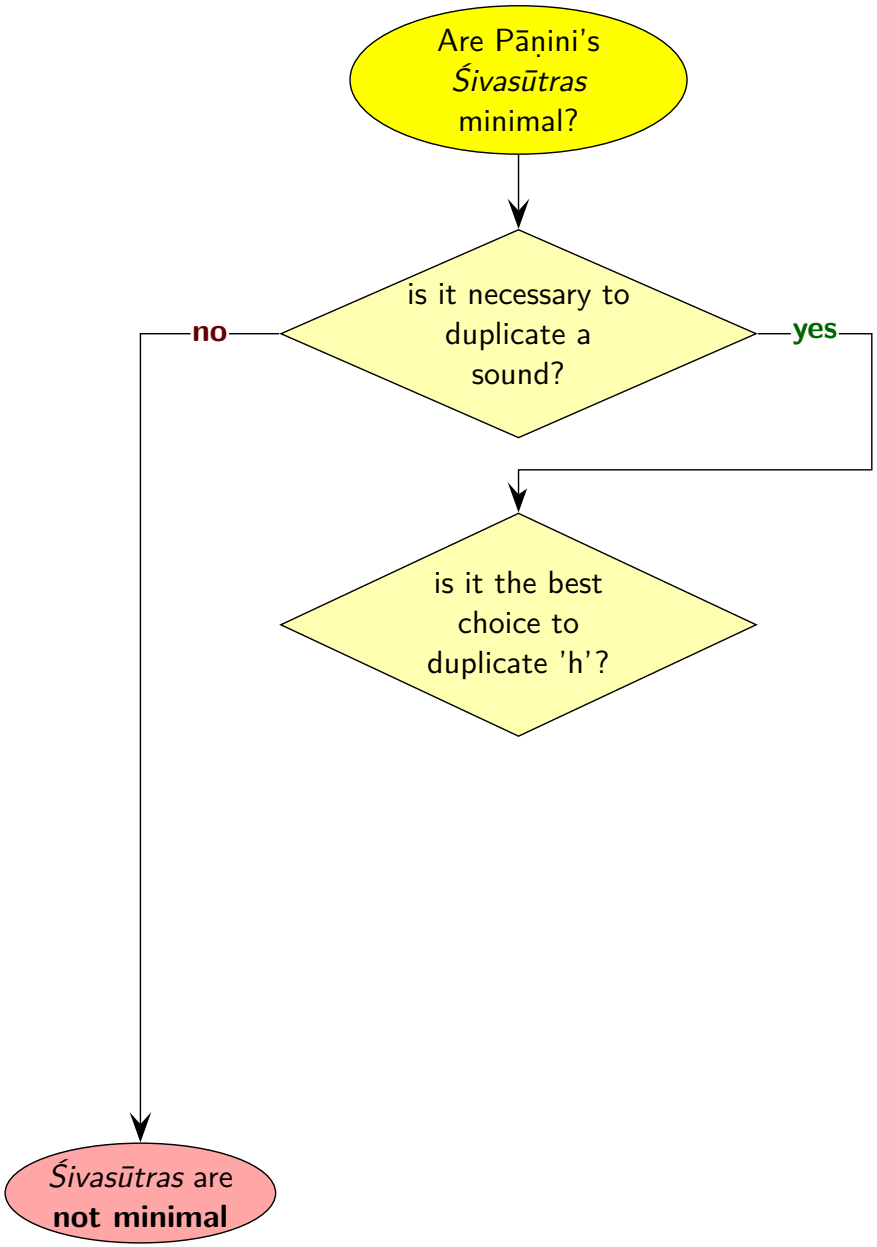
1. each *pratyāhāra* forms an interval ending before an *anubandha*,
2. no sound occurs twice

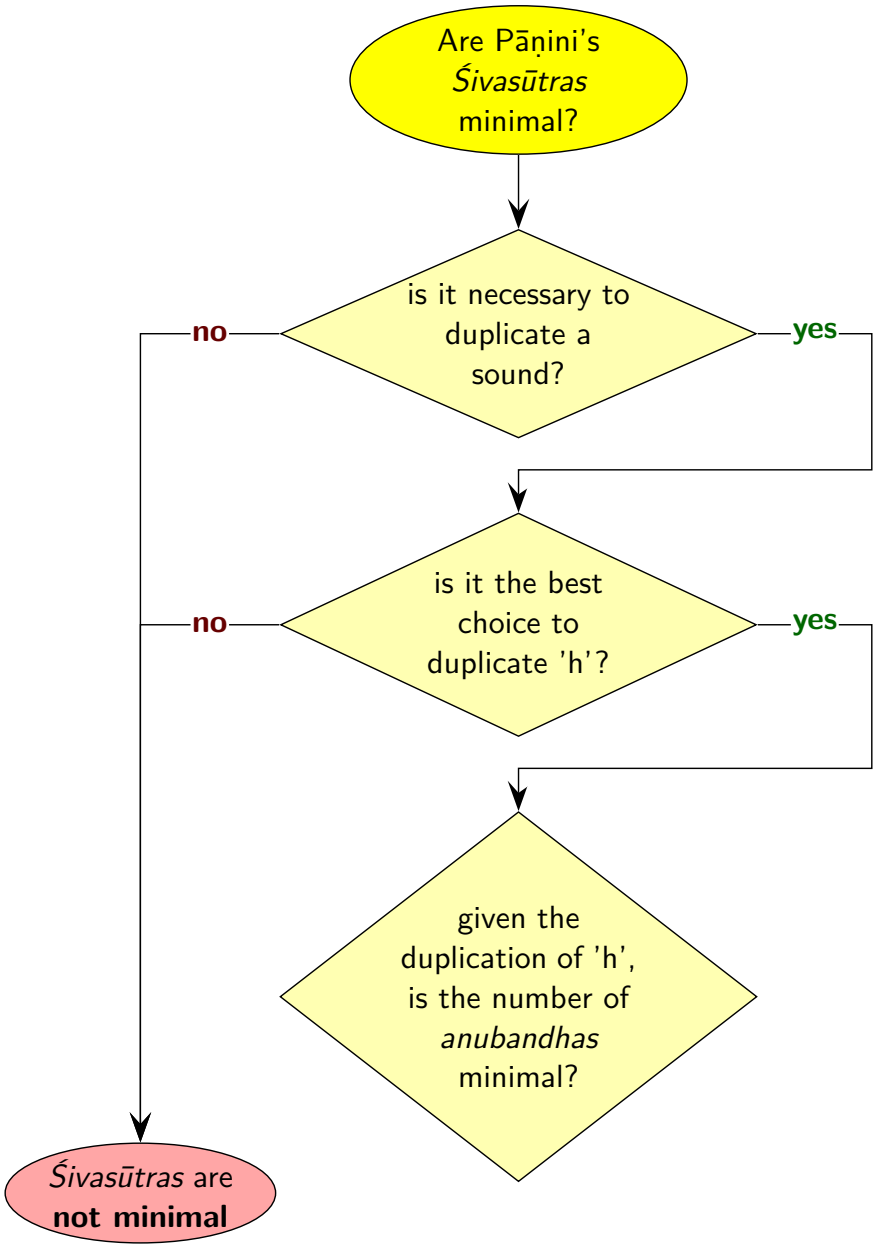
or one sound occurs twice but less *anubandhas* are needed.

