POLYTECHNIQUE MONTRÉAL

SOftware Cost-effective Change and Evolution Research Lab

Ptidej

Pattern Trace Identification, Detection, and Enhancement in Java

CONCEPTS EXTRACTION FROM EXECUTION TRACES

Soumaya Medini

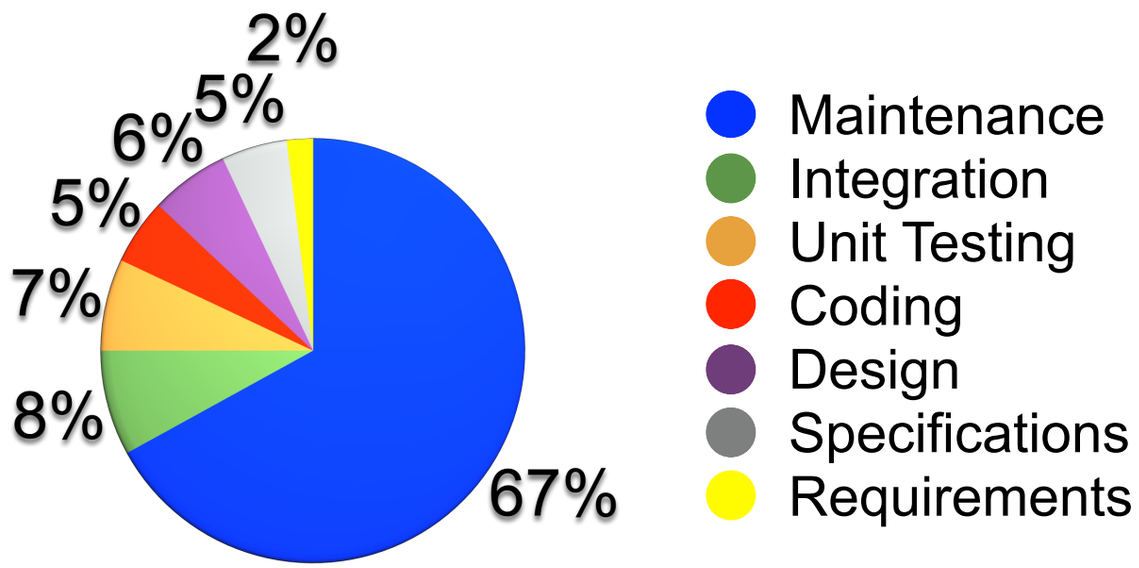
Ph.D. Defense

Montréal, November 17th 2014

Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work

Context



Software Life-Cycle Costs

- Software maintenance effort is estimated to be more than 70% of the overall software cost. [Ian Sommerville, 2000]
- Program comprehension require half of the effort devoted to software maintenance and evolution. [Dehaghani et Hajrahimi, 2013]

Context

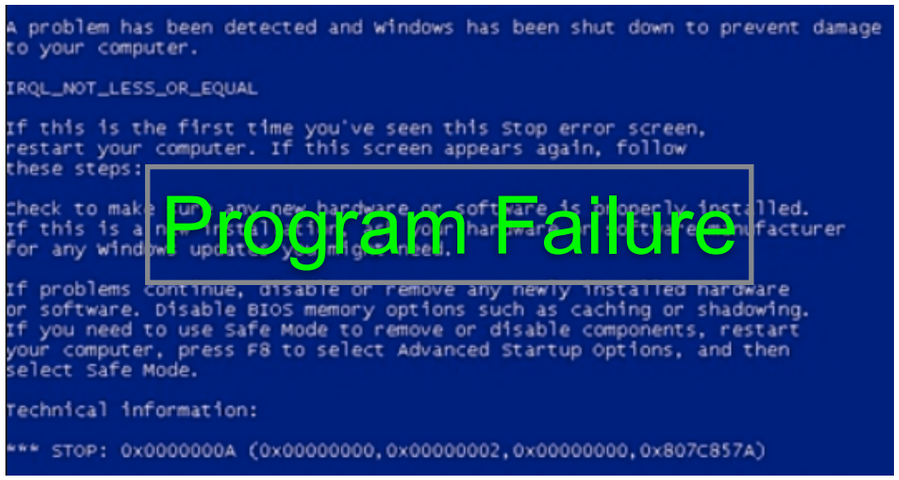
- Understand a program: identify which concept this program implements.
- Concept location aims at identifying concepts and locating them within code regions.
- A concept represents a functionality of a program.

Motivation

- A typical scenario in which concept location takes part:

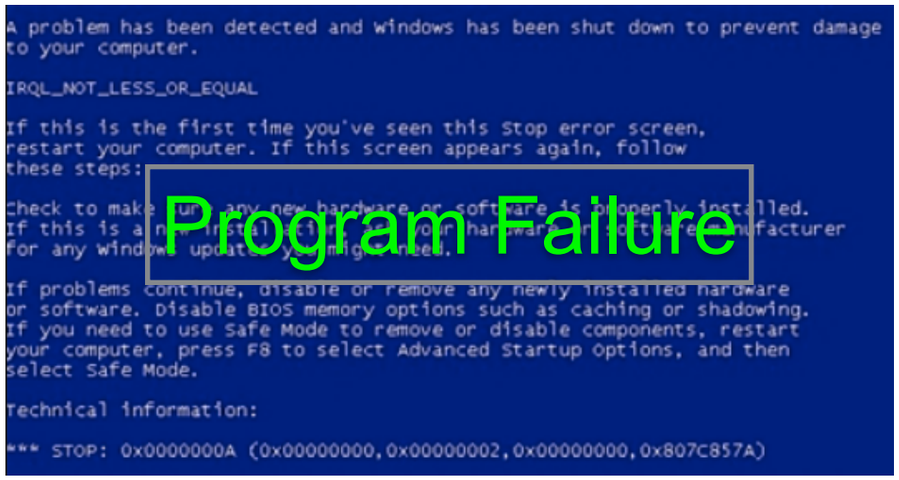
Motivation

- A typical scenario in which concept location takes part:



