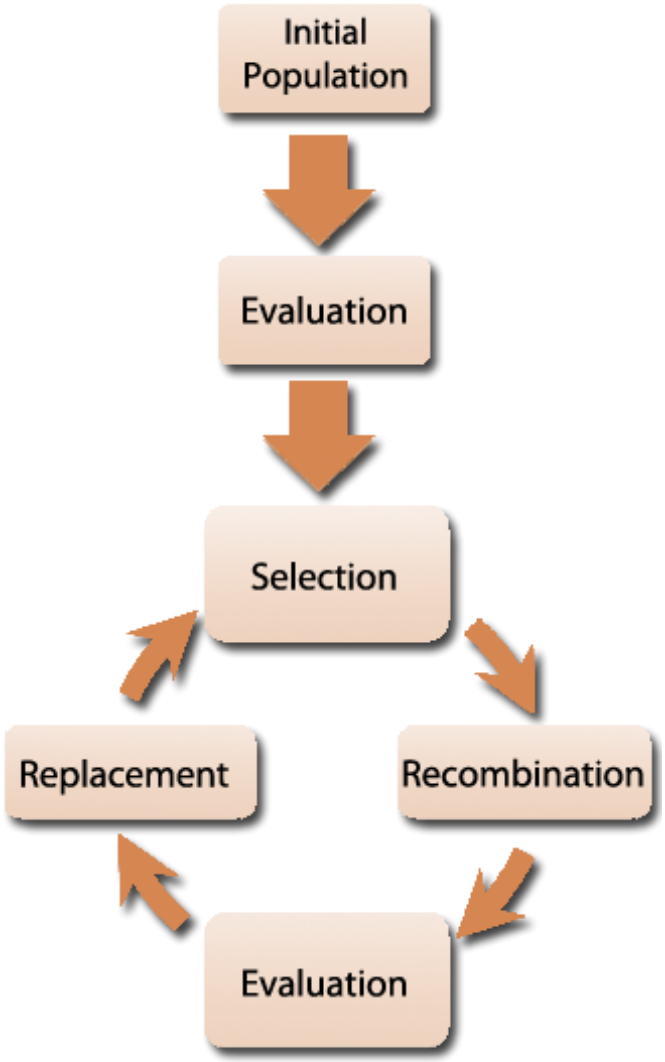
Reusing Alignments

Borja Vázquez- Barreiros
Sebastiaan van Zelst
Joos Buijs

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts

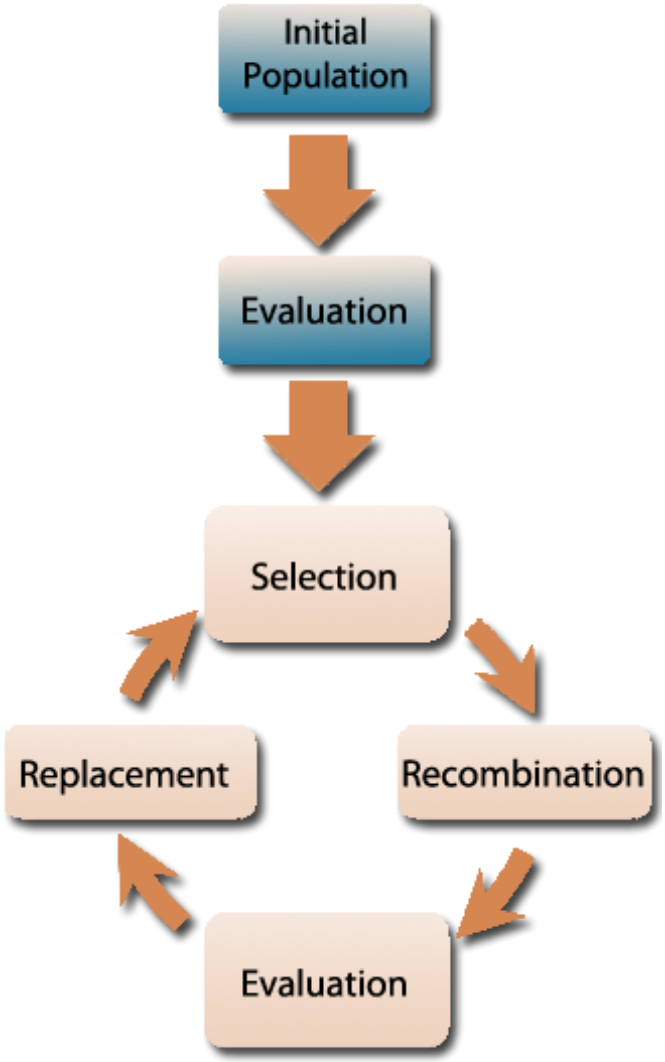
Motivation



Evolutionary Tree Miner

- **Population:** set of process trees
- **The population evolves until obtain an optimal solution**

Motivation

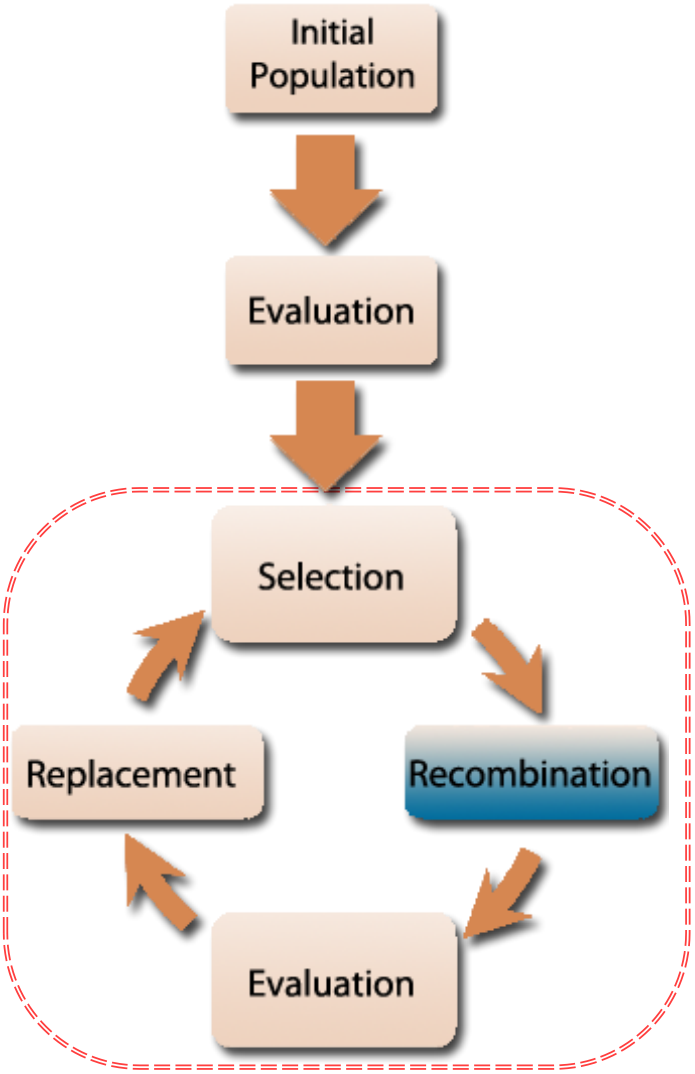


Evolutionary Tree Miner

1. Creation and evaluation of the initial population

2. Evolve the initial population in the evolutionary cycle

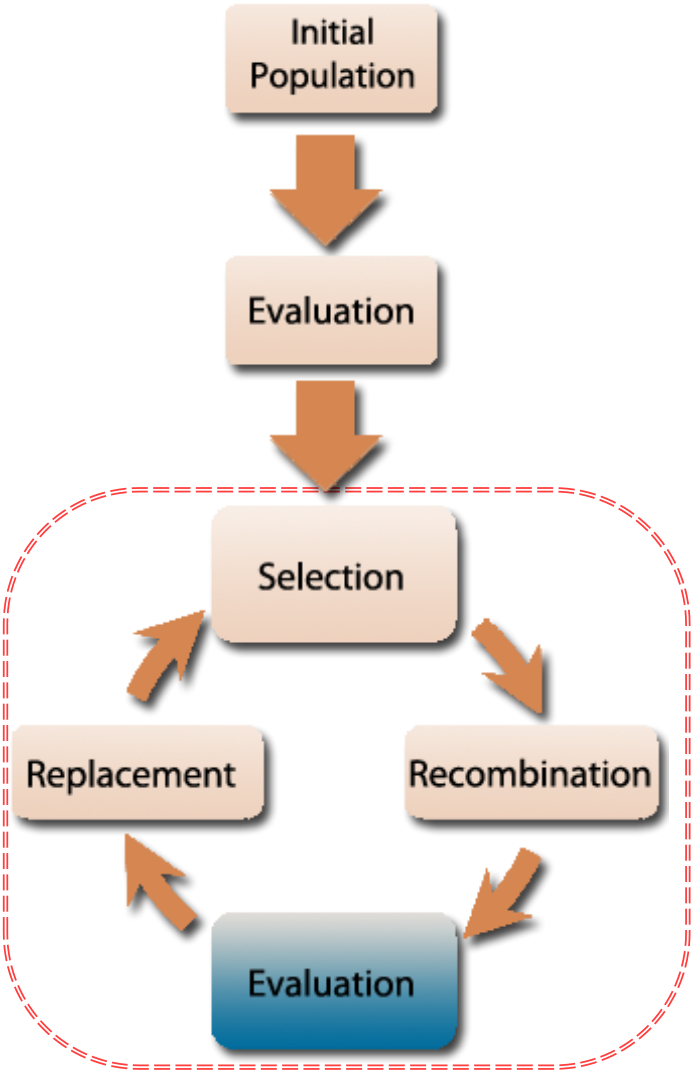
Motivation



Evolutionary cycle:

1. **Mutate the current population:**
   - Adding, changing or removing behavior

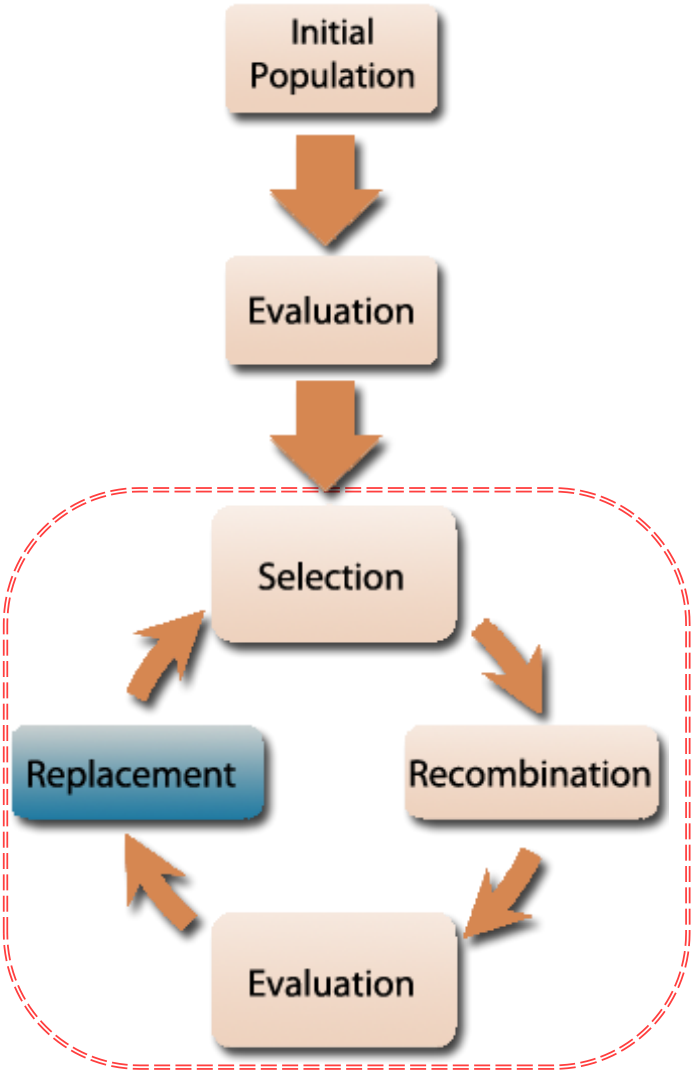
Motivation



Evolutionary cycle:

1. **Mutate the current population:**
   - Adding, changing or removing behavior
2. **Compute the alignments of each process tree**