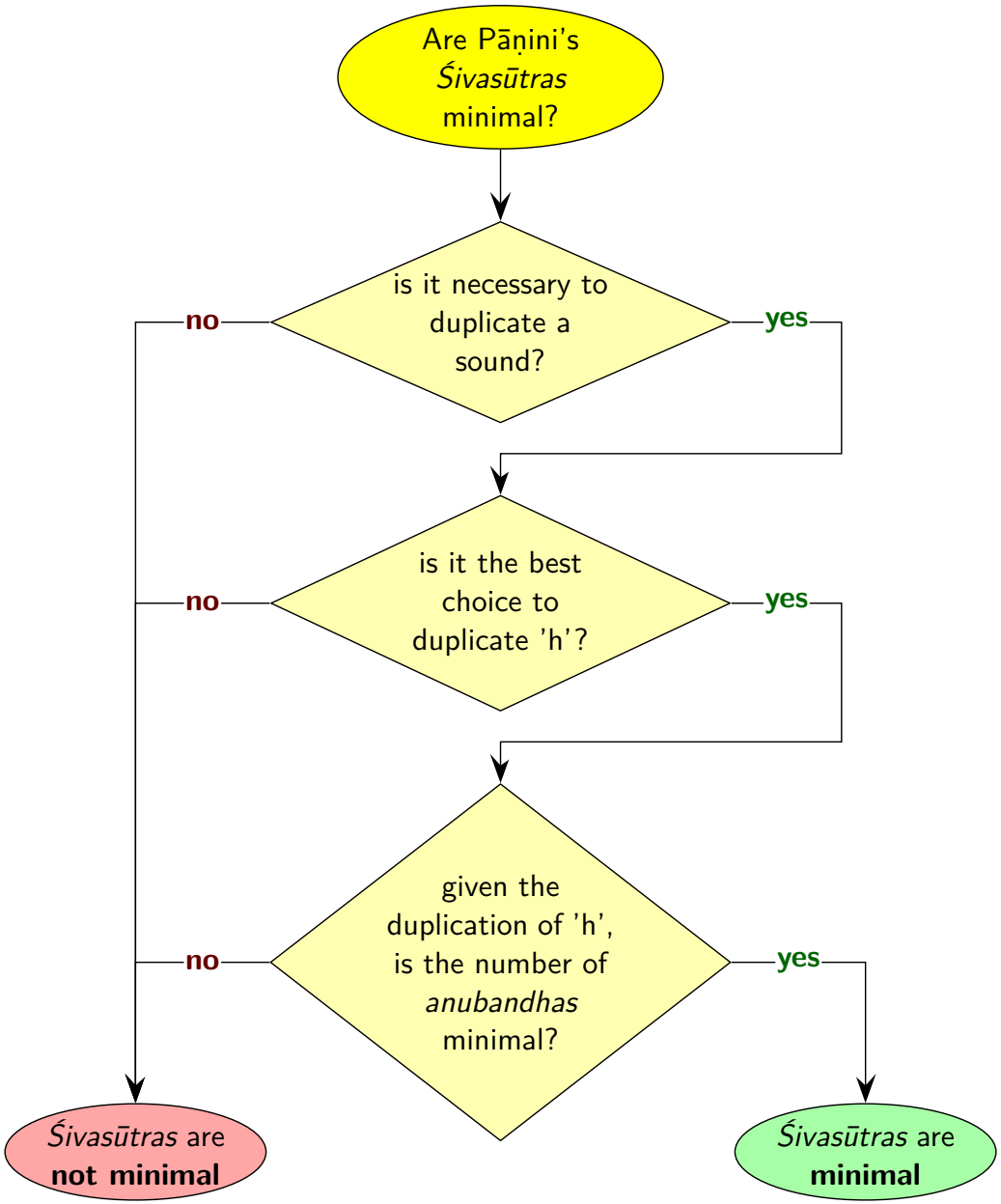
$\Rightarrow$ duplicating a sound is worse than adding *anubandhas*

Are Pāṇini's *Śivasūtras* minimal?


Are Pāṇini's
Śivasūtras
minimal?
is it necessary to
duplicate a
sound?


Are Pāṇini's *Śivasūtras* minimal?
is it necessary to duplicate a sound?
no
yes
is it the best choice to duplicate 'h'?
Śivasūtras are **not minimal**

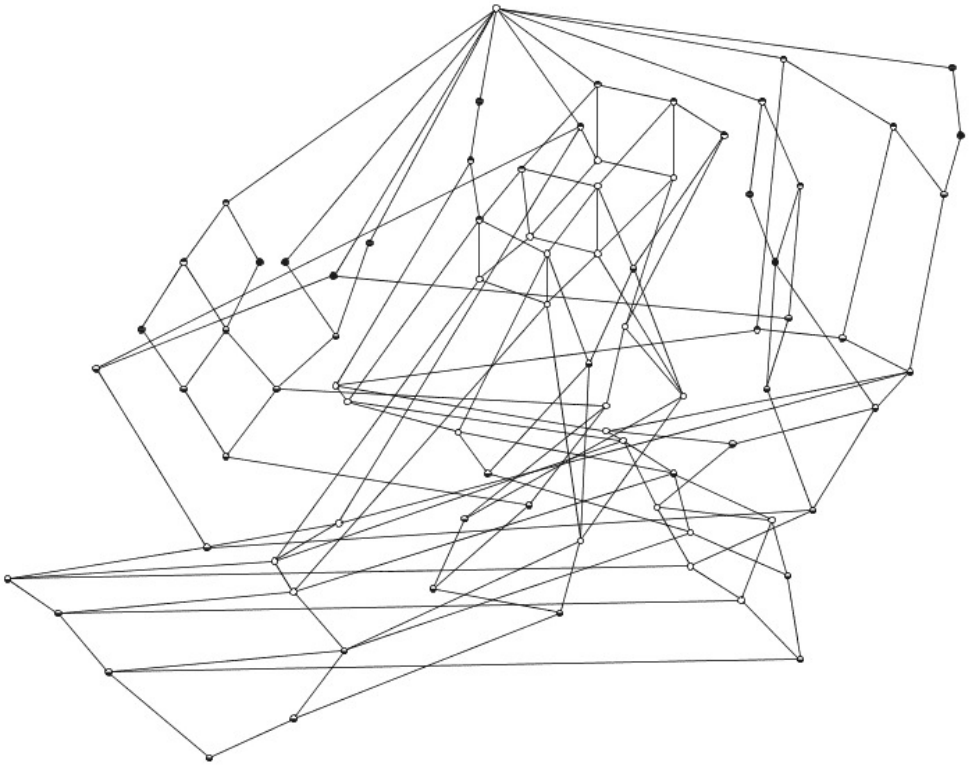

Are Pāṇini's *Śivasūtras* minimal?
is it necessary to duplicate a sound?
no
yes
is it the best choice to duplicate 'h'?
no
yes
given the duplication of 'h', is the number of *anubandhas* minimal?
Śivasūtras are **not minimal**


Are Pāṇini's *Śivasūtras* minimal?
is it necessary to duplicate a sound?
no
yes
is it the best choice to duplicate 'h'?
no
yes
given the duplication of 'h', is the number of *anubandhas* minimal?
no
yes
Śivasūtras are **not minimal**
Śivasūtras are **minimal**

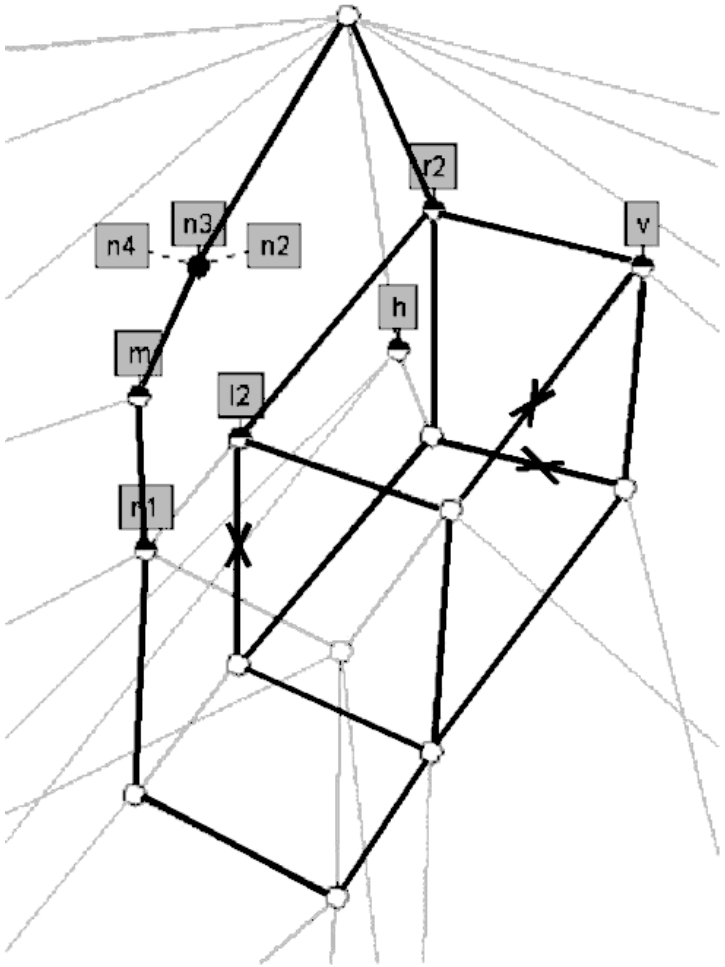
Is it necessary to duplicate a sound?