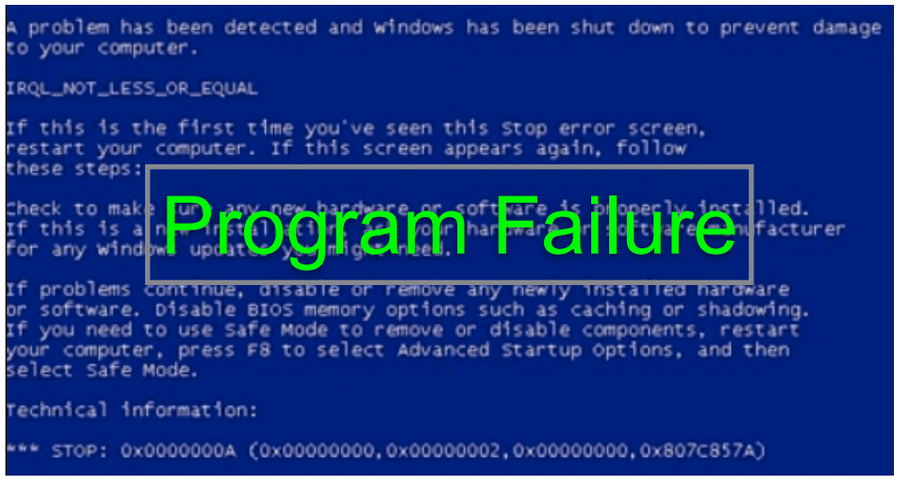
Motivation

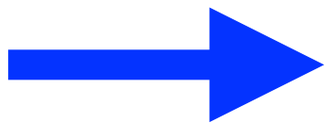
- A typical scenario in which concept location takes part:



Motivation

- A typical scenario in which concept location takes part:

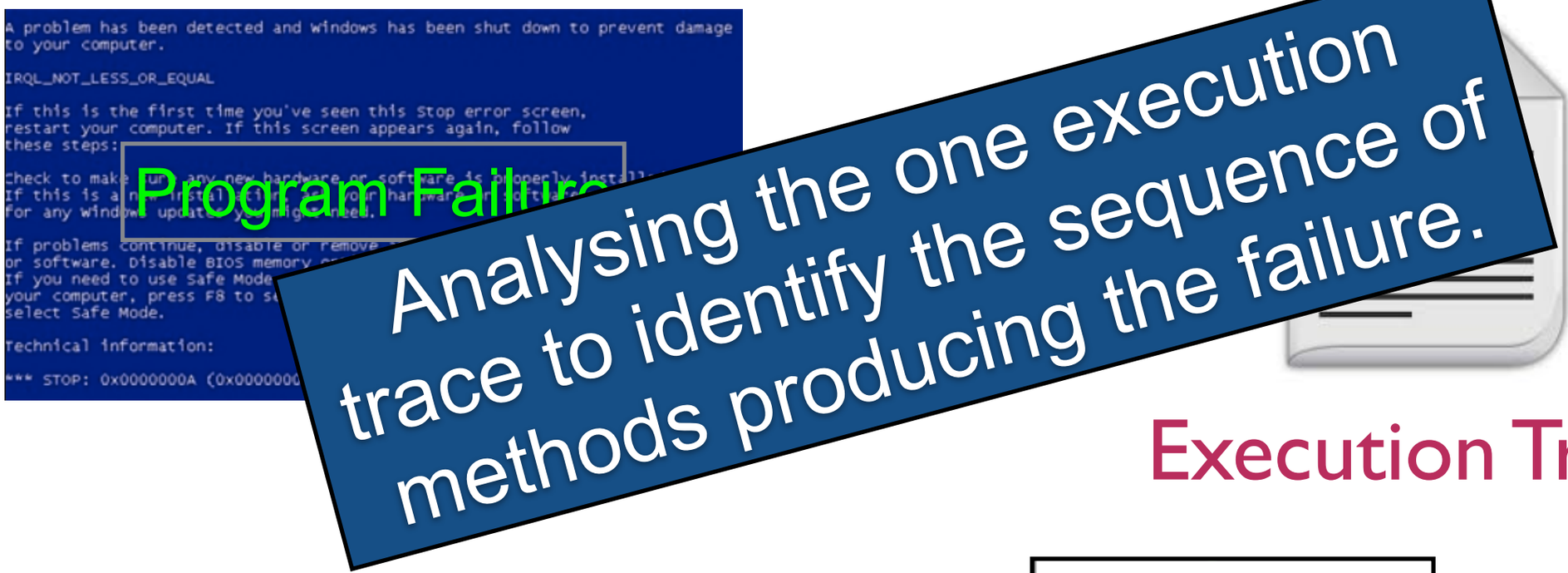


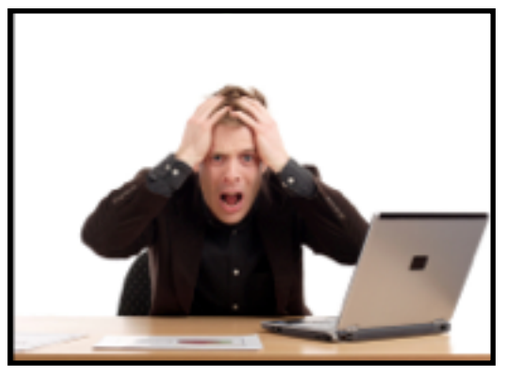


Execution Trace

Motivation

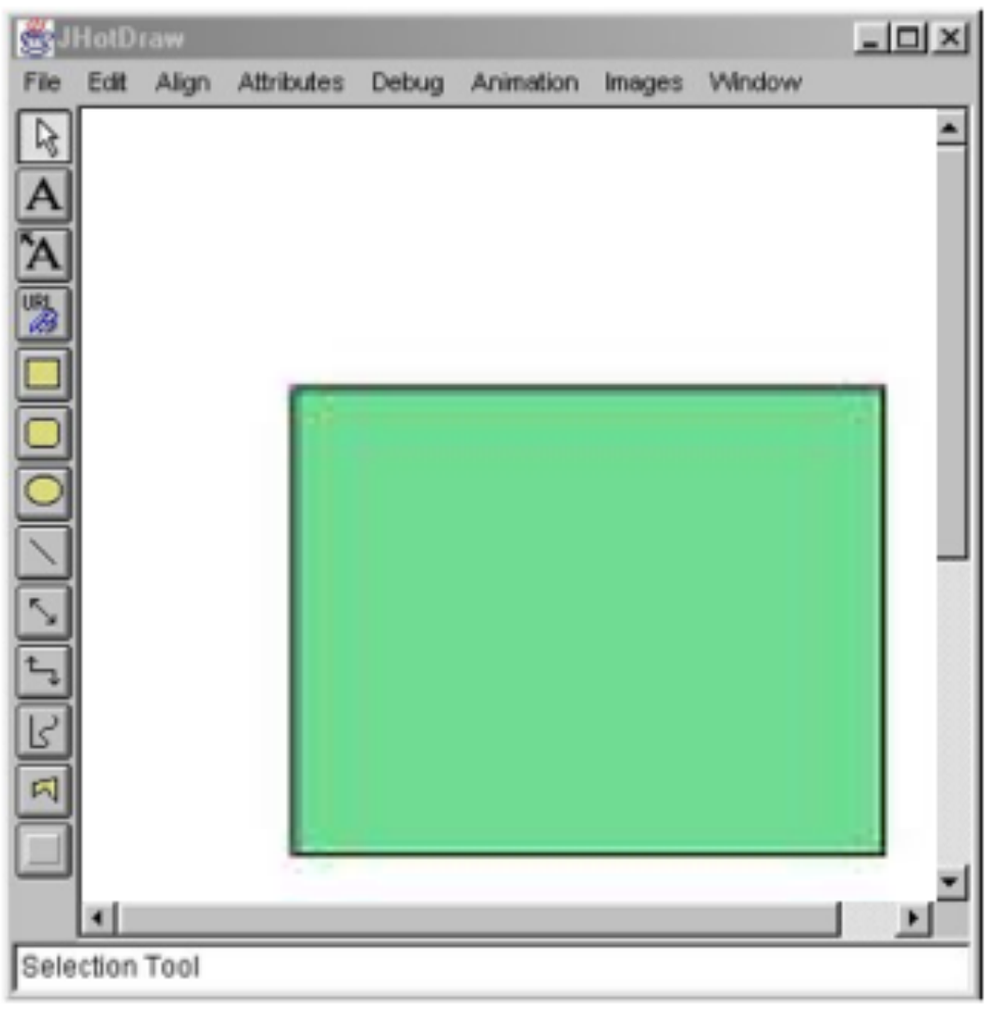
- A typical scenario in which concept location takes part:





Problem Statement

- Large and noisy:
  - Execution trace corresponding to draw a rectangle in JHotDraw contains 4,000 method calls.



Problem Statement

- Several approaches address these problems:
  - Compacting execution traces
    (encoding the whole execution as a directed acyclic graph)
    [Reiss and Renieris, 2001]
  - Building high-level behavioural models
    (detecting and filtering utilities) [Hamou-Lhadj et al., 2005]
  - Segmenting execution traces
    (textual analysis or clustering algorithms) [Asadi et al., 2010] [Pirzadeh and Hamou-Lhadj, 2011]

Problem Statement

- Several approaches address these problems:
  - Compacting execution trace
    (encoding the whole execution
    [Reiss and Renieris, 2001]
  - Build
  - Abstract traces
    (clustering algorithms) [Asadi et al., 2010] [Pirzadeh and

None of these approaches guide developers towards segments that implements the concepts to maintain.

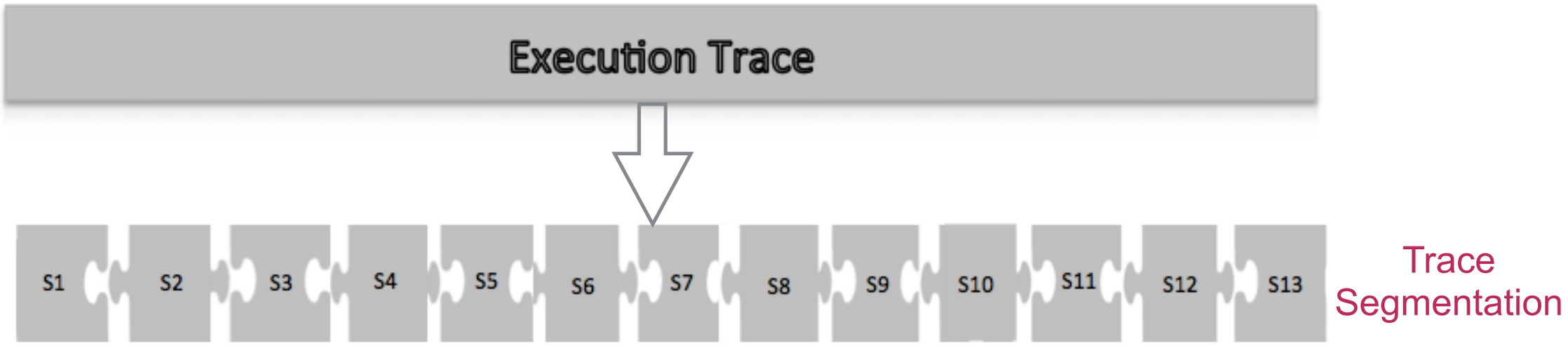
Thesis

Identify concepts and facilitate the analysis of large execution traces for maintenance tasks.