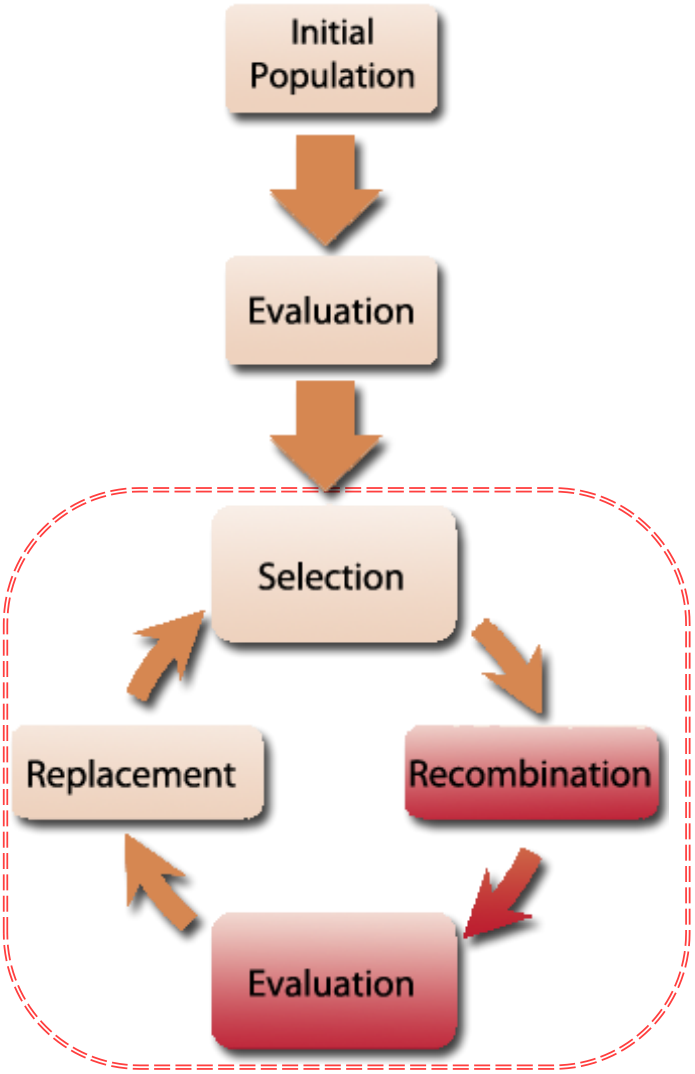
Motivation



Evolutionary cycle:

1. **Mutate the current population:**
   - Adding, changing or removing behavior
2. **Compute the alignments of each process tree**
3. **Replace population**

Motivation

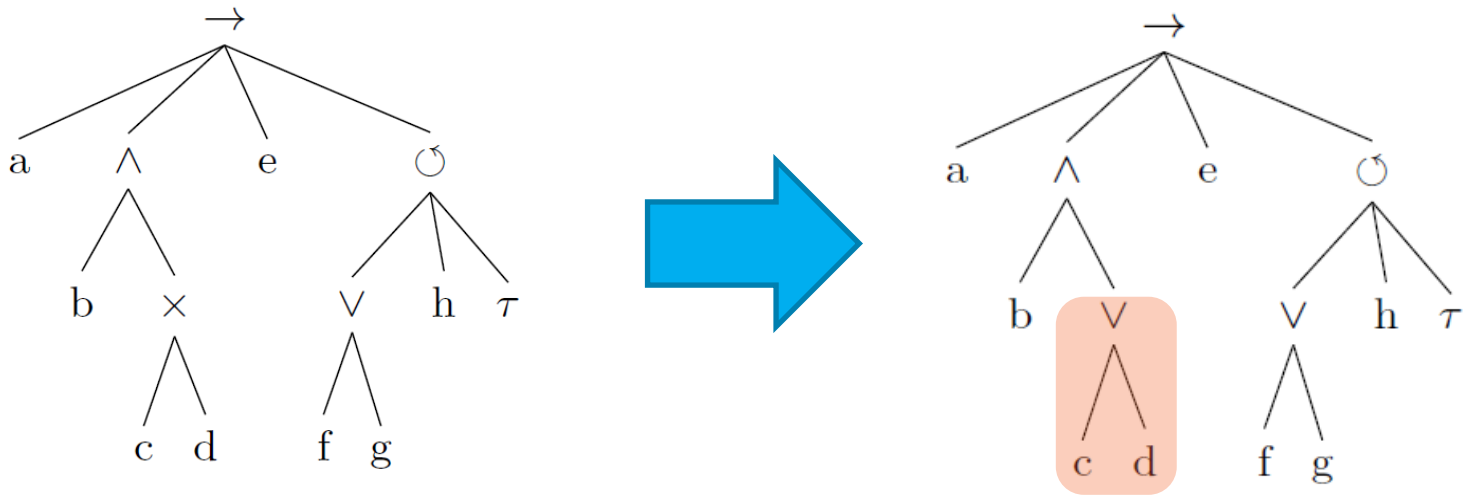


Bottleneck:

1. **Mutate the current population:**
   - Adding, changing or removing behavior
2. **Compute the alignments of each process tree**

Hypothesis

Can we compute an alignment by **reusing an old alignment**, instead of computing the alignment from scratch?

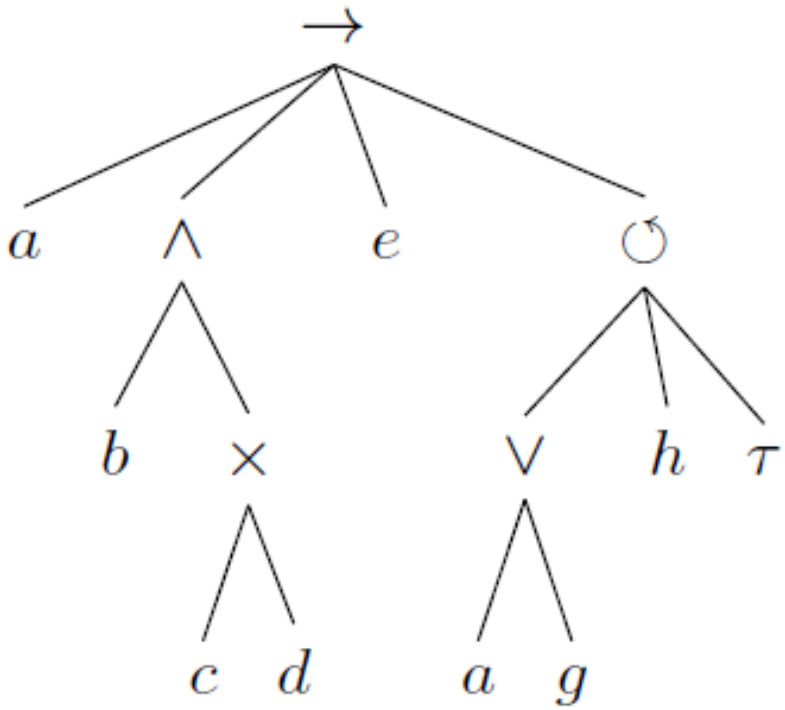


trace	**a**	**b**	**c**	**d**	**e**	**f**
model	**a**	**b**	**c**	**≫**	**e**	**f**

trace	**?**	**?**	**?**	**?**	**?**	**?**
model	?	?	?	?	?	?

- Optimality **is not** guaranteed

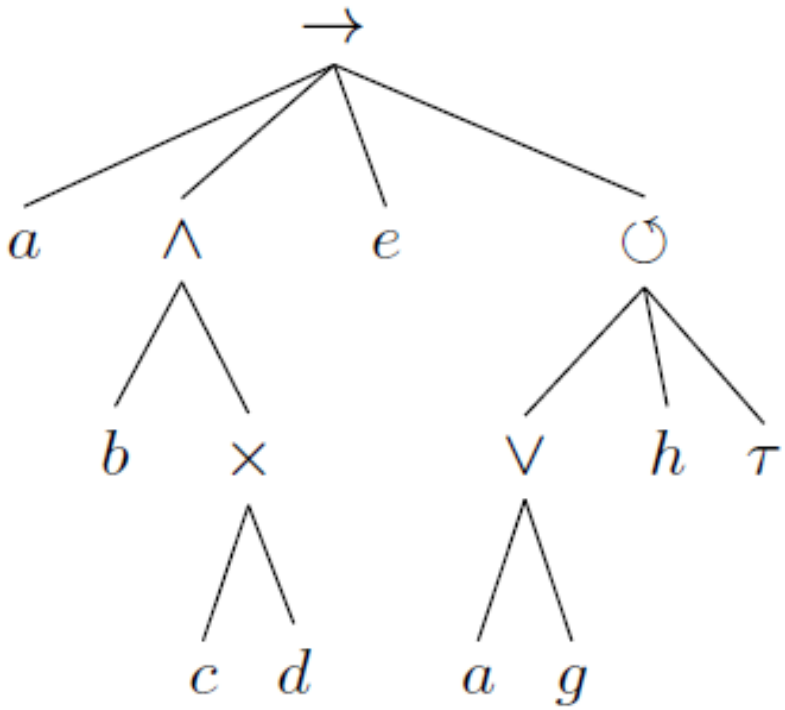
Preliminaries: Process Trees



{<a,b,c,e,a,b,g,h,a>, <a,b,d,e,a>, <a,b,d,e,g,h,a,g>}

- **Block structured workflow net**
- **Leafs:** activities
  - { a, b, c, d, e, g, h, τ }
- **Nodes:** operators
  - Sequence: →
  - AND: ∧
  - OR: ∨
  - XOR: ×
  - LOOP: ↺

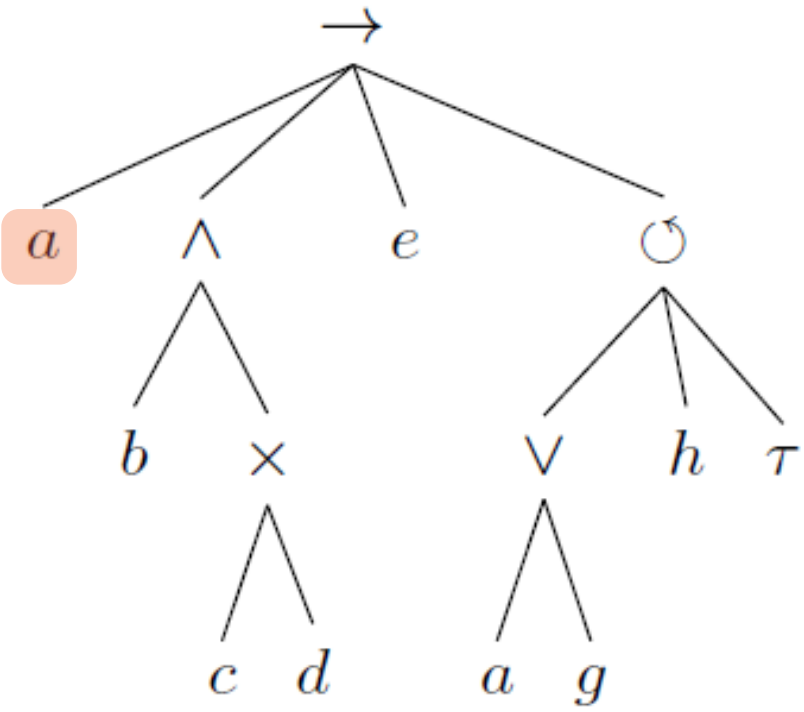
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

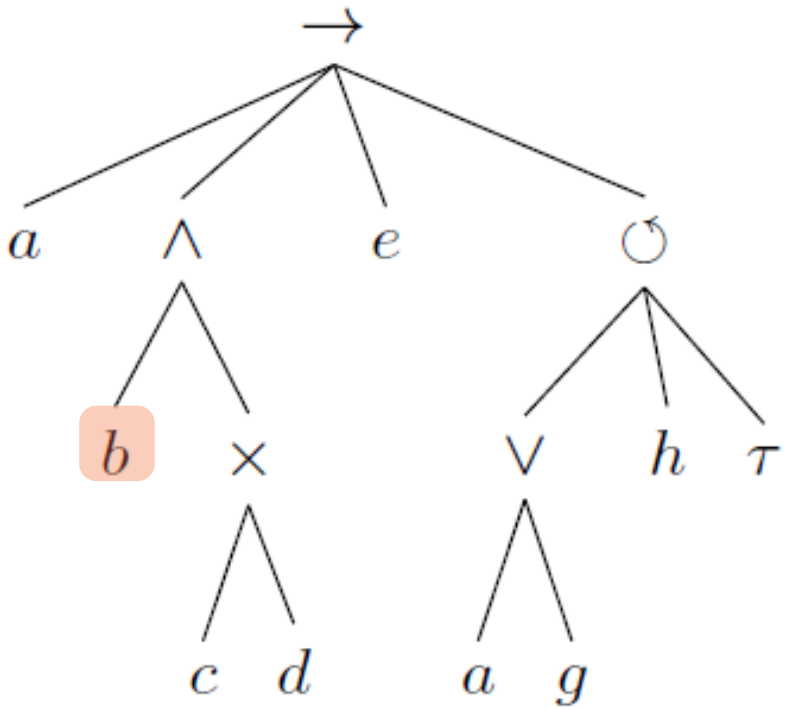
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