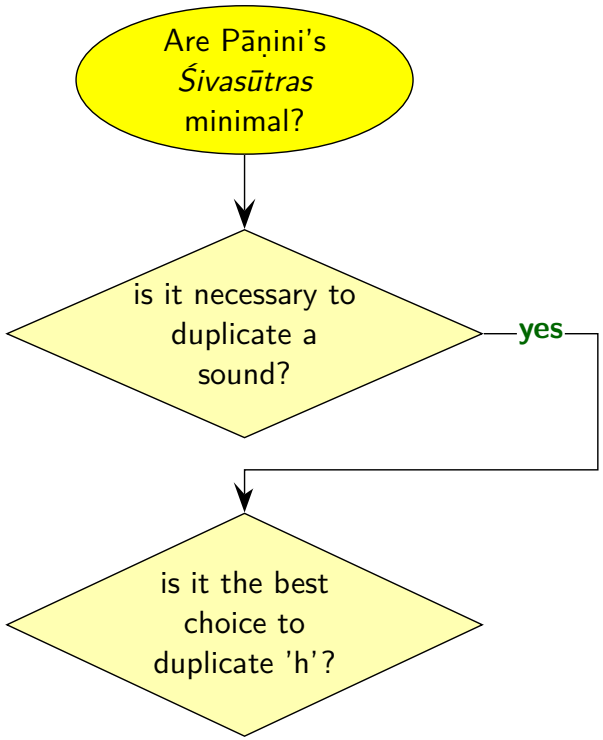
Main theorem on S-sortability (part 1a)

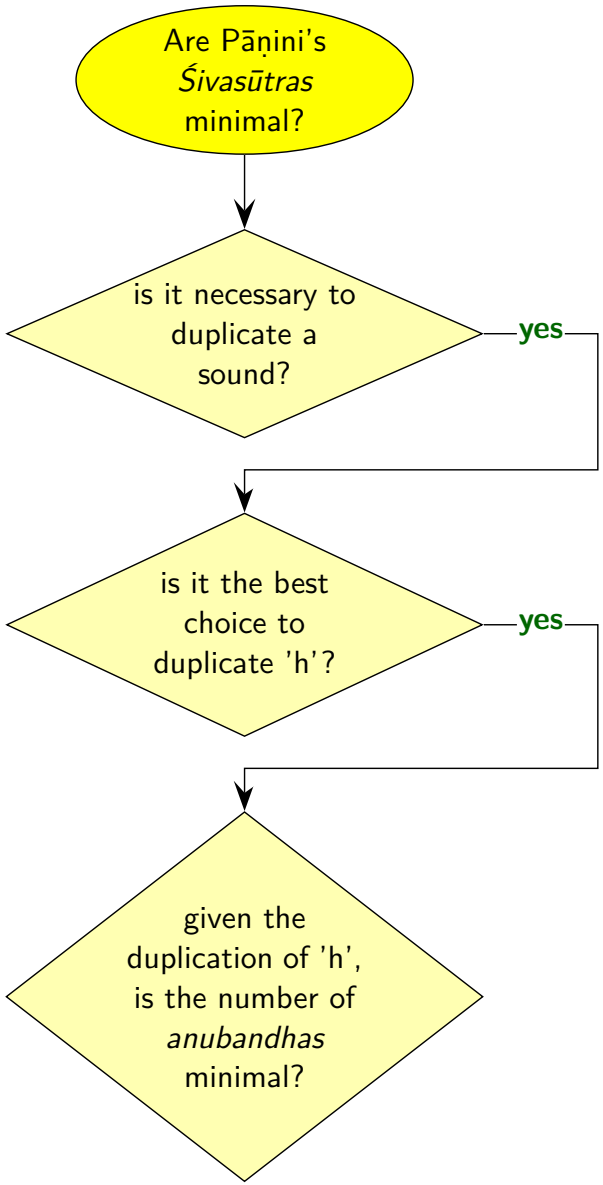
If a set of classes is S-sortable, then its concept lattice is Hasse-planar.



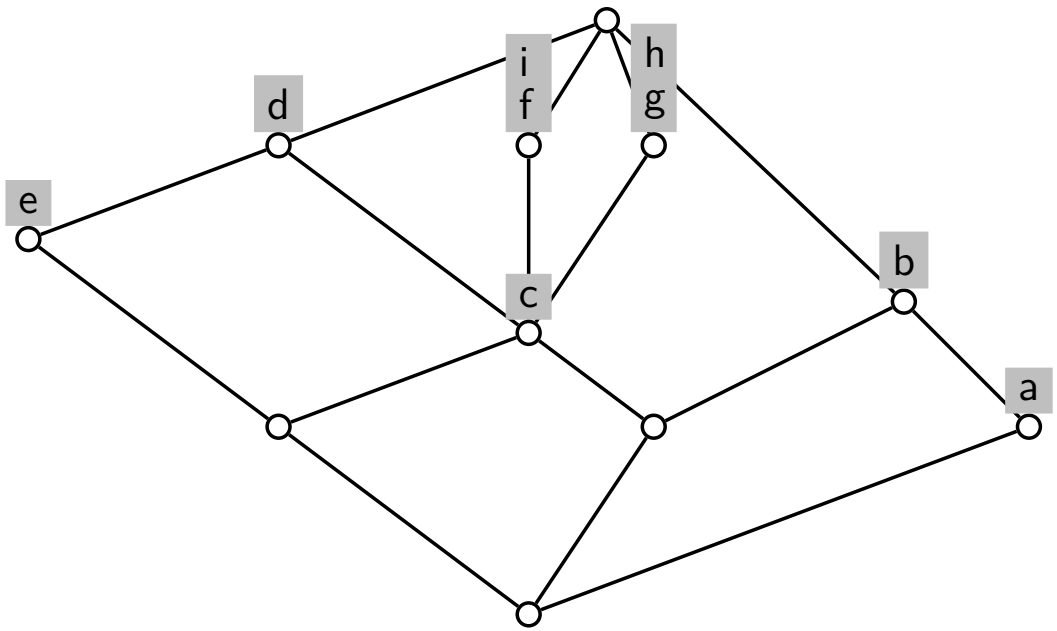
concept lattice of Pāṇini's *pratyāhāras*




Are Pāṇini's
Śivasūtras
minimal?
is it necessary to
duplicate a
sound?
yes
is it the best
choice to
duplicate 'h'?


Are Pāṇini's *Śivasūtras* minimal?
is it necessary to duplicate a sound?
yes
is it the best choice to duplicate 'h'?
yes
given the duplication of 'h', is the number of *anubandhas* minimal?

S-alphabets with a minimal number of markers

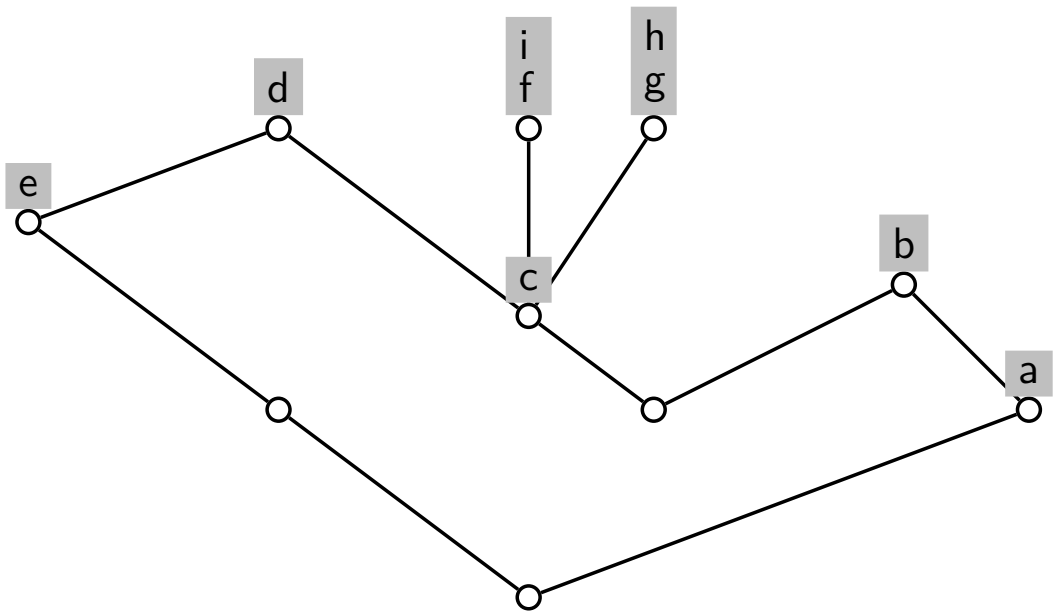


procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a labeled node is reached, add the labels in arbitrary order to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers