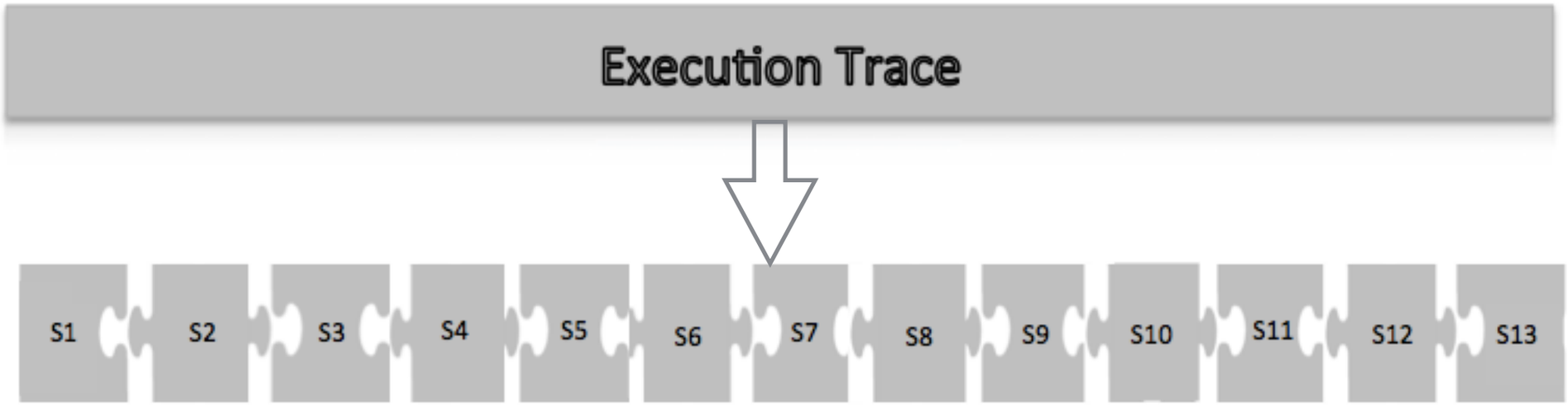
Thesis

Execution Trace

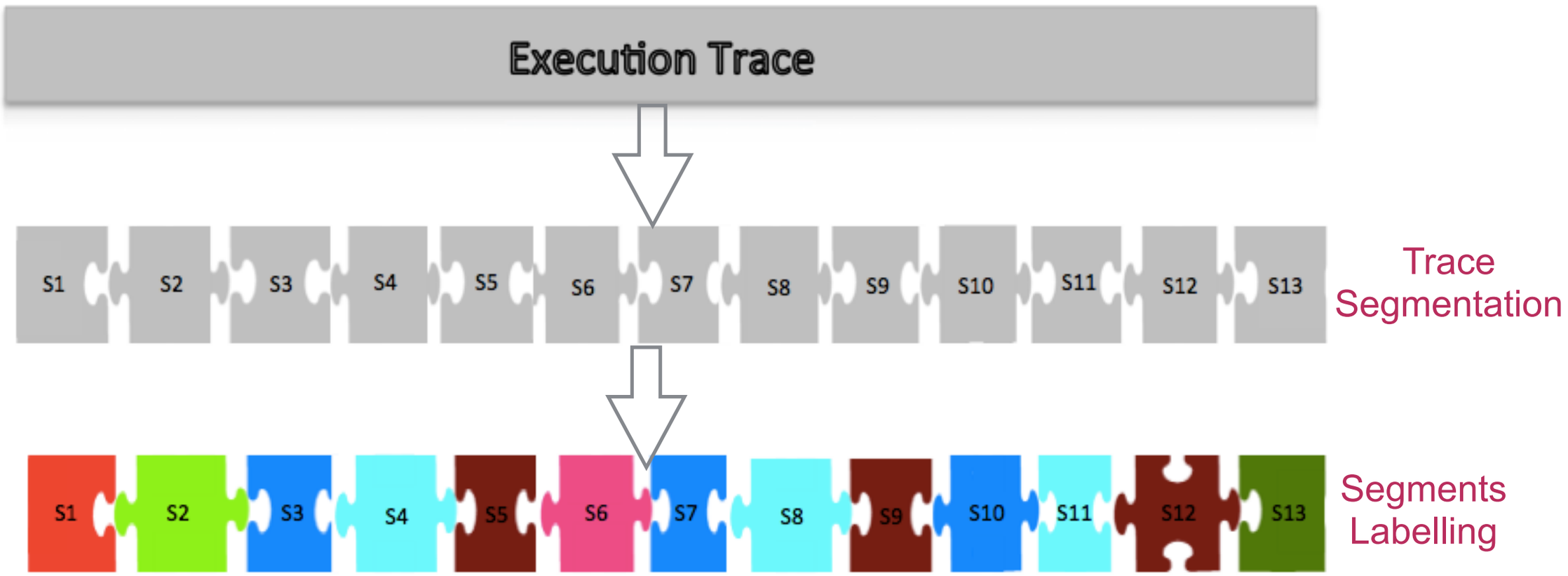
Thesis


Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation

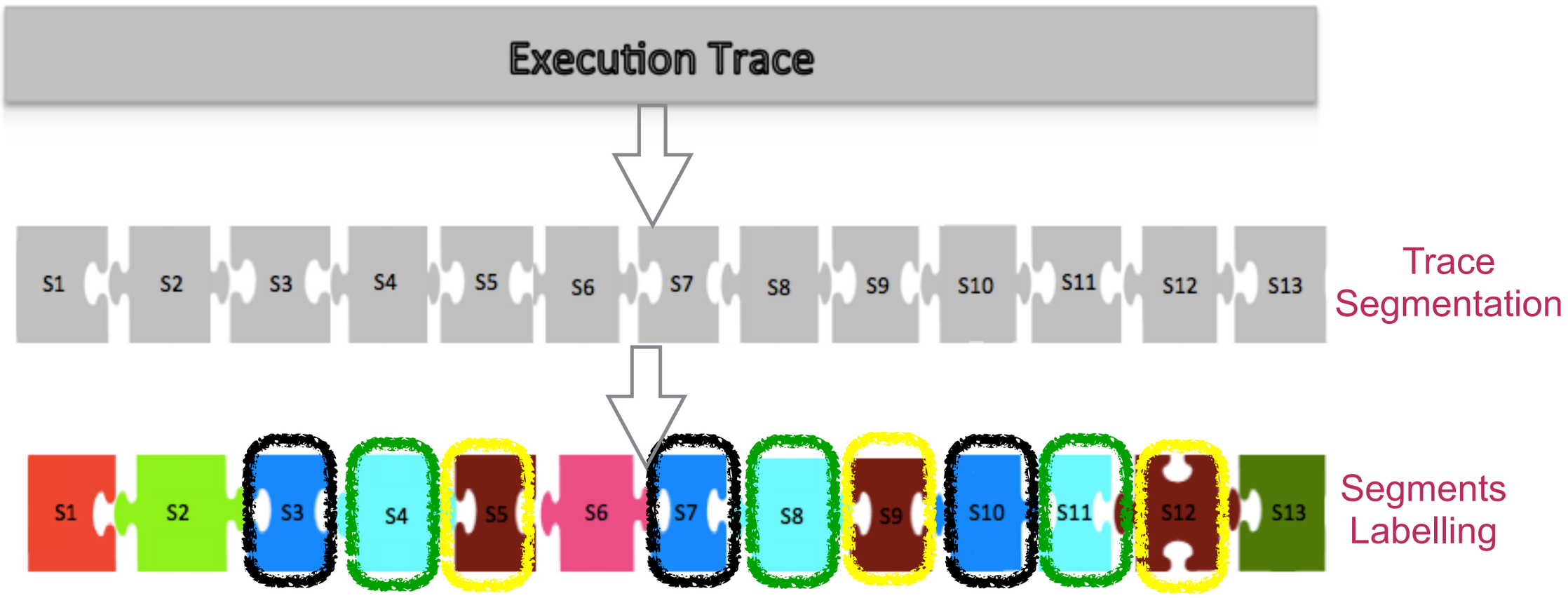
Thesis


Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation

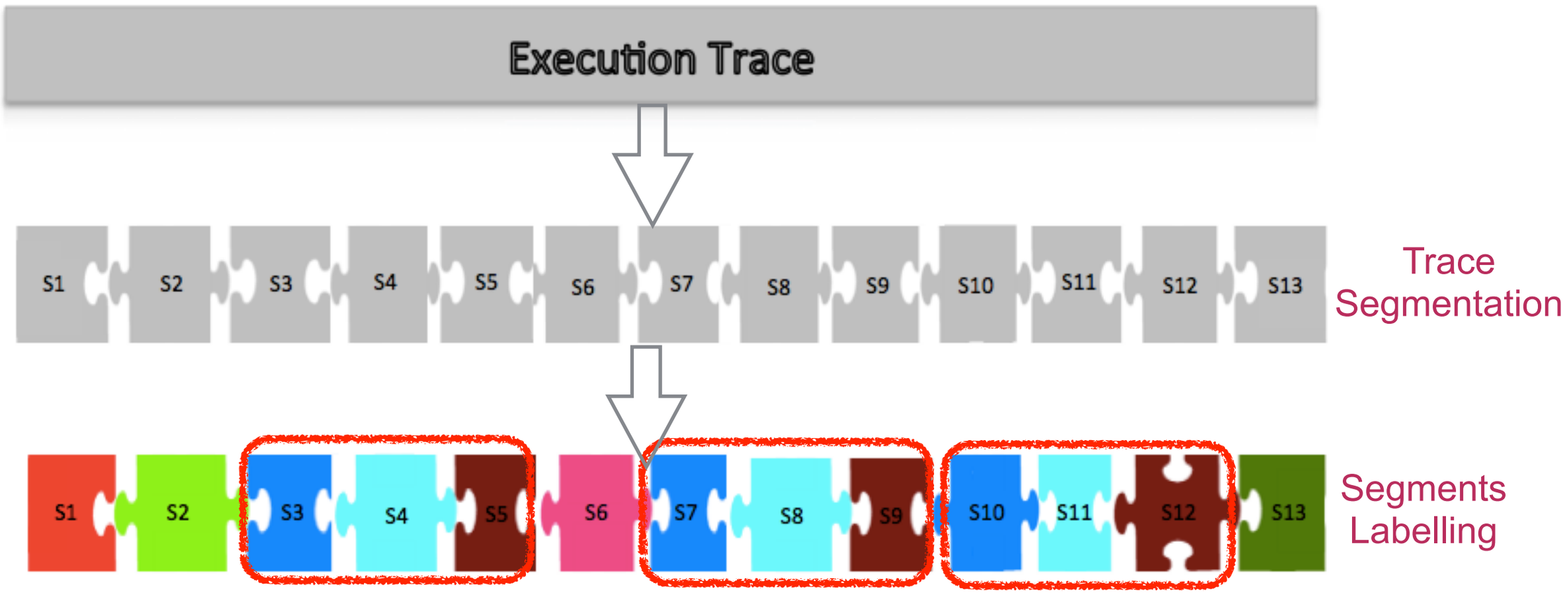
Thesis


Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Segments Labelling

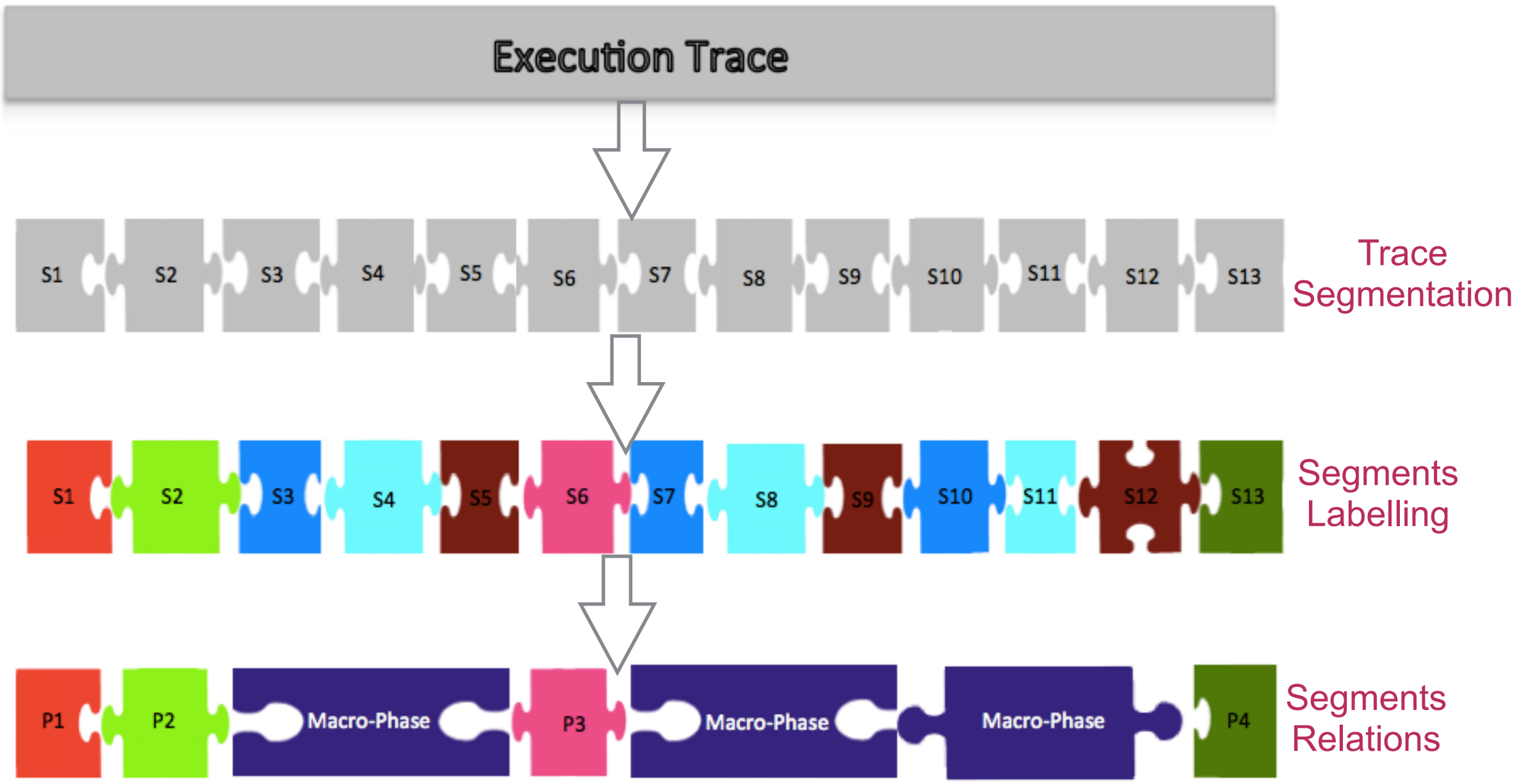
Thesis

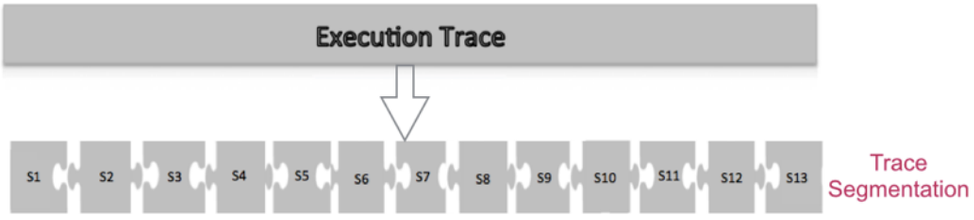

Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Segments Labelling

Thesis


Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Segments Labelling

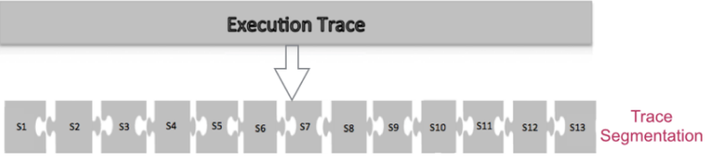
Thesis


Execution Trace
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Segments Labelling
P1
P2
Macro-Phase
P3
Macro-Phase
Macro-Phase
P4
Segments Relations



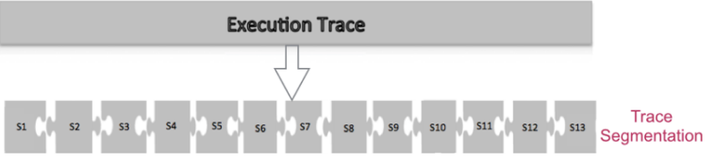
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work



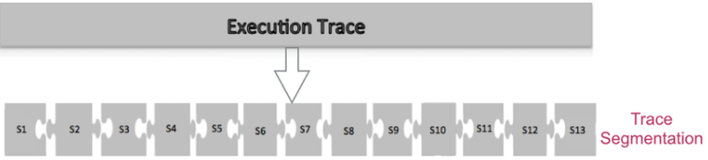
Trace Segmentation

- Asadi et al. [2010]: identify concepts in execution trace by finding cohesive and decoupled fragments of the trace using Genetic Algorithm (GA).
- Limitations:
  - Not scalable (7 hours).
  - Stability problems (different segmentation).



Background

- Steps:
  1. System instrumentation and trace collection;
  2. Pruning and compressing traces;
  3. Textual analysis of method source code;
  4. Trace splitting using optimization techniques.

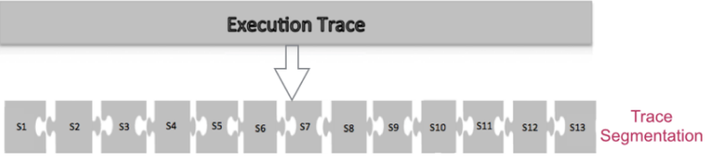


Step1: Program instrumentation and trace collection

- We collect and tag traces.