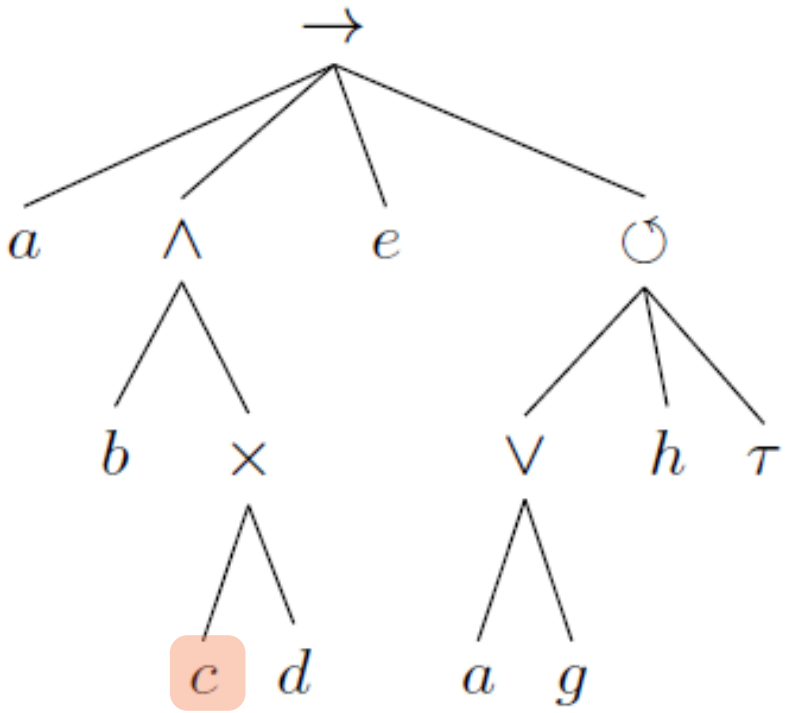
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

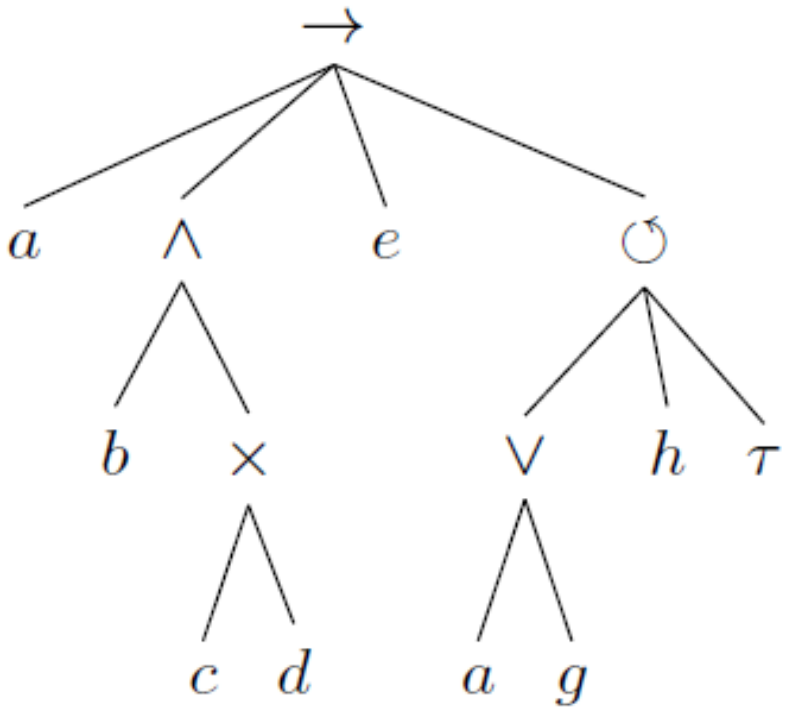
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

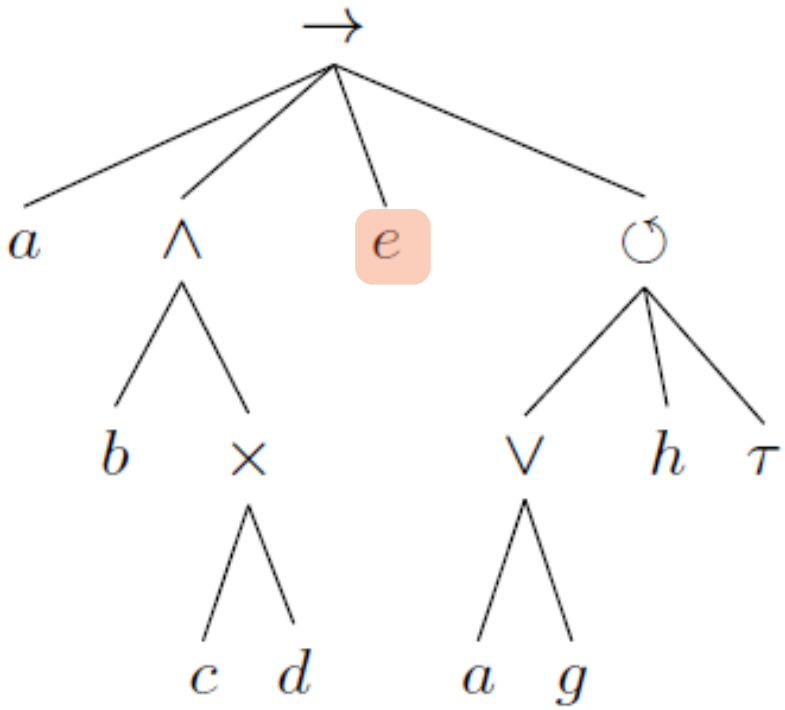
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

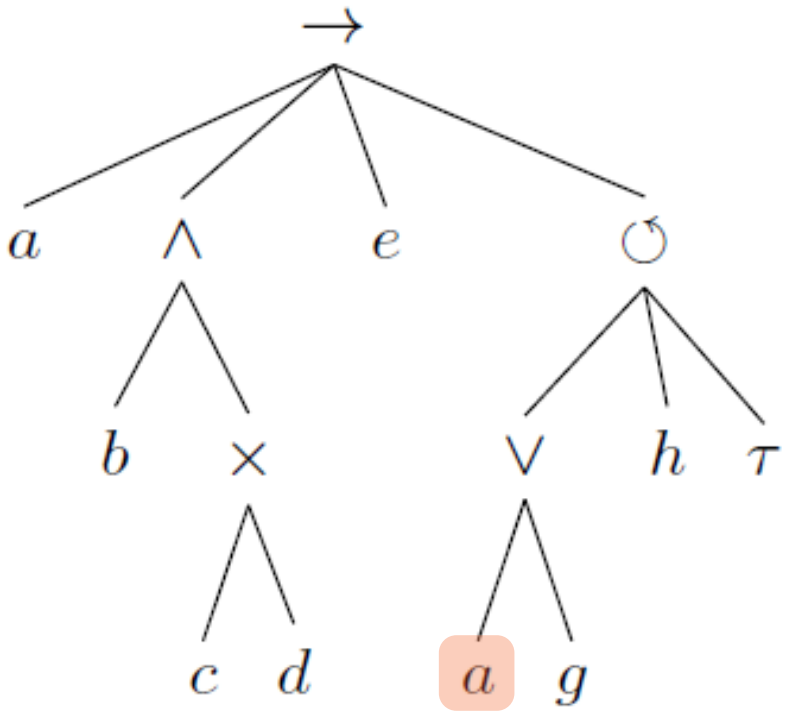
Preliminaries: Alignments in process trees (i)



- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

Preliminaries: Alignments in process trees (i)



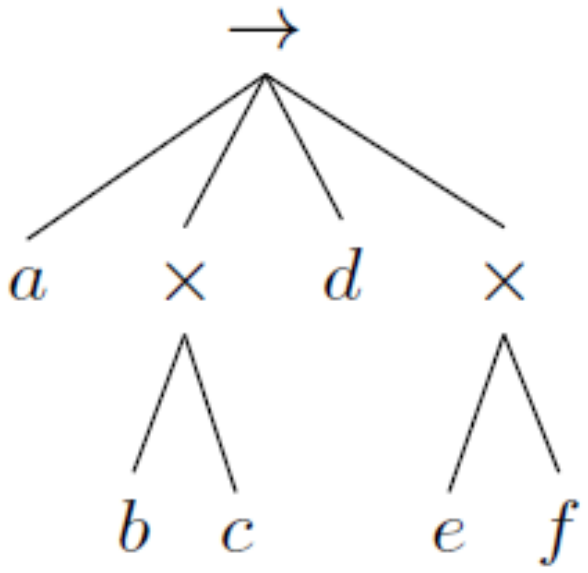
- **Compute the deviations between a model a a log**
- **Map events in a trace to leafs**
  - **Example**
    - Trace <a,b,c,d,a>

trace	a	b	c	d	$\gg$	a
model	a	b	c	$\gg$	e	a

Overview

- **General approach:**
  - **Example of how we can reuse an alignment**
  - **Example proving non-optimality**
- **Validation and experiments**

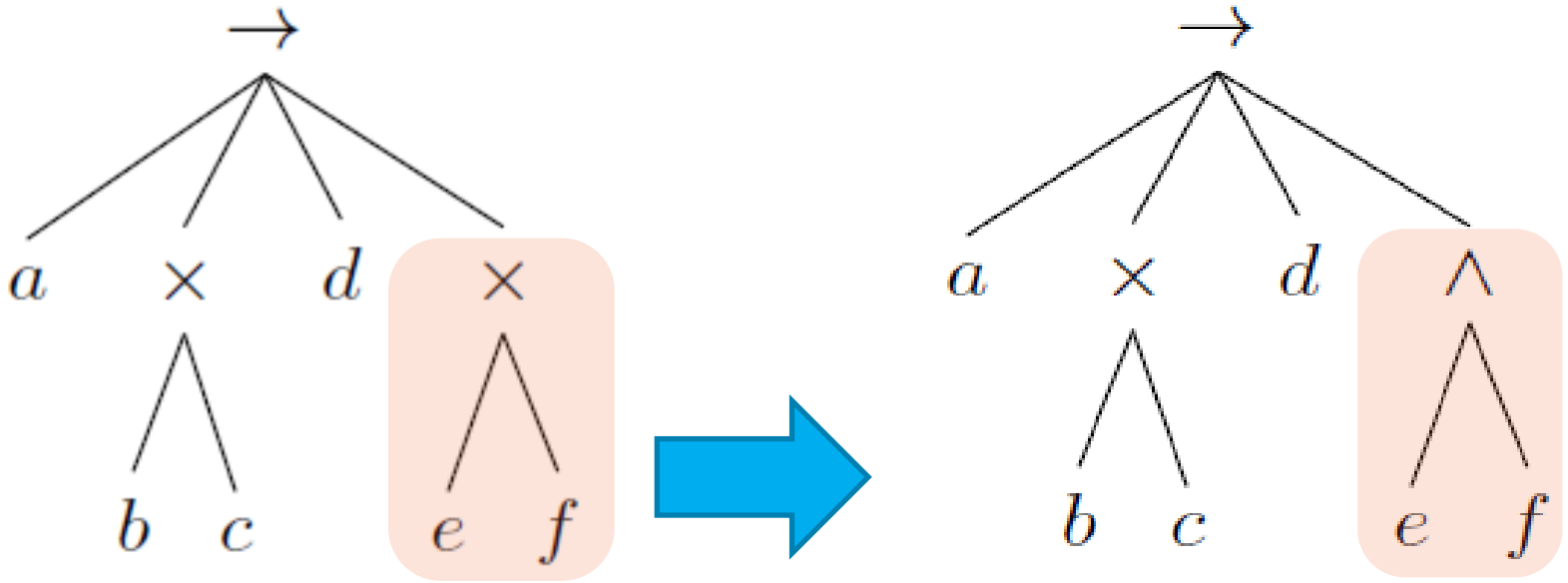
Reusing existing alignments (i)



trace	**a**	**b**	**m**	**d**	**e**	**f**
model	**a**	**b**	≫	**d**	**e**	≫

Reusing existing alignments (ii)