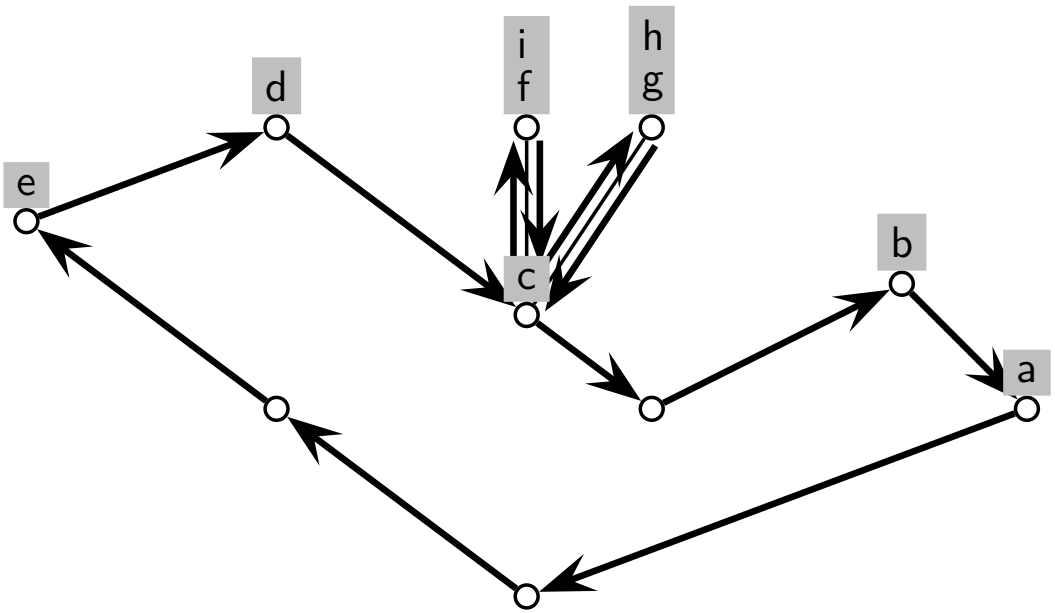


procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a labeled node is reached, add the labels in arbitrary order to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers

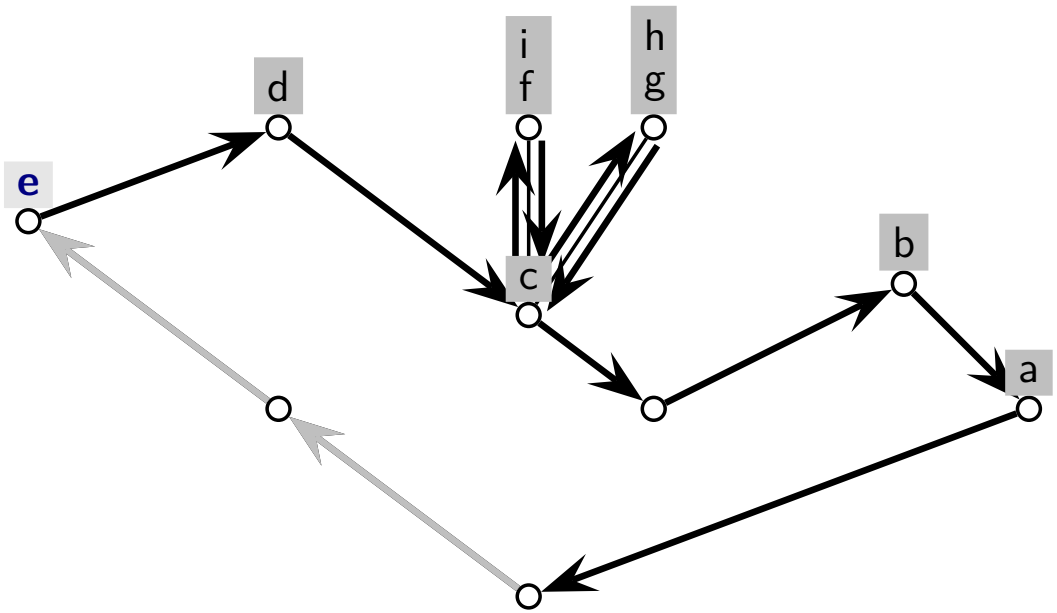


procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers

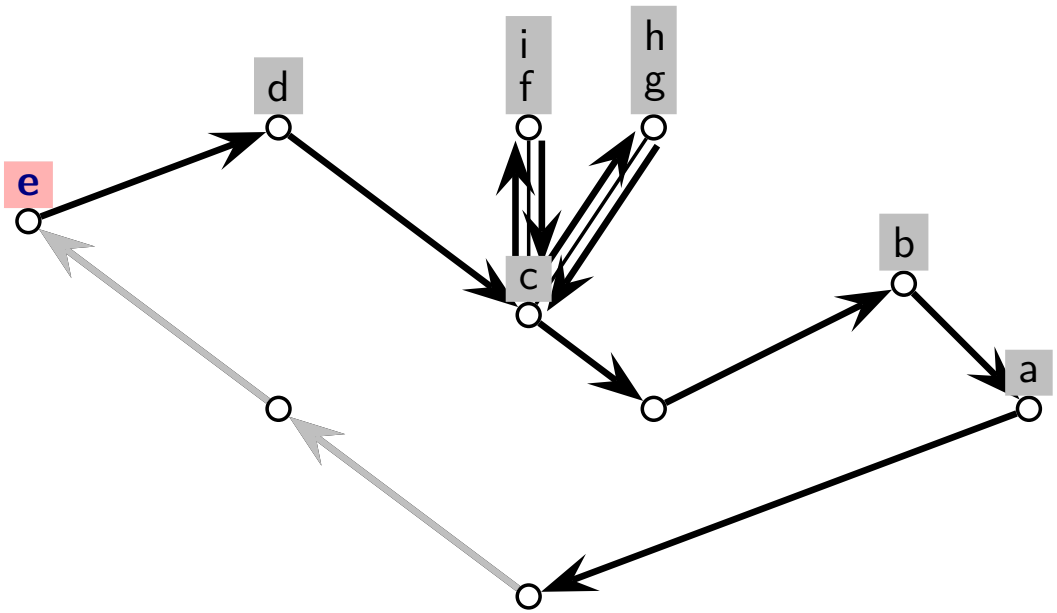


procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



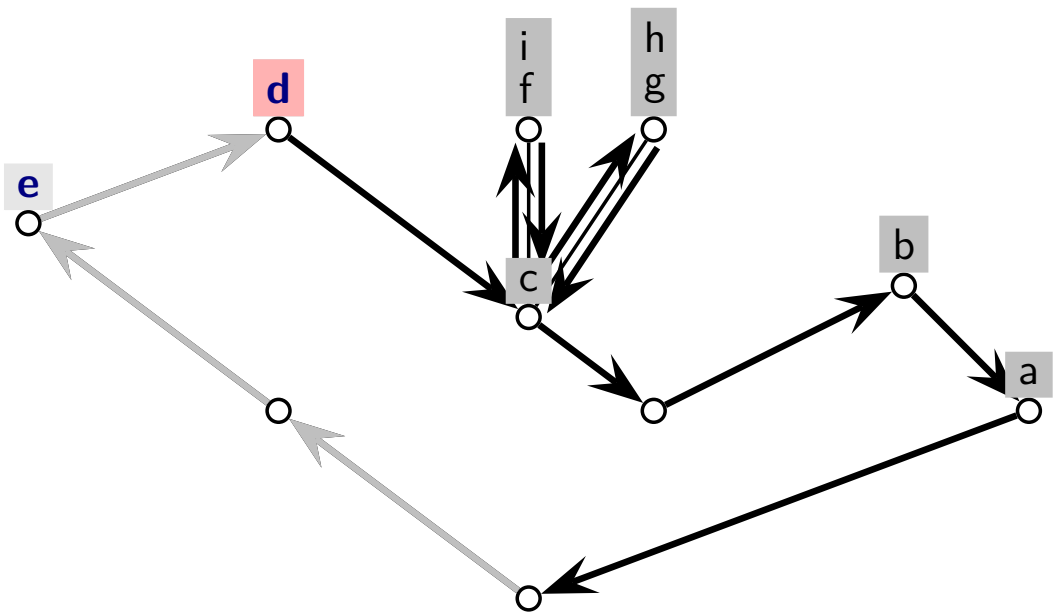
e

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



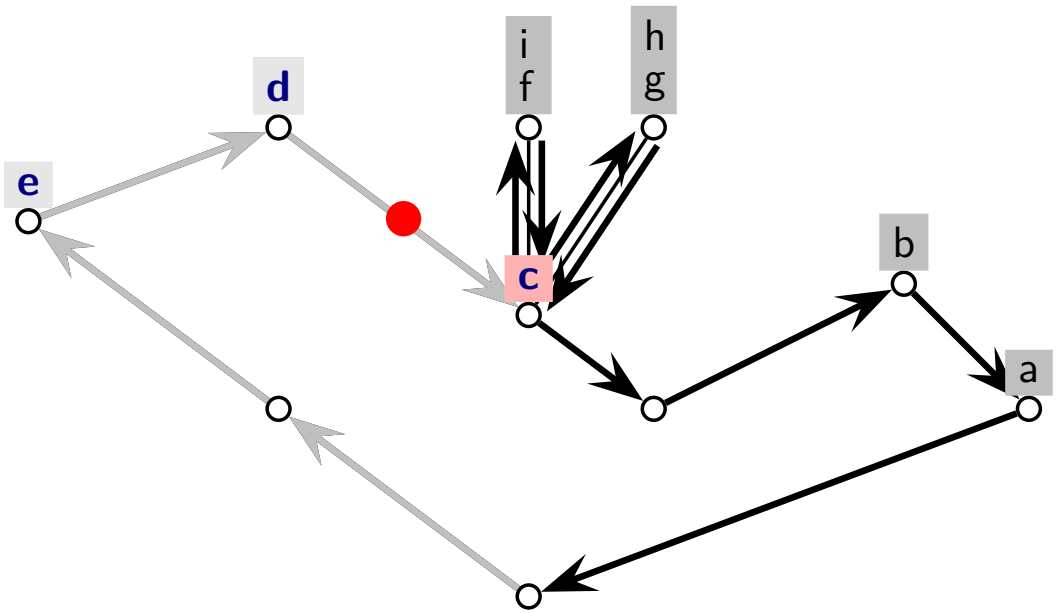
ed

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



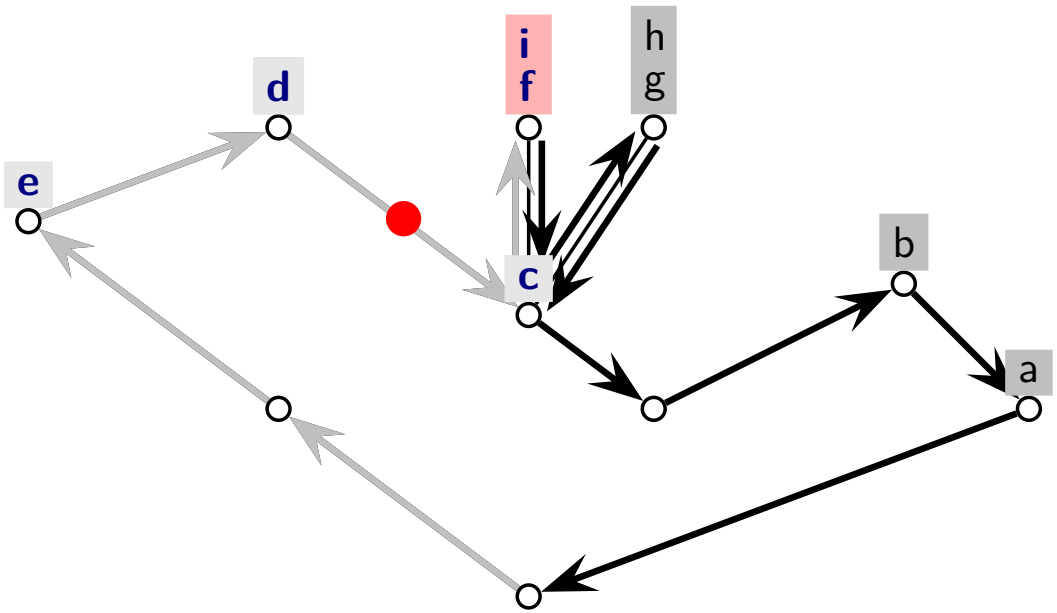
$ed M_1 c$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



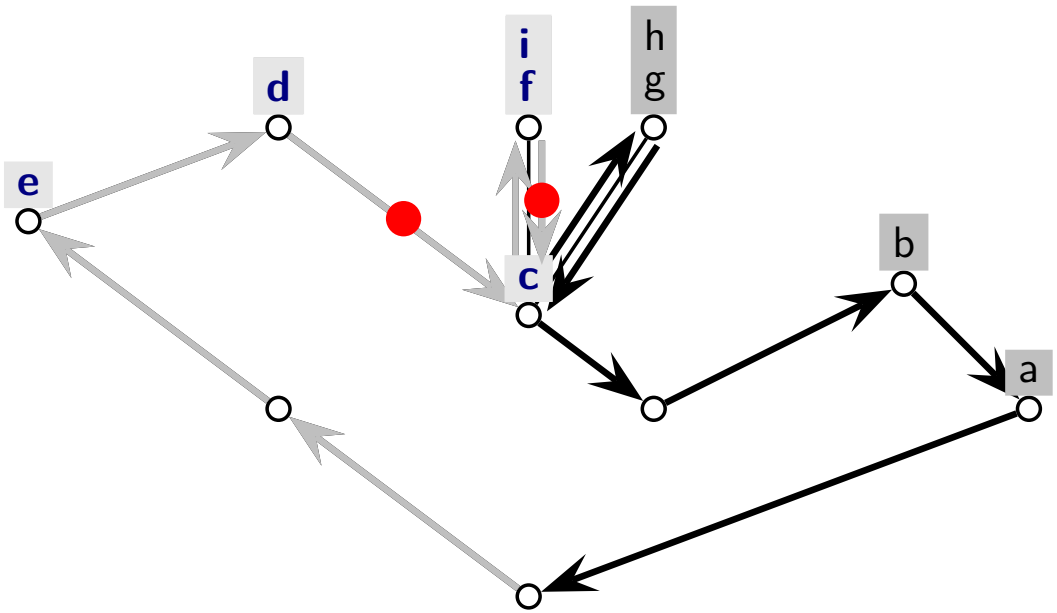
$ed M_1 cfi$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



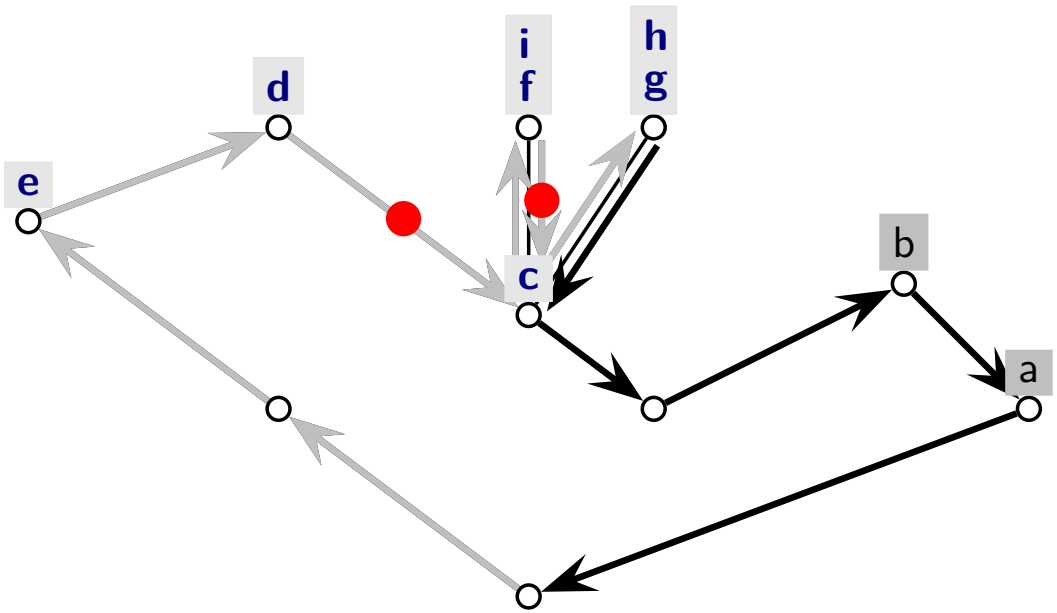
$ed\,M_1\,cfi\,M_2$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