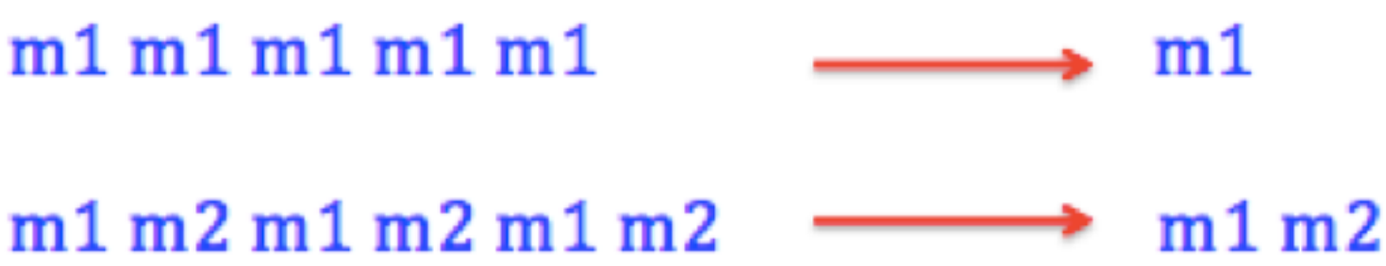
Trace.txt

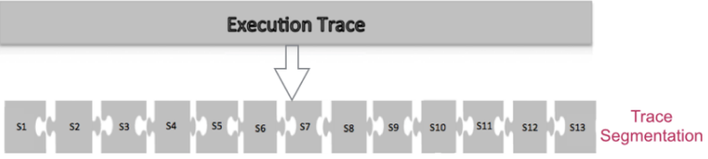
```
CH.ifa.draw.application.DrawApplication.open()
CH.ifa.draw.util.Iconkit.Iconkit(Component)
CH.ifa.draw.application.DrawApplication.createDrawingView()
# MARKER START: Draw Rectangle
CH.ifa.draw.standard.StandardDrawingView.tool()
CH.ifa.draw.application.DrawApplication.tool()
CH.ifa.draw.standard.StandardDrawingView.tool()
CH.ifa.draw.application.DrawApplication.tool()
CH.ifa.draw.standard.AbstractTool.mouseMove(MouseEvent-int-int)
CH.ifa.draw.standard.StandardDrawingView.mouseMoved(MouseEvent)
……
……
……
CH.ifa.draw.standard.ToolButton.paintBackground(Graphics)
CH.ifa.draw.standard.ToolButton.paintNormal(Graphics)
CH.ifa.draw.util.PaletteIcon.normal()
# MARKER STOP: Draw Rectangle
CH.ifa.draw.application.DrawApplication.tool()
```



Step2: Pruning and compressing traces

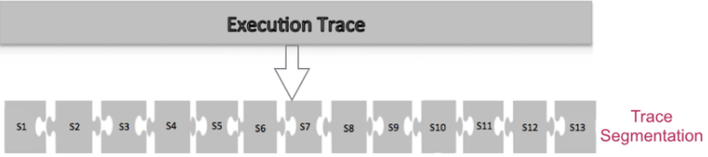
- Pruning: Remove too frequent method invocations.
- Compressing: Remove repetitions.





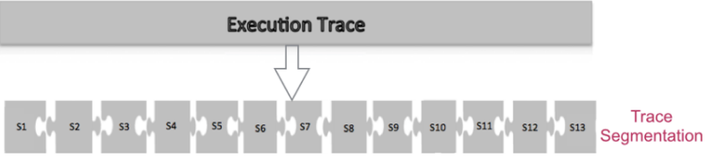
Step3: Textual analysis of Method source code

- Extract identifiers from source code and comments.
- Split identifiers using Camel-Case (getBook ⟶ get and book).
- Perform stemming (waited,waiting,waits ⟶ wait).
- Remove programming language keywords and english stop words.
- Index terms and documents using the *TF-IDF* indexing mechanisms and apply *LSI*.



Step4: Trace Splitting through optimization techniques

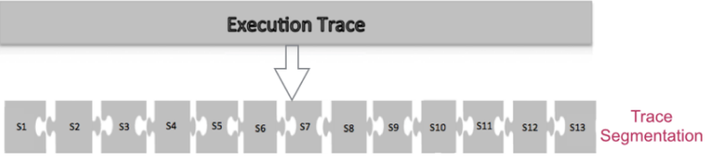
- Execution trace segmentation solution must be found in large search spaces.
- We must apply some optimization techniques to segment the trace.
- Approach built upon a dynamic programming algorithm to:
  - Improve scalability;
  - Compute the exact splitting.



Dynamic Programming (DP) Approach

- Solve a problem by dividing the problem into sub-problems that are recursively solved.
- The solution of the problem: combining the solutions of the sub-problems.
- The quality of the segmentation of a trace into K segments:

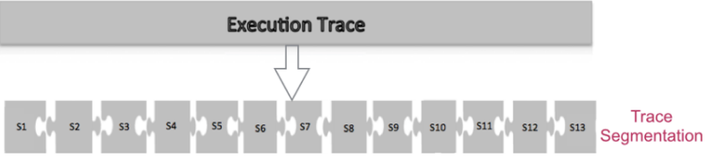
$$fit(segmentation) = \frac{1}{K} \times \sum_{i=1}^{K} \frac{COH_i}{COU_i + 1}$$



Cohesion and Coupling

- Cohesion

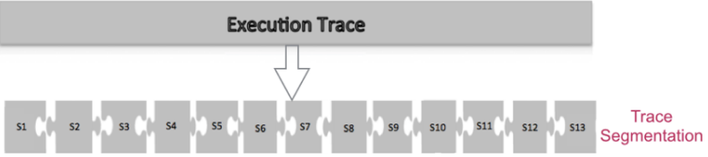
S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



Cohesion and Coupling

- Cohesion

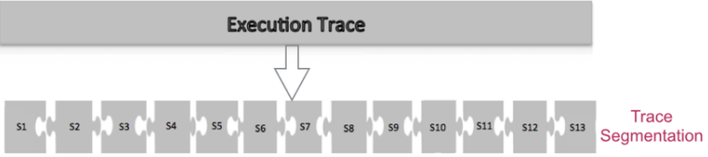
S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



Cohesion and Coupling

- Cohesion

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



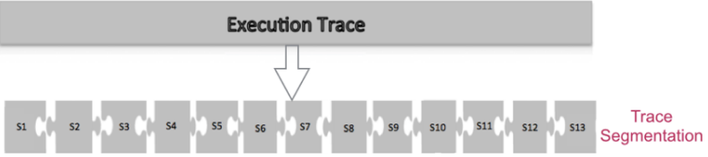
Cohesion and Coupling

- Cohesion

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

- Coupling

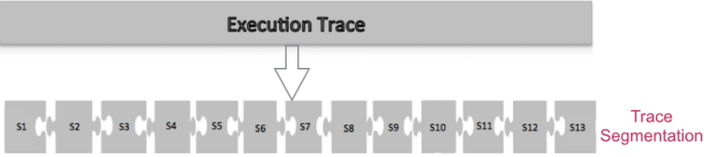
S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



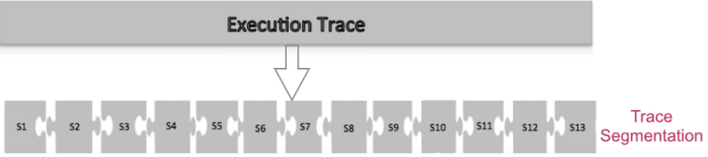
Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

  - Create a new segment.

S1				S2	S3	S4			S5			S6		
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.

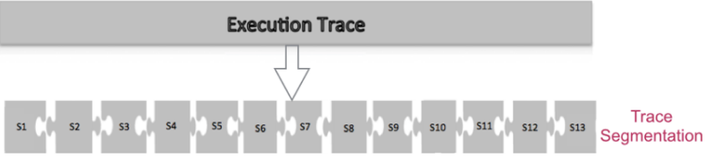
S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

  - Create a new segment.