- **Change** (node 7) **from a XOR to an AND operator**

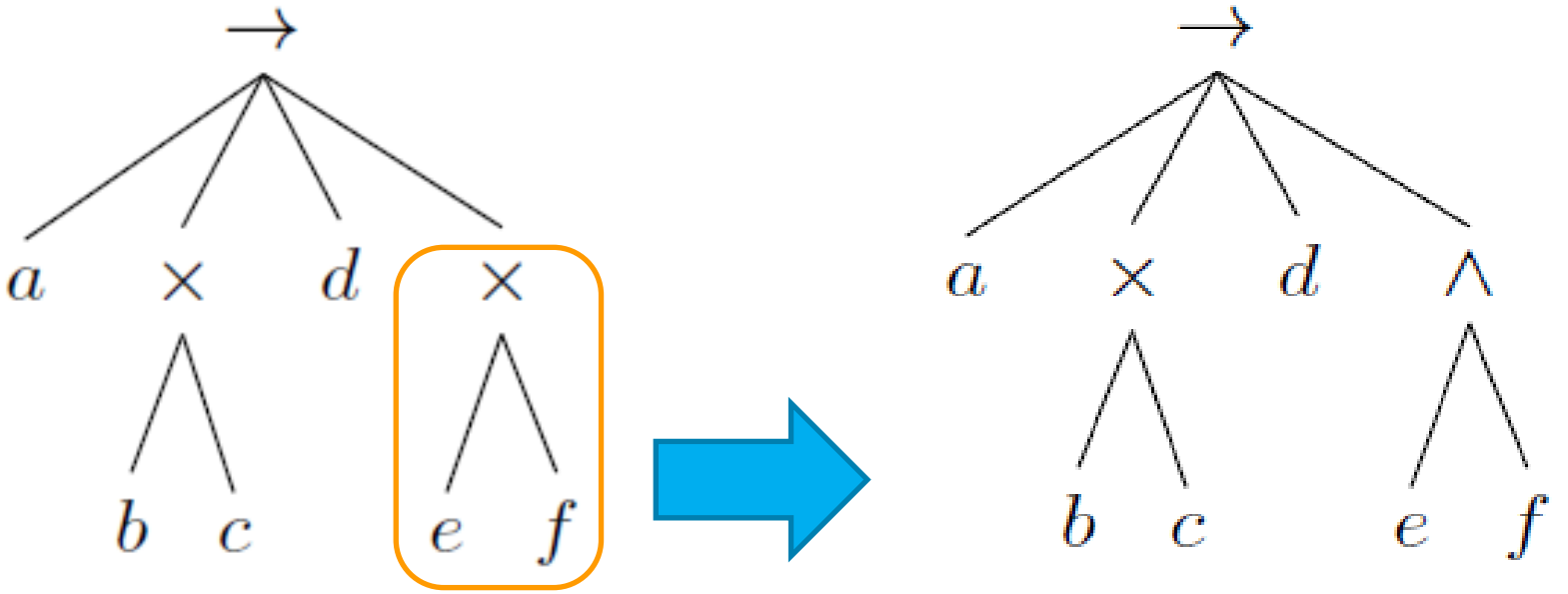


trace	**a**	**b**	**m**	**d**	**e**	**f**
model	**a**	**b**	≫	**d**	**e**	≫

trace	**…**
model	**…**

Reusing existing alignments (iii)

- **Scope of change: everything under node** × (node 7)

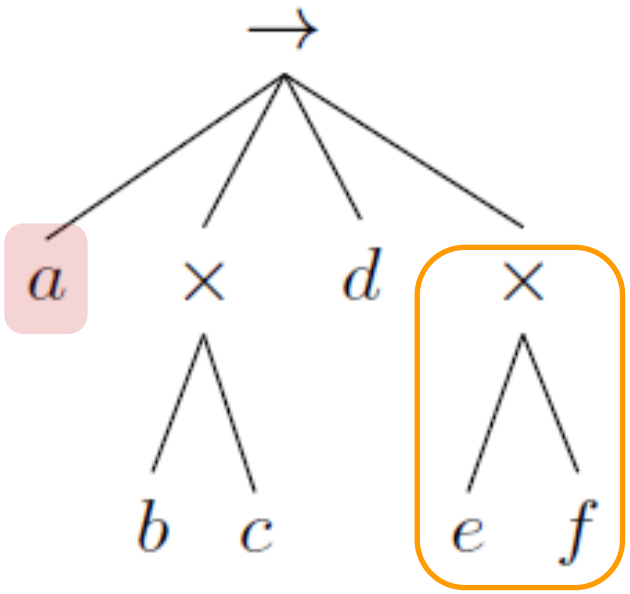


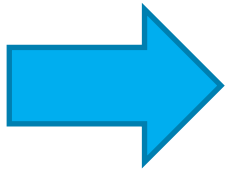
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

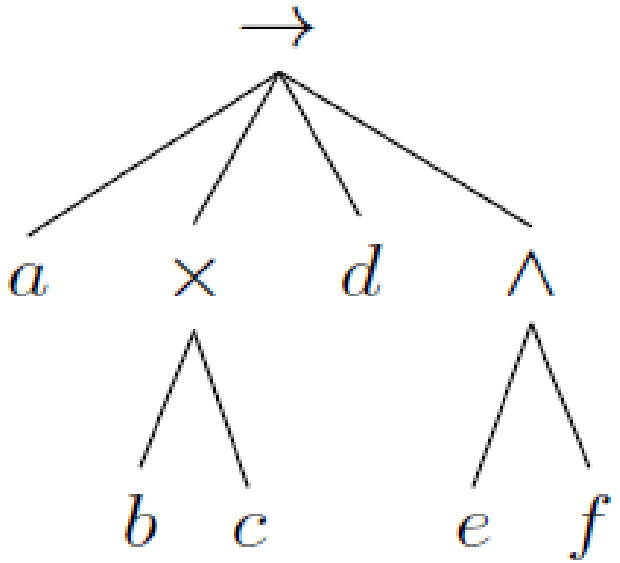
trace	…
model	…

Reusing existing alignments (iv)

- **Sync move outside the scope**





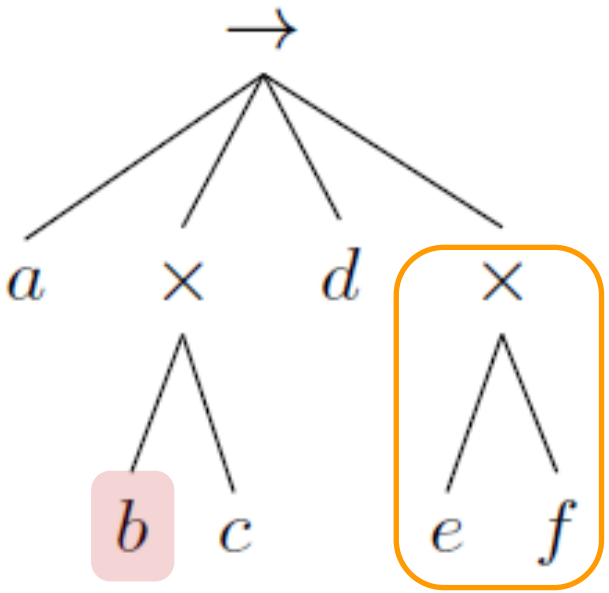


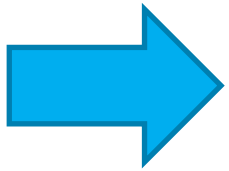
trace	**a**	**b**	**m**	**d**	**e**	**f**
model	**a**	**b**	≫	**d**	**e**	≫

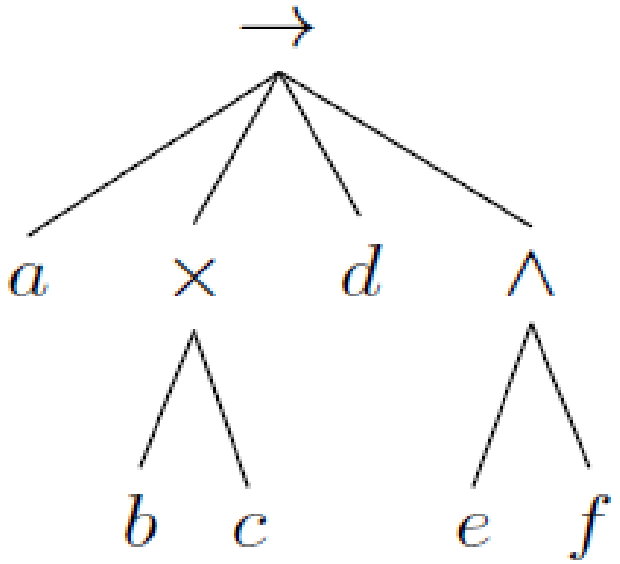
trace	**a**	…
model	**a**	…

Reusing existing alignments (v)

- **Sync move outside the scope**





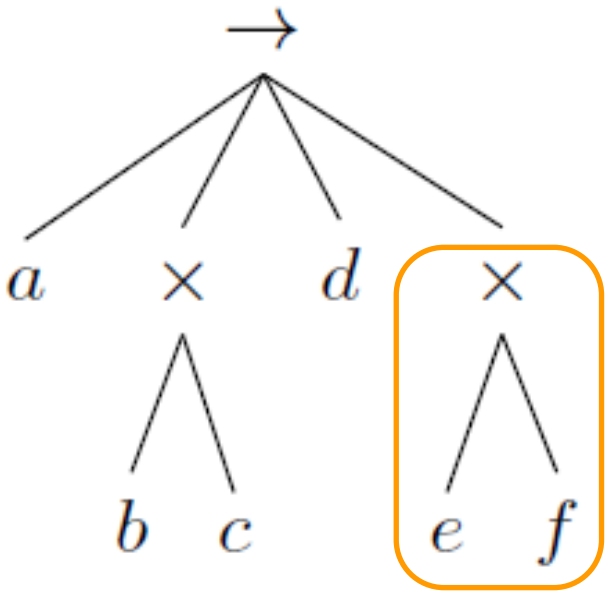


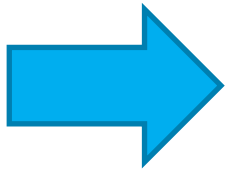
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

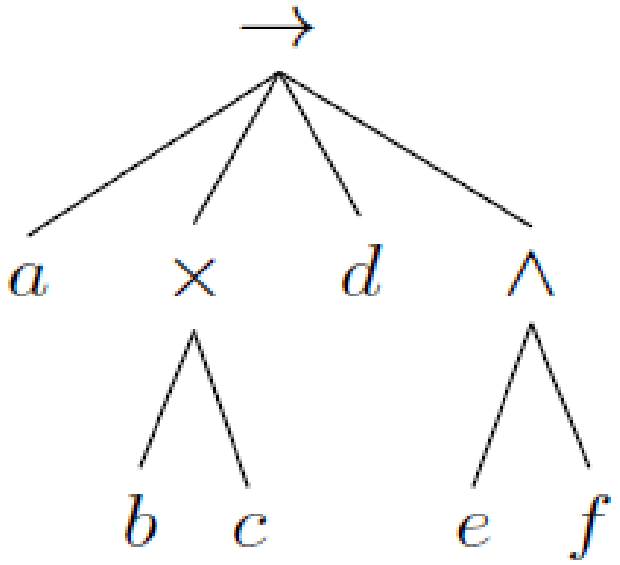
trace	a	b	…
model	a	b	…

Reusing existing alignments (vi)

- **Log move ? the scope:** check previous move.