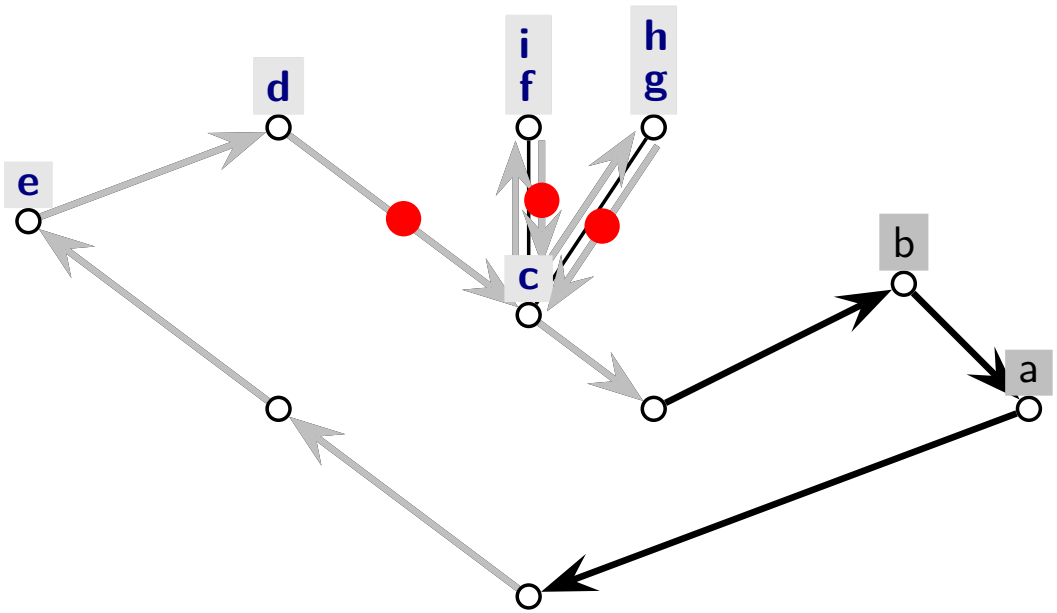
$ed\,M_1\,cfi\,M_2\,gh$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



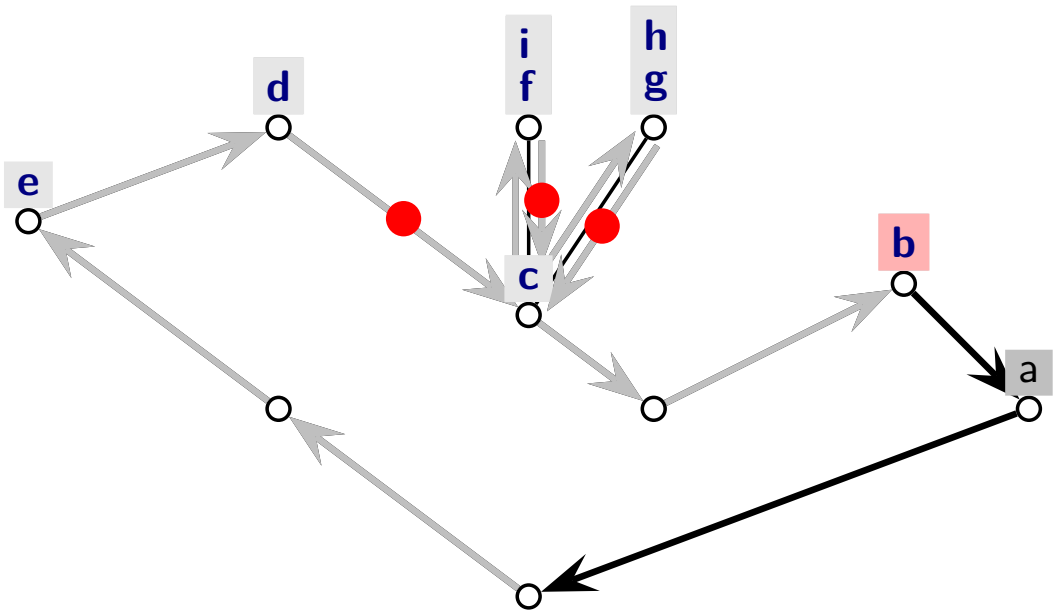
$ed\,M_1\,cfi\,M_2\,gh\,M_3$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



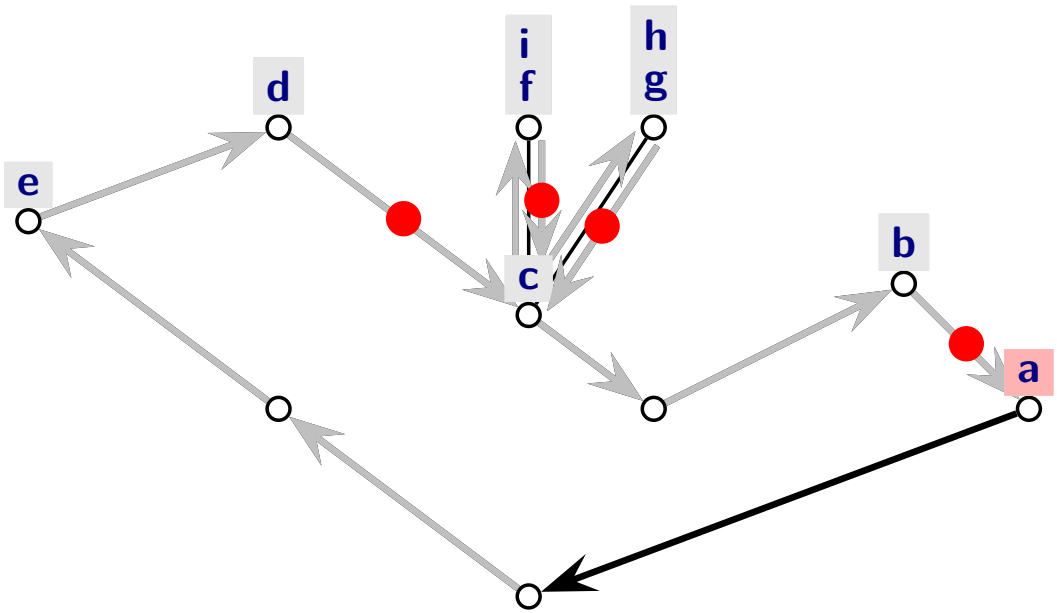
$ed\,M_1\,cfi\,M_2\,ghM_3\,b$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