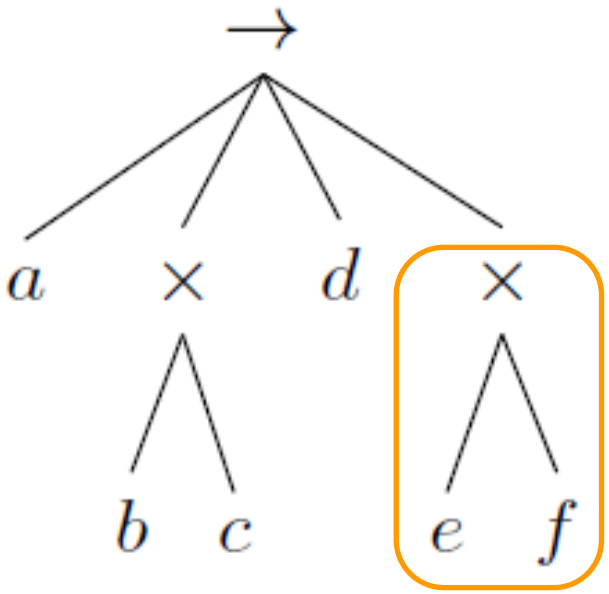


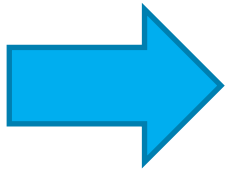
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

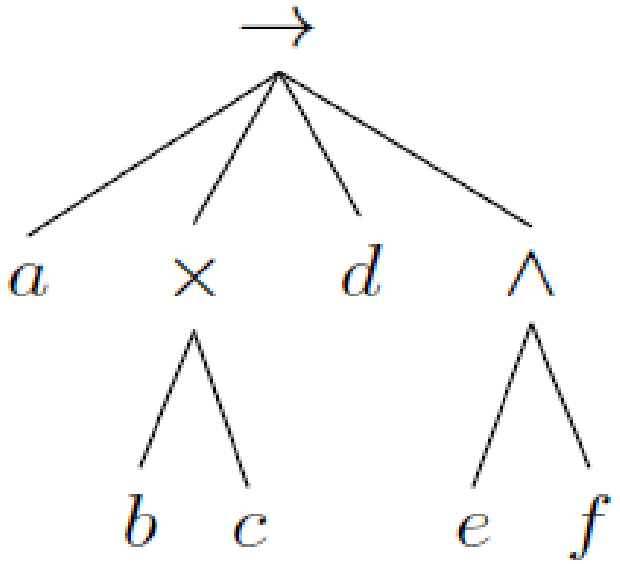
trace	a	b	…
model	a	b	…

Reusing existing alignments (vii)

- **Log move outside the scope**





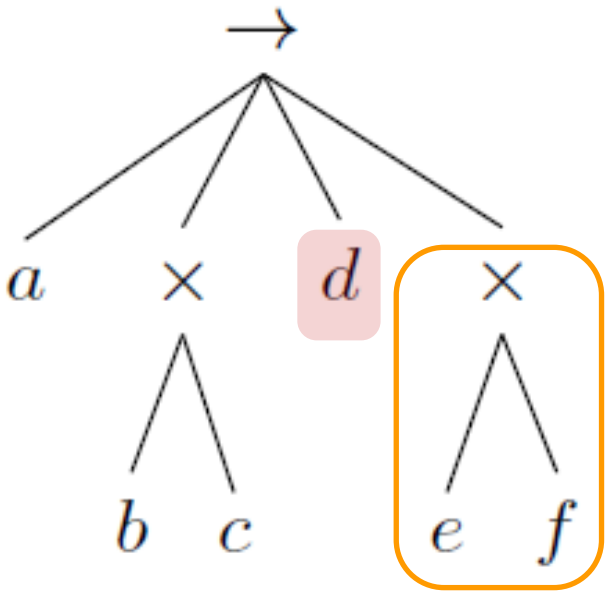


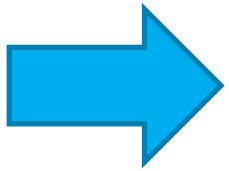
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

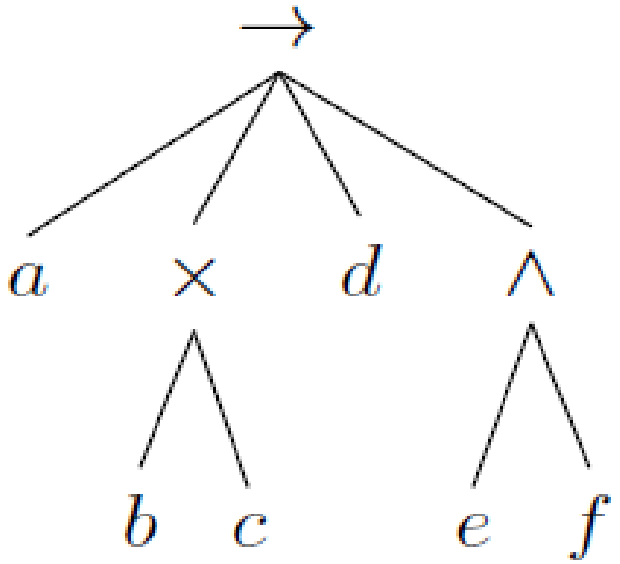
trace	a	b	m	…
model	a	b	≫	…

Reusing existing alignments (viii)

- **Sync move outside the scope**





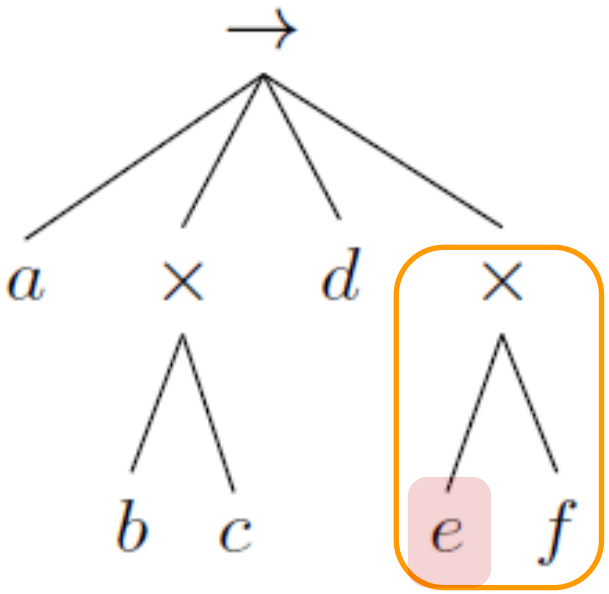


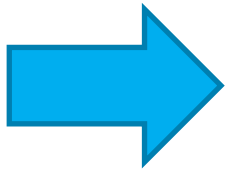
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

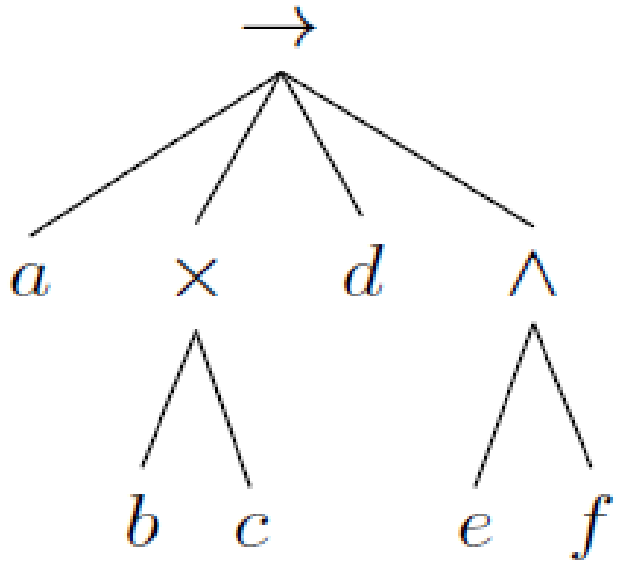
trace	a	b	m	d	…
model	a	b	≫	d	…

Reusing existing alignments (ix)

- **Sync move inside the scope:** new trace <e>





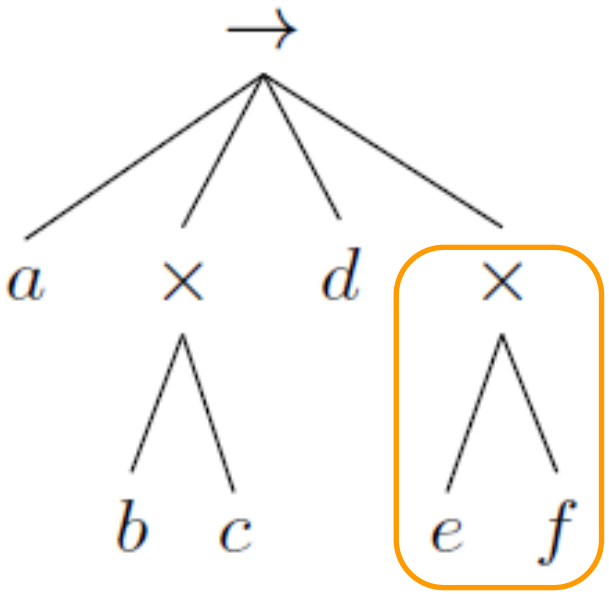


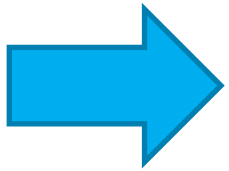
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

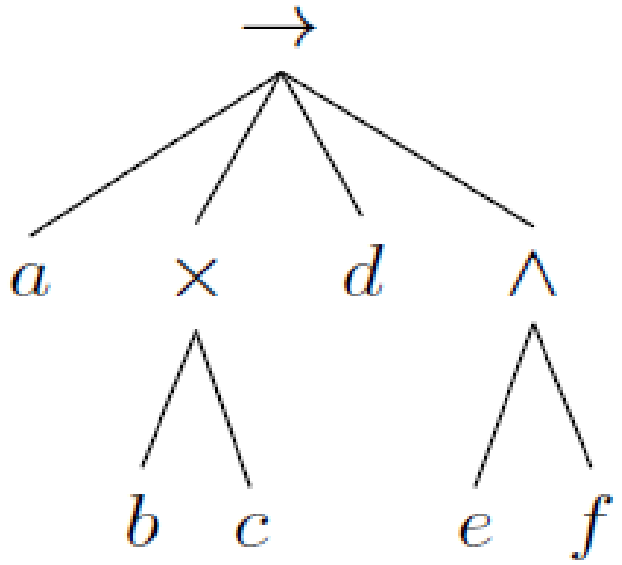
trace	a	b	m	d	e	…
model	a	b	≫	d	-	…

Reusing existing alignments (x)

- **Log move inside the scope:** add to new trace <e,f>





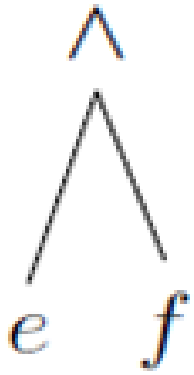


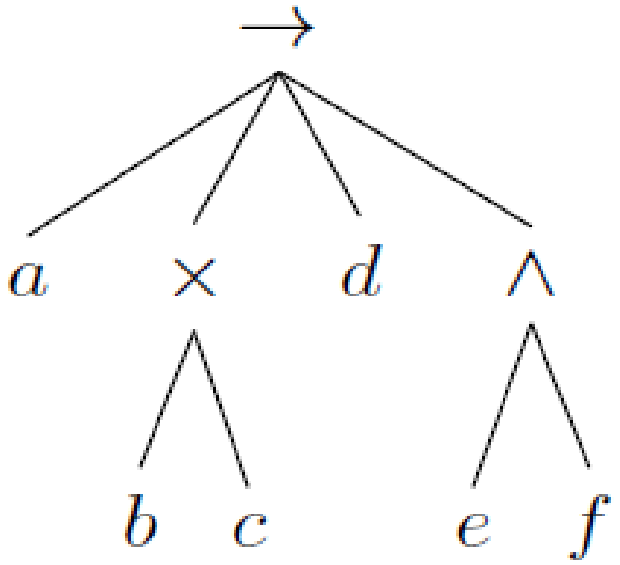
trace	a	b	m	d	e	f
model	a	b	≫	d	e	≫

trace	a	b	m	d	e	f
model	a	b	≫	d	-	-

Reusing existing alignments (xi)