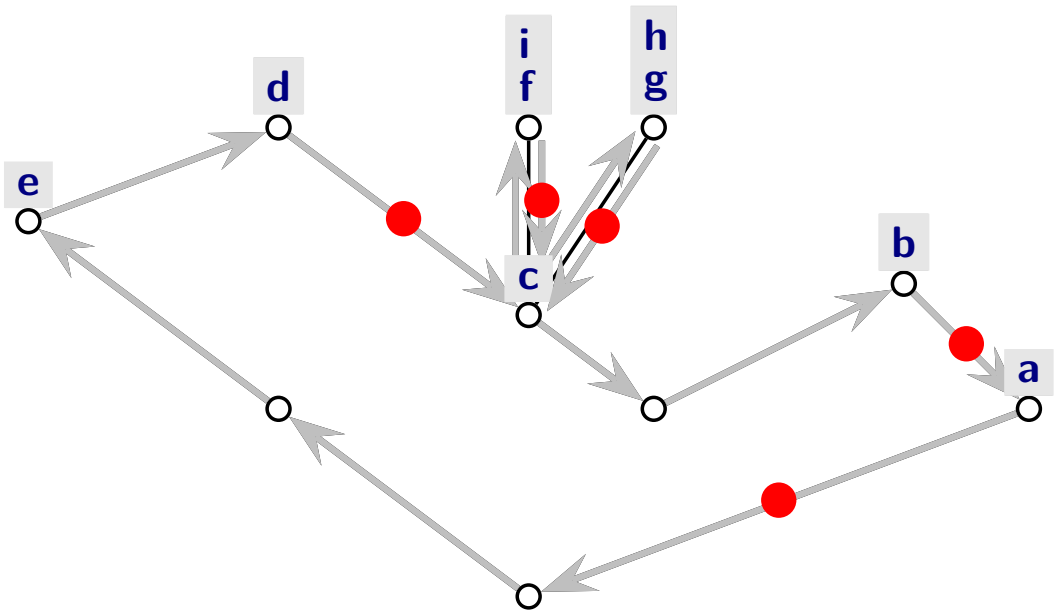
$ed\,M_1\,cfi\,M_2\,ghM_3\,bM_4\,a$

procedure

Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

S-alphabets with a minimal number of markers



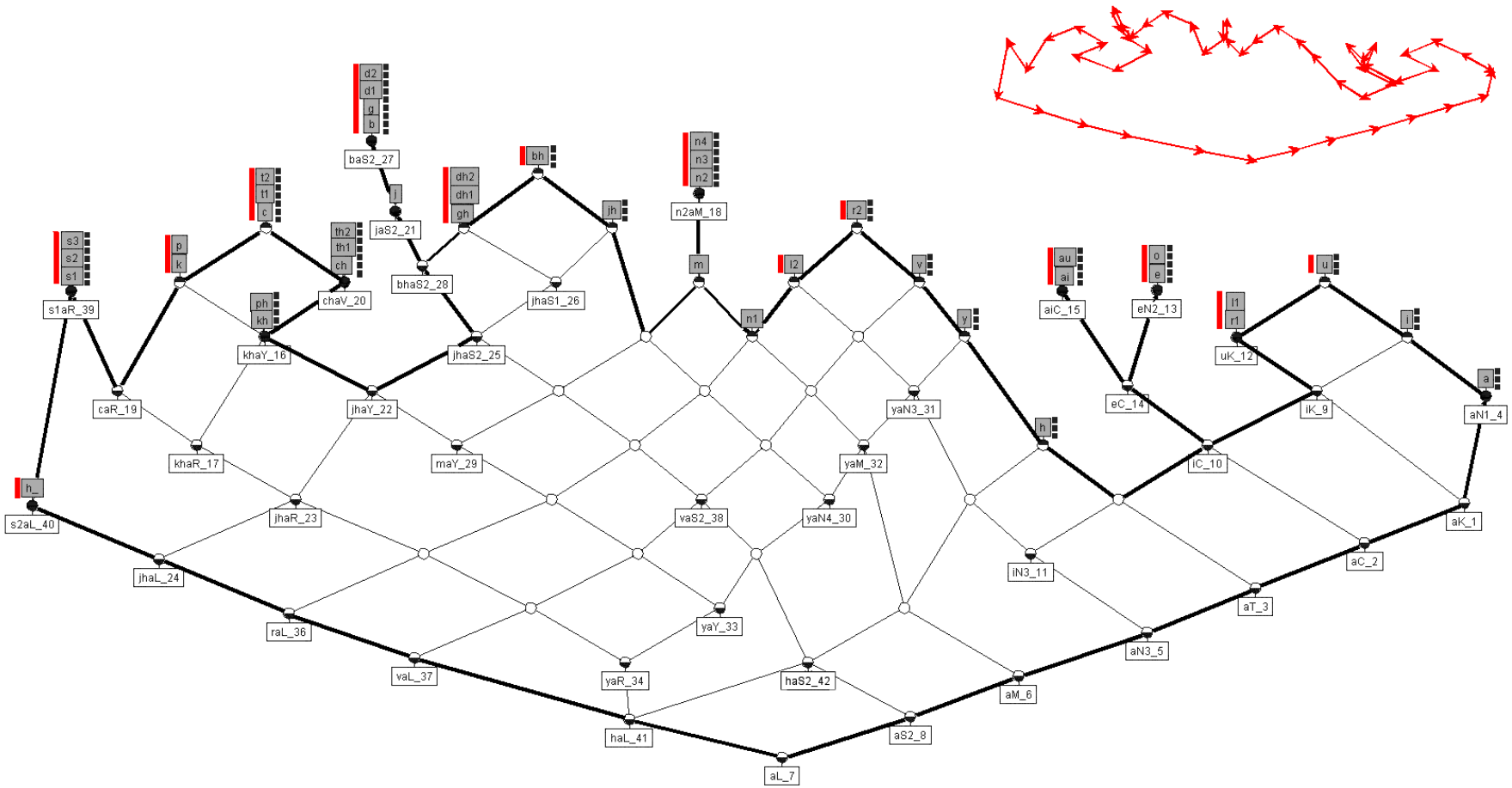
$$\underline{ed\,M_1\,cfi\,M_2\,ghM_3\,b\,M_4\,a\,M_5}$$

procedure

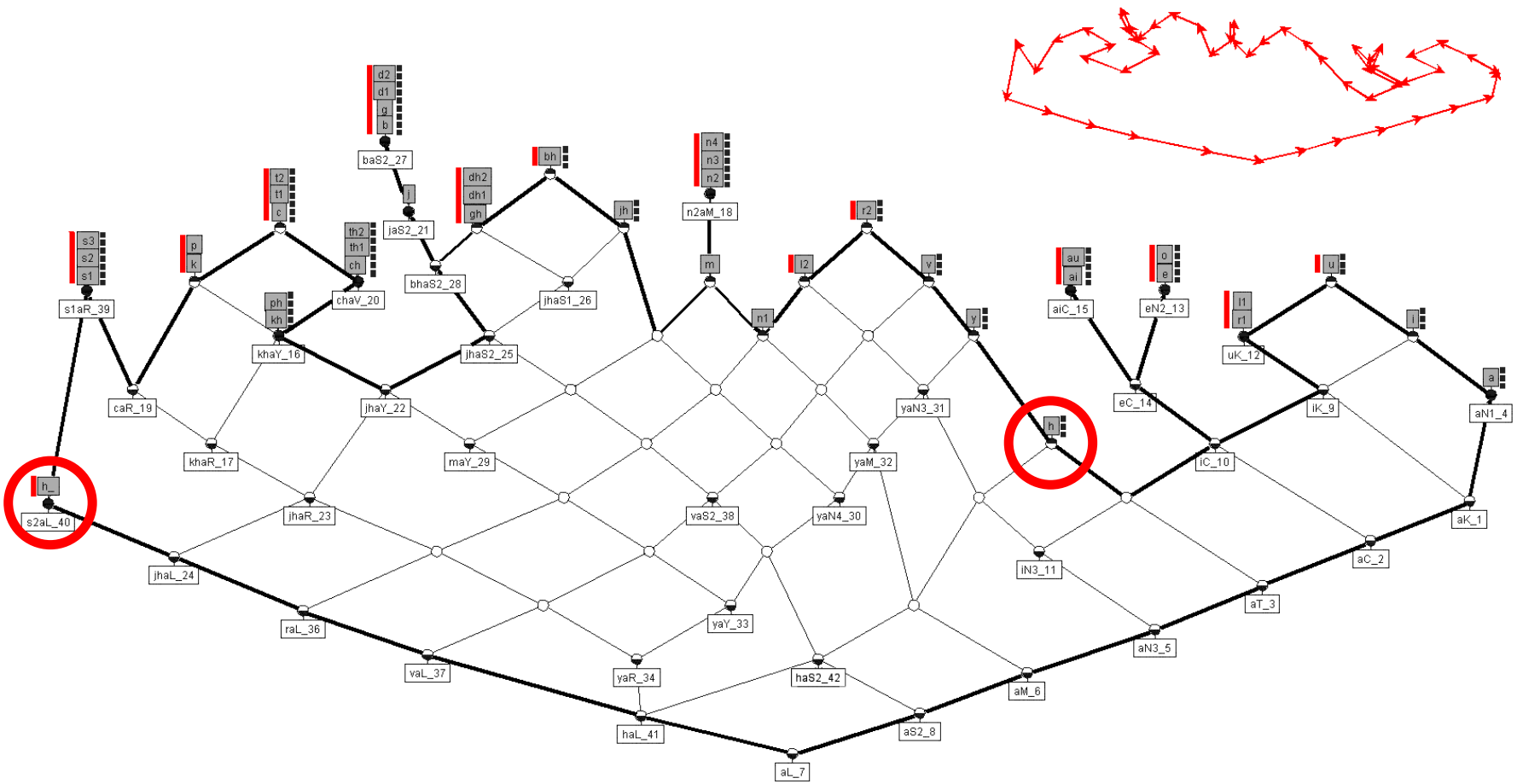
Start with the empty sequence and choose a walk through the S-graph:

- While moving upwards do nothing.
- While moving downwards along an edge add a new marker to the sequence unless its last element is already a marker.
- If a sound is reached, add the sound to the sequence, unless it has been added before.

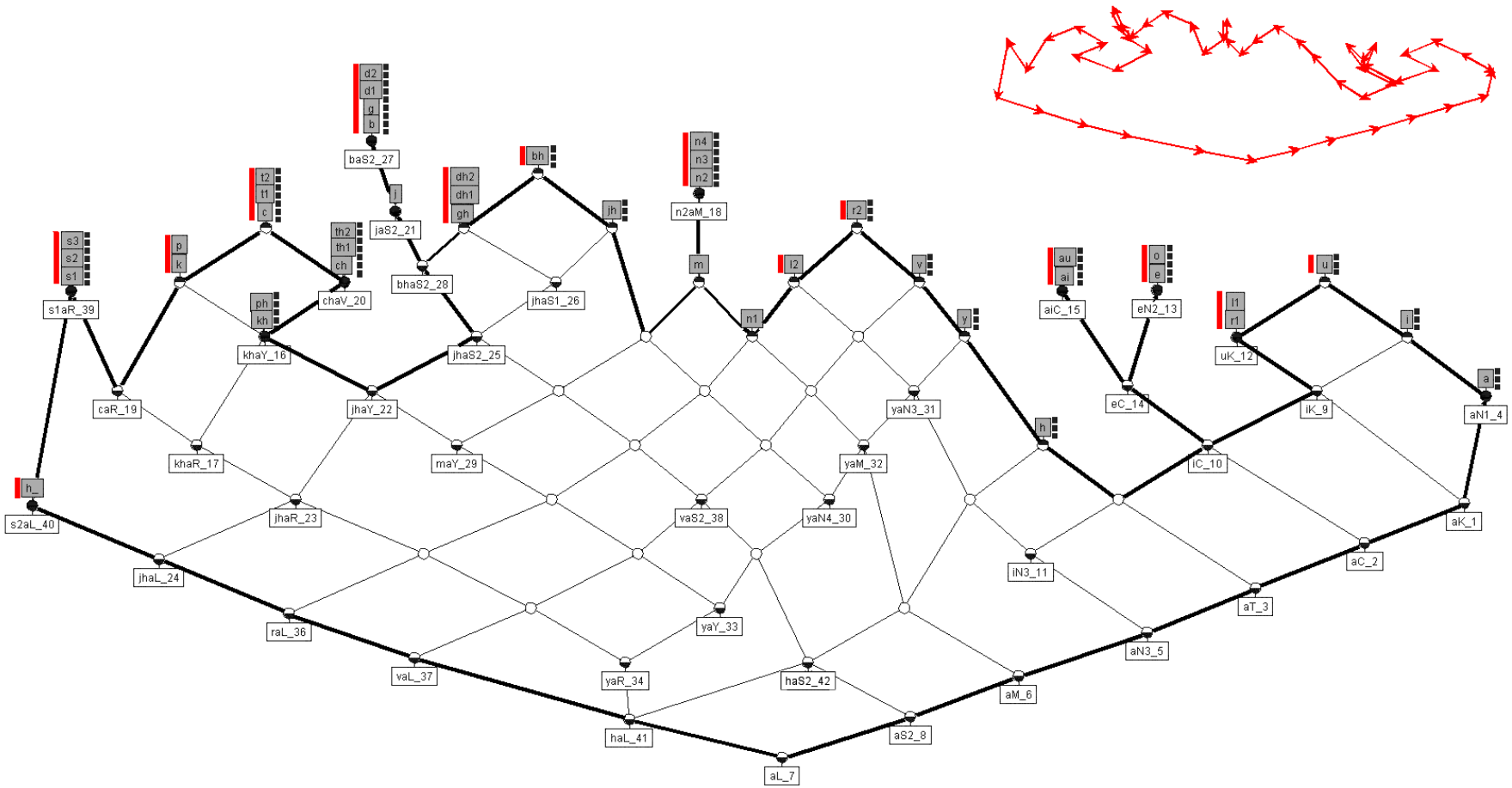
Concept lattice of Pāṇini's *pratyāhāras* with duplicated *h*



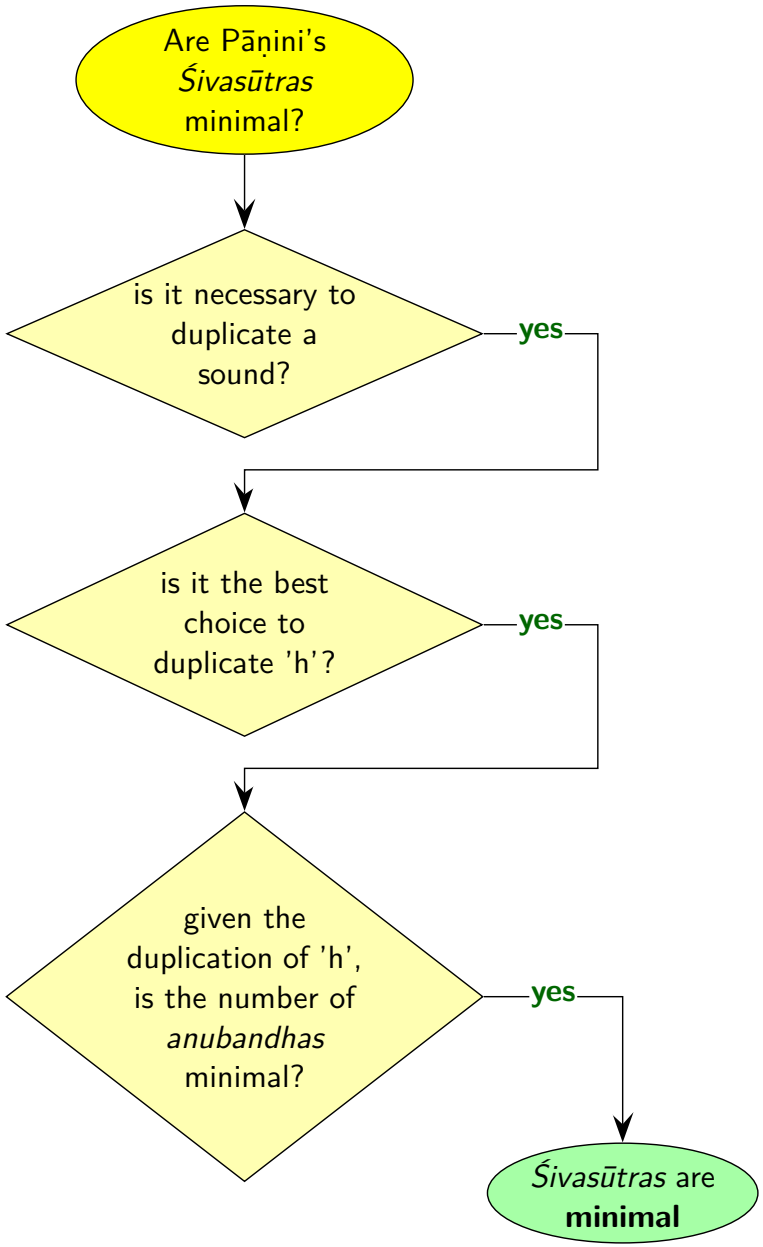
Concept lattice of Pāṇini's *pratyāhāras* with duplicated *h*



Concept lattice of Pāṇini's *pratyāhāras* with duplicated *h*



With the *Śivasūtras* Pāṇini has chosen one out of nearly 12 million minimal S-alphabets!


Are Pāṇini's *Śivasūtras* minimal?
is it necessary to duplicate a sound?
yes
is it the best choice to duplicate 'h'?
yes
given the duplication of 'h', is the number of *anubandhas* minimal?
yes
Śivasūtras are **minimal**

Open problems

What explains the actual structure of the *Śivasūtras*?

- **principle of homorganic continuity** (Staal, 1962)
- **principle of historic continuity** (Cardona, 1969)
- **principle of economy** and **logic of the special case and the general case** (Kiparsky 1991) or **Pāṇini's razor** (Kiparsky 2007)

The presented approach cannot give an answer to this question

The story is much more intricate

- We have neither shown that Pāṇini's technique for the representation of sound classes is optimal
- nor that he has used his technique in an optimal way.

Open problems

What explains the actual structure of the *Śivasūtras*?

- **principle of homorganic continuity** (Staal, 1962)
- **principle of historic continuity** (Cardona, 1969)
- **principle of economy** and **logic of the special case and the general case** (Kiparsky 1991) or **Pāṇini's razor** (Kiparsky 2007)

The presented approach cannot give an answer to this question

The story is much more intricate

- We have neither shown that Pāṇini's technique for the representation of sound classes is optimal
- nor that he has used his technique in an optimal way.

Open problems

What explains the actual structure of the *Śivasūtras*?

- **principle of homorganic continuity** (Staal, 1962)
- **principle of historic continuity** (Cardona, 1969)
- **principle of economy** and **logic of the special case and the general case** (Kiparsky 1991) or **Pāṇini's razor** (Kiparsky 2007)

The presented approach cannot give an answer to this question

The story is much more intricate

- We have neither shown that Pāṇini's technique for the representation of sound classes is optimal
- nor that he has used his technique in an optimal way.

Literature

- Kiparsky, P. (1991), Economy and the construction of the Śivasūtras. In: M. M. Deshpande & S. Bhate (eds.), *Pāṇinian Studies*, Michigan: Ann Arbor.
- Petersen, W. (2008), Zur Minimalität von Pāṇinis *Śivasūtras* – Eine Untersuchung mit Mitteln der Formalen Begriffsanalyse. PhD thesis, university of Düsseldorf.
- Petersen, W. (2009), On the Construction of Sivasutra-Alphabets. In: A. Kulkarni and G. Huet (eds.): *Sanskrit Computational Linguistics.* LNCS 5406, Springer.
- Staal, F. (1962), A Method of Linguistic Description. Language **38**, 1-10.

Origin of Pictures

- libraries (left):
  `http://www.meduniwien.ac.at/medizinischepsychologie/bibliothek.htm`
- libraries (middle): `http://www.math-nat.de/aktuelles/allgemein.htm`
- libraries (right):
  `http://www.geschichte.mpg.de/deutsch/bibliothek.html`
- warehouses:
  `http://www.metrogroup.de/servlet/PB/menu/1114920_l1/index.html`
- stores: `http://www.einkaufsparadies-schmidt.de/01bilder01/`