- **Align the new subtree with the subtrace:** {<e,f>}



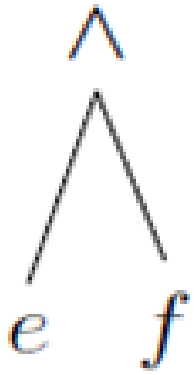


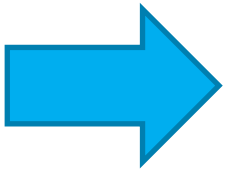
trace	**e**	**f**
model	**e**	**f**

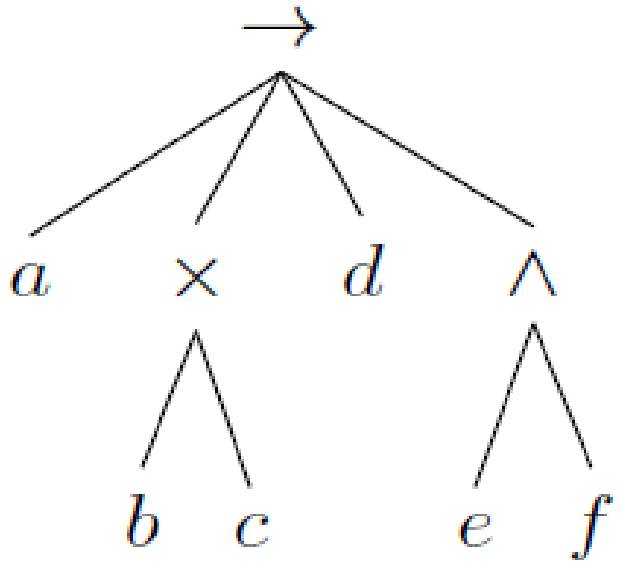
trace	**a**	**b**	**m**	**d**	**e**	**f**
model	**a**	**b**	≫	**d**	**-**	**-**

Reusing existing alignments (xii)

- **Align the new subtree with the new trace:** {<e,f>}



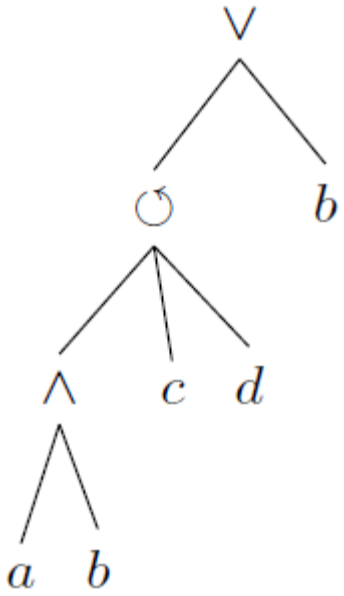




trace	**e**	**f**
model	**e**	**f**

trace	**a**	**b**	**m**	**d**	**e**	**f**
model	**a**	**b**	≫	**d**	**e**	**f**

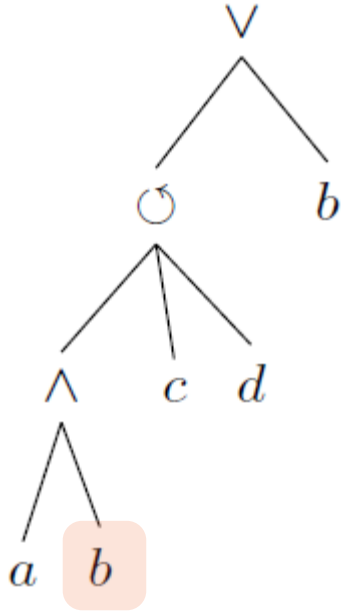
Reusing existing alignments (i)

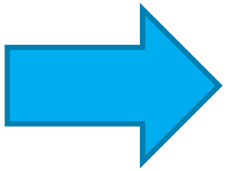


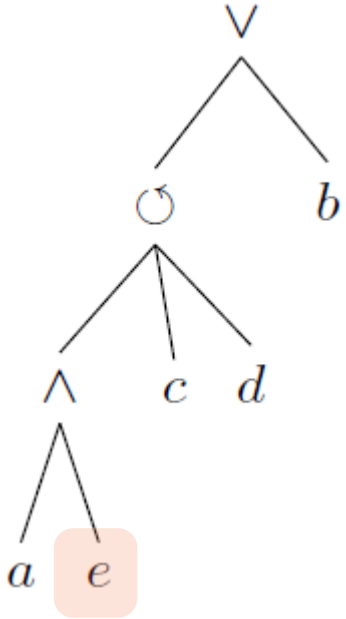
trace	**a**	**b**	**c**	**a**	**e**	**d**
model	**a**	**b**	**c**	**a**	≫	**d**

Reusing existing alignments (ii)

- **Change node 2 from a XOR to an AND operator**





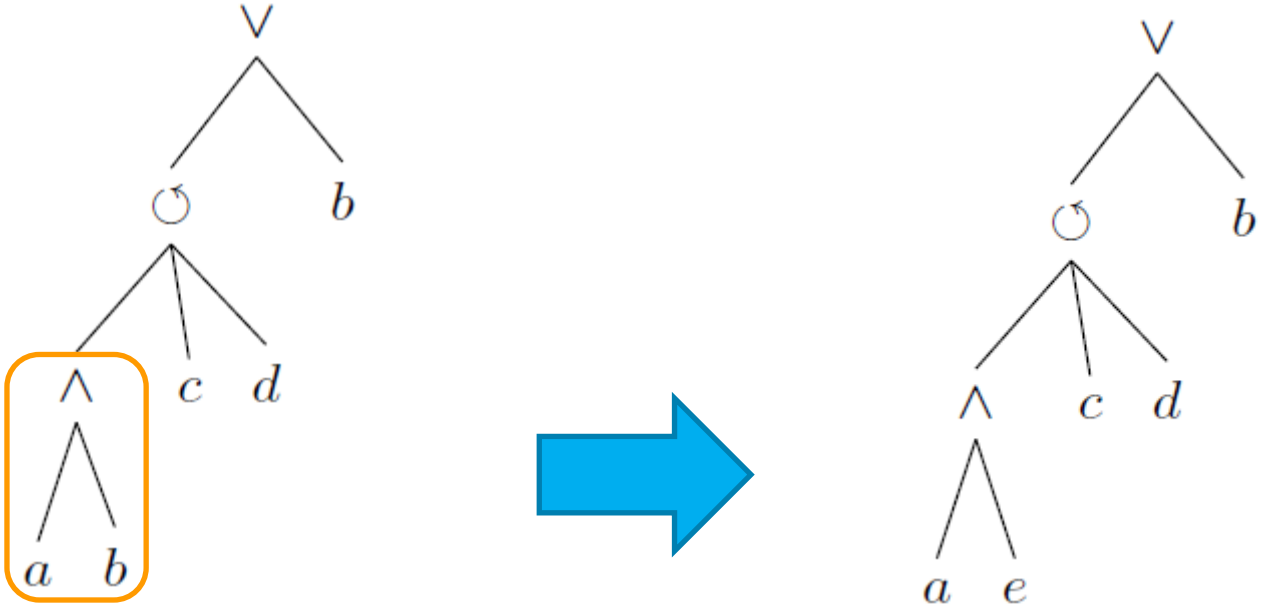


trace	**a**	**b**	**c**	**a**	**e**	**d**
model	**a**	**b**	**c**	**a**	≫	**d**

trace	…
model	…

Reusing existing alignments (iii)

- **Scope:** when changing a leaf, take the parent.

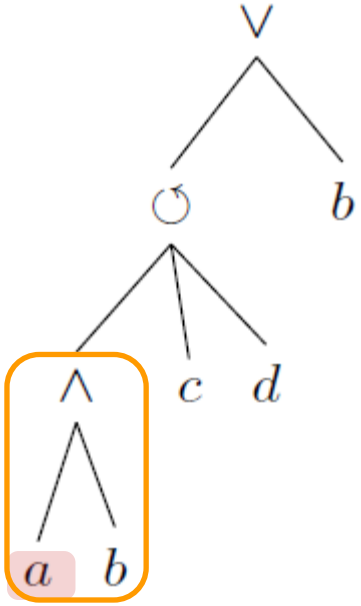


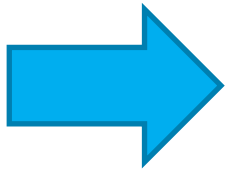
trace	**a**	**b**	**c**	**a**	**e**	**d**
model	**a**	**b**	**c**	**a**	≫	**d**

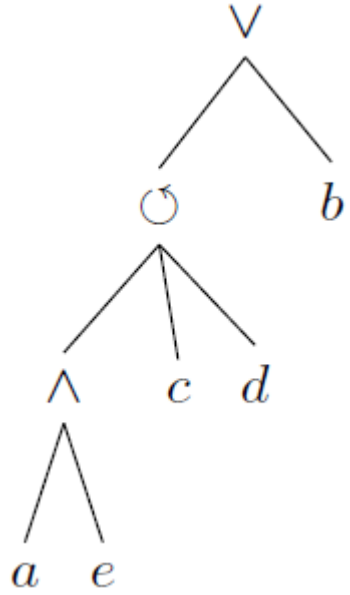
trace	**a**	…
model	**-**	…

Reusing existing alignments (iii)

- **Sync move inside the scope**: new trace <a>





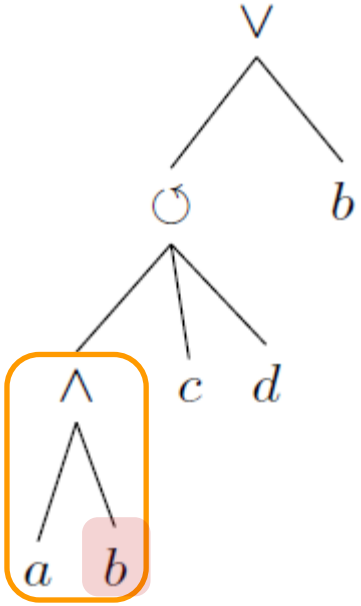


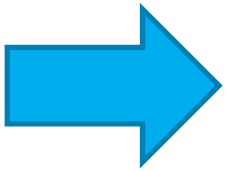
trace	**a**	**b**	**c**	**a**	**e**	**d**
model	**a**	**b**	**c**	**a**	≫	**d**

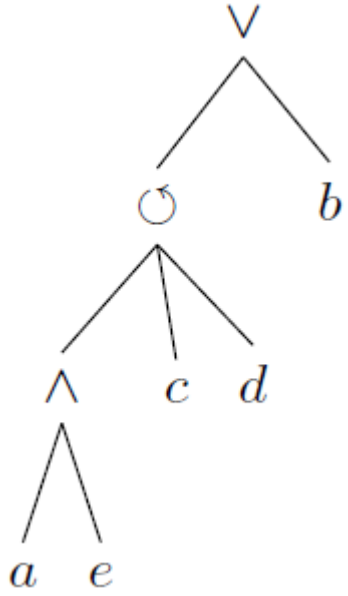
trace	**a**	…
model	**-**	…

Reusing existing alignments (iv)

- **Sync move inside the scope**: new trace <a,b>





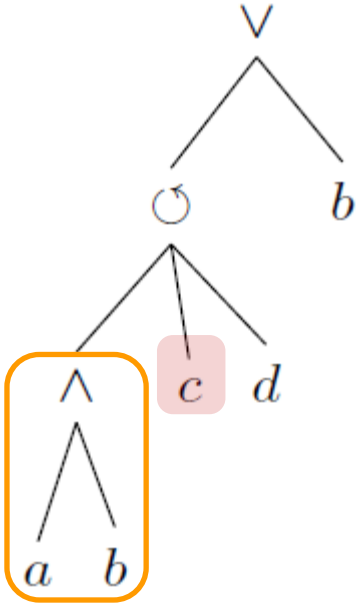


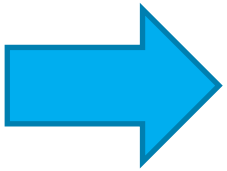
trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

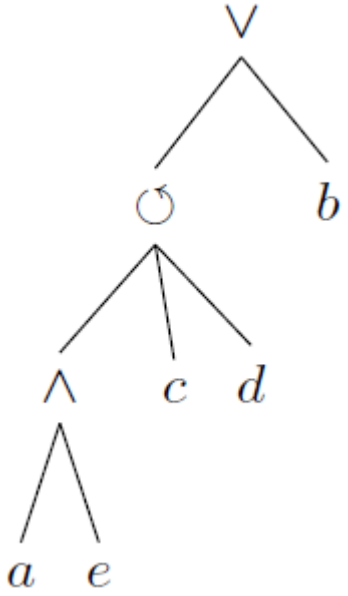
trace	a	b	…
model	-	-	…

Reusing existing alignments (v)

- **Sync move outside the scope**





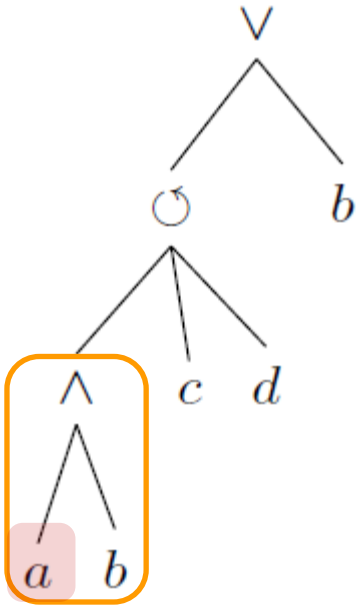


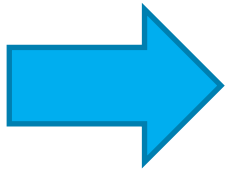
trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

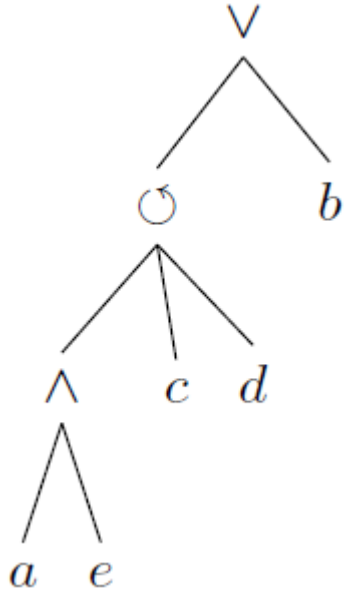
trace	a	b	c	…
model	-	-	c	…

Reusing existing alignments (vi)

- **Sync move inside the scope**





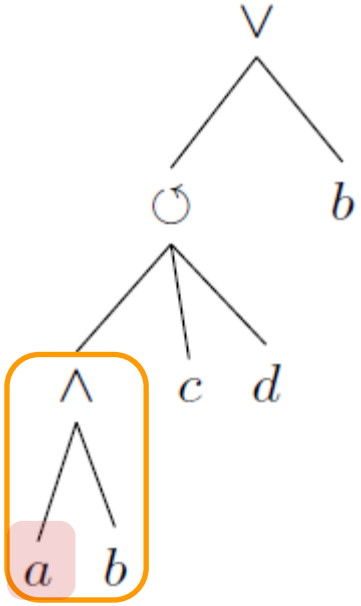


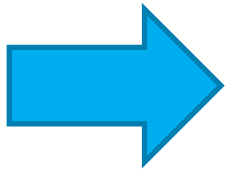
trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

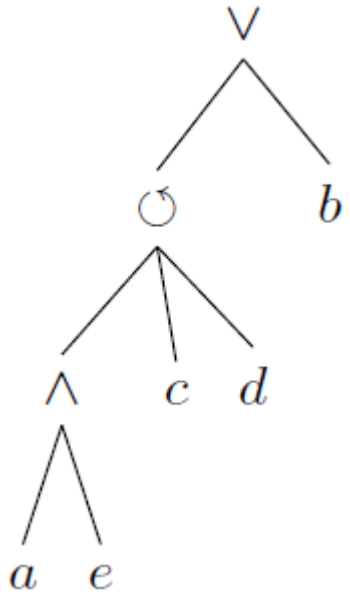
trace	a	b	c	…
model	-	-	c	…

Reusing existing alignments (vii)

- **Sync move inside the scope**
- **New instance of the loop:** create a new trace: <a>





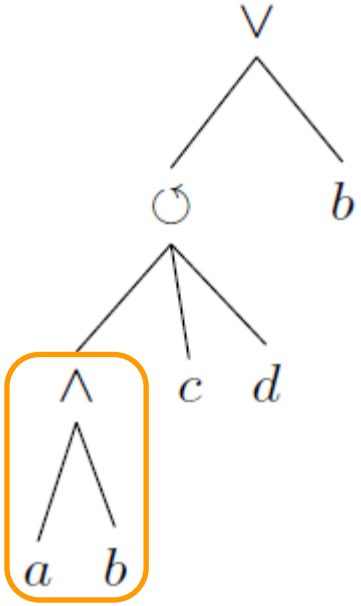


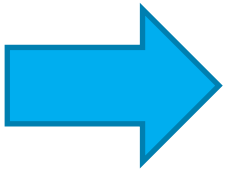
trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

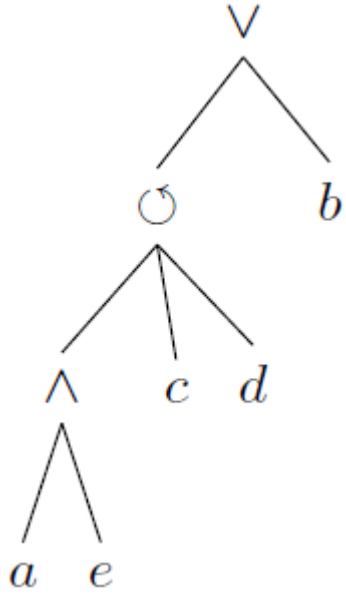
trace	a	b	c	a	…
model	-	-	c	-	…

Reusing existing alignments (viii)

- **Sync move inside the scope:** add to new trace <a,e>





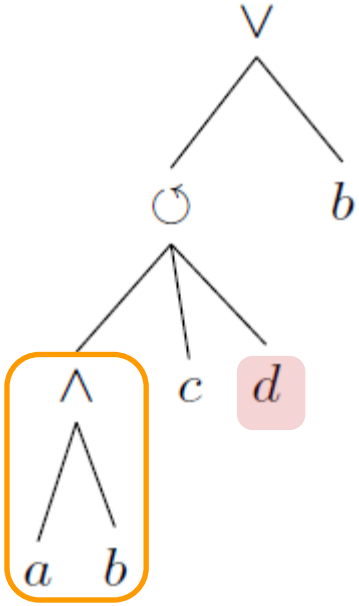


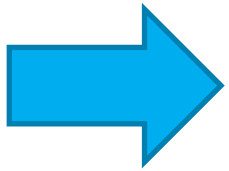
trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

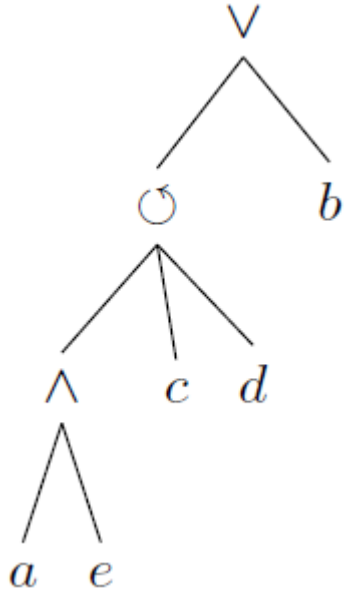
trace	a	b	c	a	e	…
model	-	-	c	-	-	…

Reusing existing alignments (ix)

- **Sync move outside the scope**





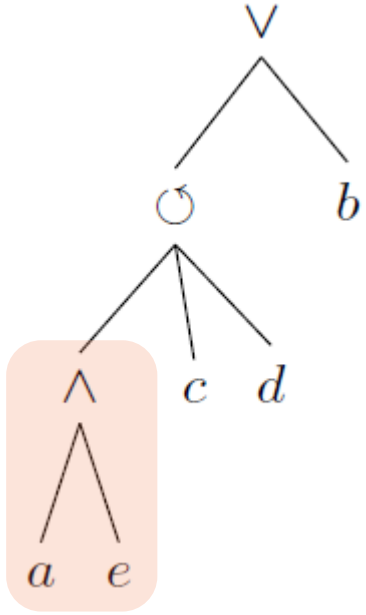


trace	a	b	c	a	e	d
model	a	b	c	a	≫	d

trace	a	b	c	a	e	d
model	-	-	c	-	-	d

Reusing existing alignments (x)

- **Align the new subtree with the new traces: {<ab>,<ae>}**



1st Part

trace	**a**	≫	**b**
model	**a**	**e**	≫

2nd Part

trace	**a**	**e**
model	**a**	**e**

trace	**a**	**b**	**c**	**a**	**e**	**d**
model	**-**	**-**	**c**	**-**	**-**	**d**

Reusing existing alignments (xi)

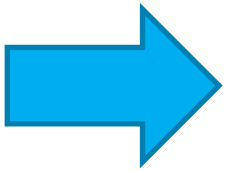
- **Align the new subtree with the new traces: {<ab>,<ae>}**

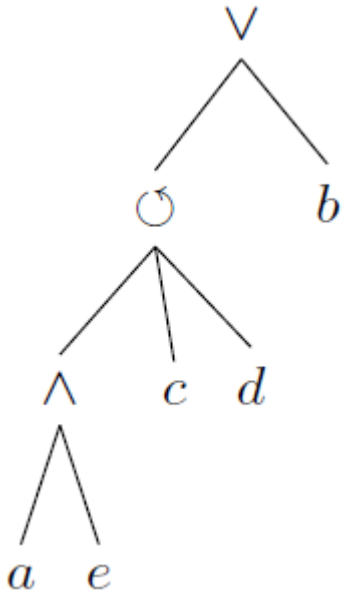
1st Part

trace	a	≫	b
model	a	e	≫

2nd Part

trace	a	e
model	a	e





trace	a	≫	b	c	a	e	d
model	a	e	≫	c	-	-	d

Reusing existing alignments (xii)

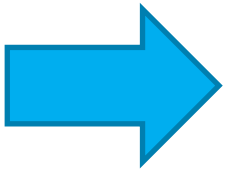
- **Align the new subtree with the new traces: {<ab>,<ae>}**

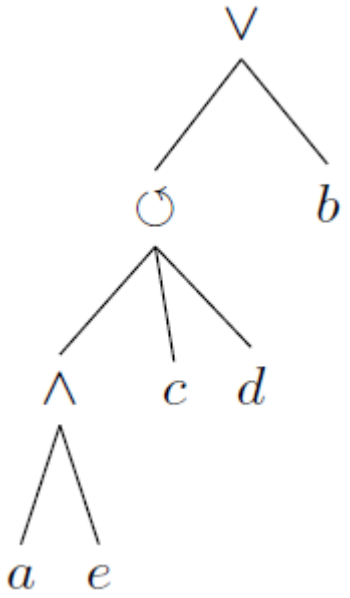
1st Part

trace	a	≫	b
model	a	e	≫

2nd Part

trace	a	e
model	a	e





trace	a	≫	b	c	a	e	d
model	a	e	≫	c	a	e	d

Reusing existing alignments (xiii)

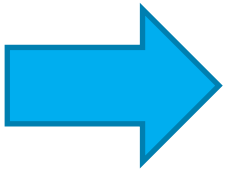
The resulting alignment is **not optimal**

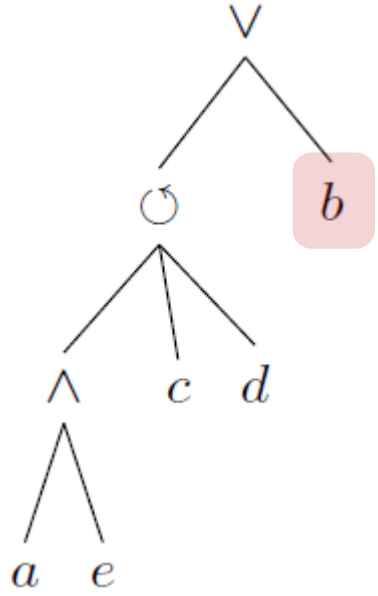
1st Part

trace	**a**	≫	**b**
model	**a**	**e**	≫

2nd Part

trace	**a**	**e**
model	**a**	**e**





trace	**a**	≫	**b**	**c**	**a**	**e**	**d**
model	**a**	**e**	≫	**c**	**a**	**e**	**d**

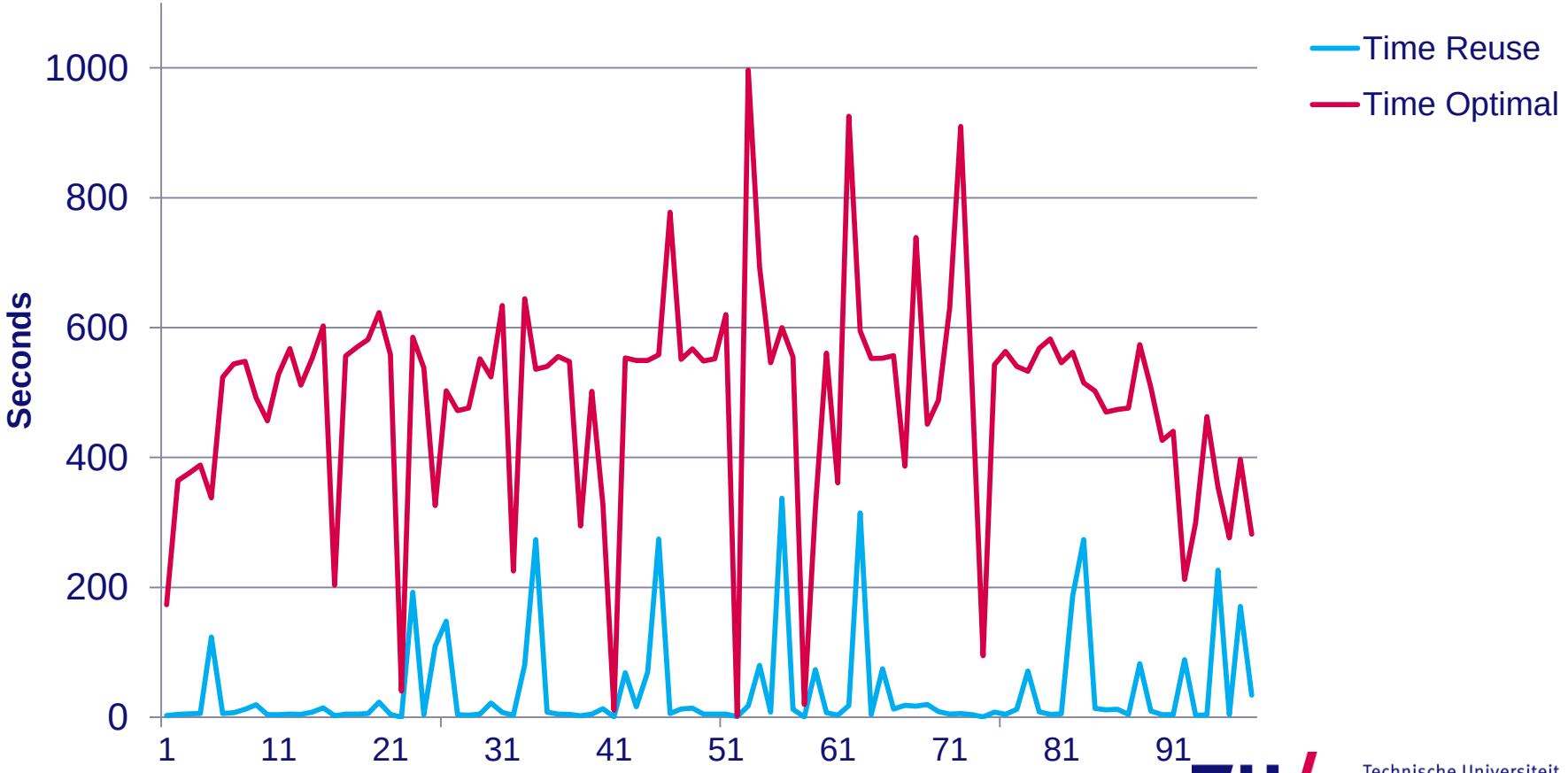
Experiments – Set up

1. **Random tree: >20** nodes
   - **Random log:** 2000 traces

2. **100 mutations of the original random tree**
   - **Time comparison:**
     - Optimal alignment vs Reusing the old alignment
   - **Quality of the new alignment**

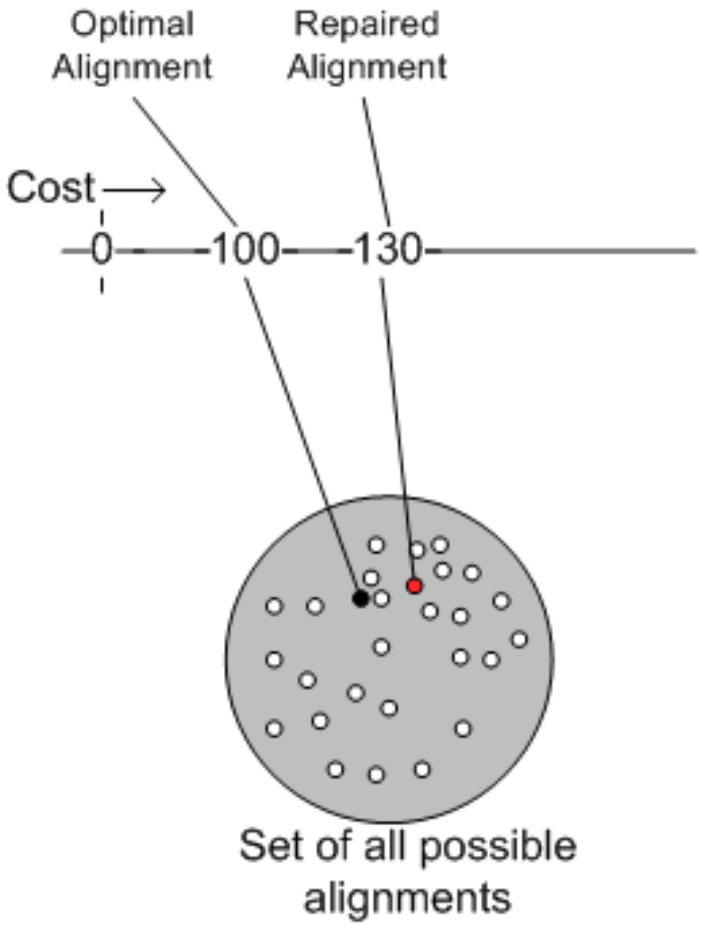
Experiments – Results (Time)

100 mutations of the original tree



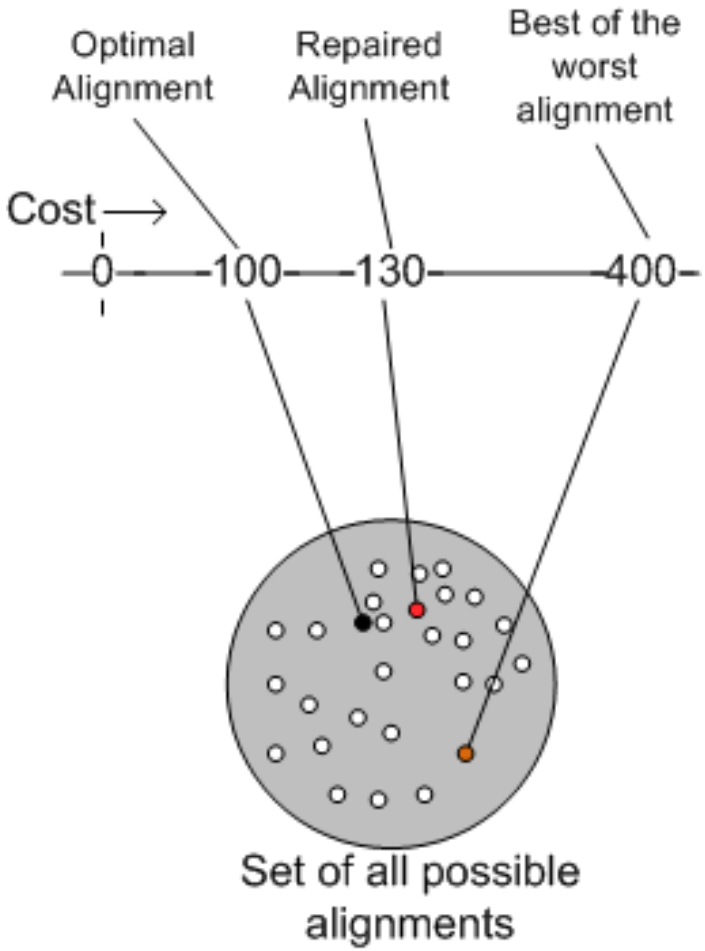
Experiments – Results (Quality)

How to grade the new alignment



Experiments – Results (Quality)

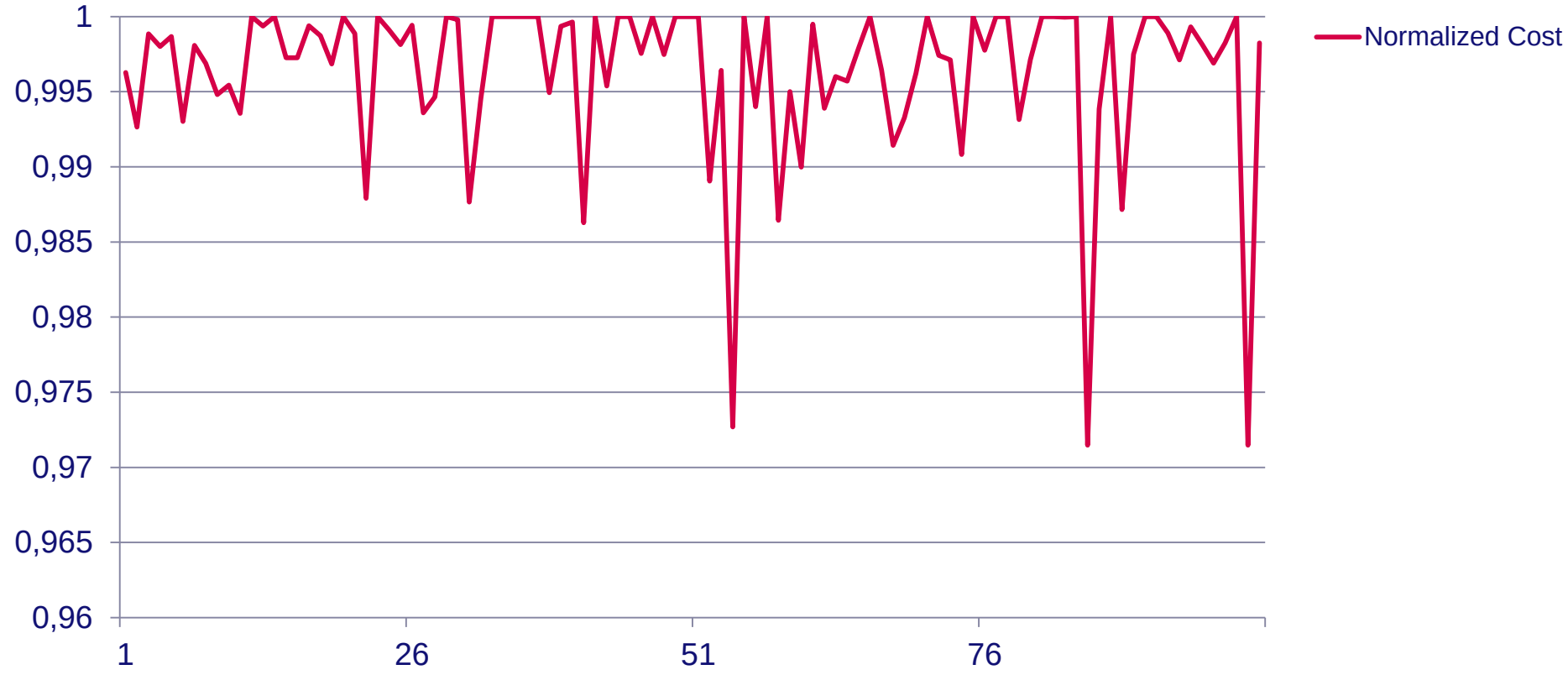
How to grade the new alignment



- **Normalize it based on:**
  - **The optimal alignment**
  - **"Best of the worst" alignment**

trace	**a**	**e**	≫	≫
model	≫	≫	**a**	**e**

Experiments – Results (Quality)



Future work

- **Predict optimality**
  - When is it better to compute the optimal alignment?
  - How many times can we keep reusing an alignment?

- **Use cases:**

  - **ETM**

  - **Streaming**

Reusing Alignments

Borja Vázquez- Barreiros
Sebastiaan van Zelst
Joos Buijs

Technische Universiteit
Eindhoven
University of Technology

Where innovation starts