S1				S2	S3	S4			S5			S6		
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

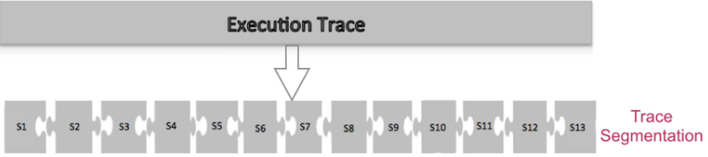
  - Add the method to the last segment.

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445



Case Study Design

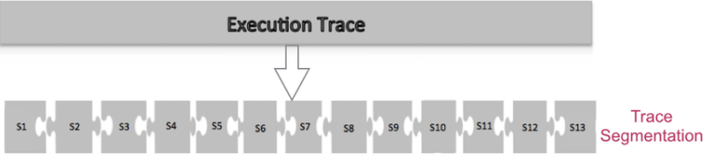
- Research Questions:
  - RQ1: How do the performances of the GA and DP approaches compare?
  - RQ2: How do the GA and DP approaches perform?
- Programs:



Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?

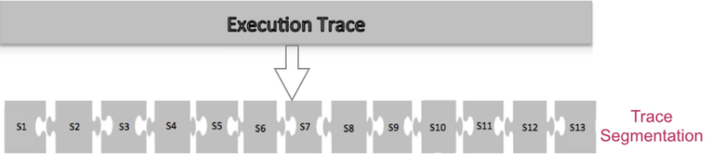
Programs	Scenarios	Number of Segments		Fitness		Time (s)	
		GA	DP	GA	DP	GA	DP
ArgoUML	Create Note	24	13	0.54	0.58	7,080	2.13
	Create Class, Create Note	73	19	0.52	0.60	10,800	4.33
JHotDraw	Draw Rectangle	17	21	0.39	0.67	2,040	0.13
	Add Text, Draw Rectangle	21	21	0.38	0.69	1,260	0.64
	Draw Rectangle, Cut Rectangle	56	20	0.46	0.72	1,200	0.86
	Spawn Window, Draw Circle	63	26	0.34	0.69	240	1



Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?

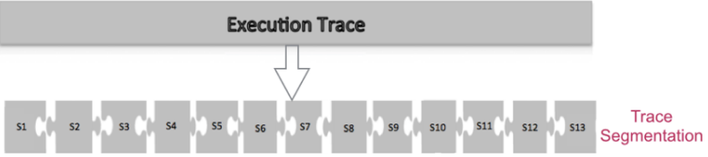
Programs	Scenarios	Number of Segments		Fitness		Time (s)	
		GA	DP	GA	DP	GA	DP
ArgoUML	Create Note	24	**13**	0.54	0.58	7,080	2.13
	Create Class, Create Note	73	**19**	0.52	0.60	10,800	4.33
JHotDraw	Draw Rectangle	17	21	0.39	0.67	2,040	0.13
	Add Text, Draw Rectangle	21	21	0.38	0.69	1,260	0.64
	Draw Rectangle, Cut Rectangle	56	**20**	0.46	0.72	1,200	0.86
	Spawn Window, Draw Circle	63	**26**	0.34	0.69	240	1



Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?

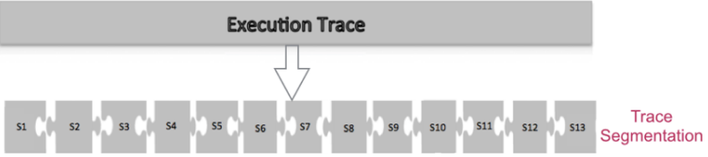
Programs	Scenarios	Number of Segments		Fitness		Time (s)	
		GA	DP	GA	DP	GA	DP
ArgoUML	Create Note	24	13	0.54	**0.58**	7,080	2.13
	Create Class, Create Note	73	19	0.52	**0.60**	10,800	4.33
JHotDraw	Draw Rectangle	17	21	0.39	**0.67**	2,040	0.13
	Add Text, Draw Rectangle	21	21	0.38	**0.69**	1,260	0.64
	Draw Rectangle, Cut Rectangle	56	20	0.46	**0.72**	1,200	0.86
	Spawn Window, Draw Circle	63	26	0.34	**0.69**	240	1



Case Study Results

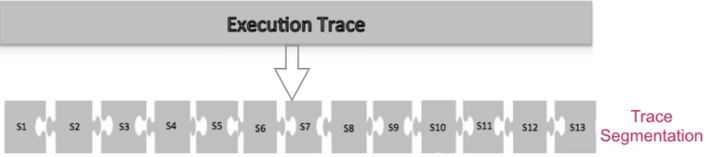
- RQ1: How do the performances of the GA and DP approaches compare?

Programs	Scenarios	Number of Segments		Fitness		Time (s)	
		GA	DP	GA	DP	GA	DP
ArgoUML	Create Note	24	13	0.54	0.58	7,080	**2.13**
	Create Class, Create Note	73	19	0.52	0.60	10,800	**4.33**
JHotDraw	Draw Rectangle	17	21	0.39	0.67	2,040	**0.13**
	Add Text, Draw Rectangle	21	21	0.38	0.69	1,260	**0.64**
	Draw Rectangle, Cut Rectangle	56	20	0.46	0.72	1,200	**0.86**
	Spawn Window, Draw Circle	63	26	0.34	0.69	240	**1**



Case Study Results

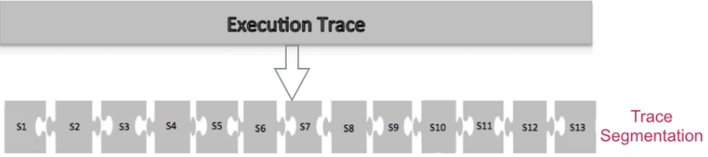
- RQ1: How do the performances of the GA and DP approaches compare?
  - Wilxocon test and Cliff's delta effect size:
    - ☑ Difference of the number of segments;
    - ☑ Values of fitness function;
    - ☑ Computation times.



Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?
  - Wilxocon test and Cliff's delta effect size:

- ☑ Difference of the number of segments;
- ☑ Values of fitness function;
- ☑ Computation times.



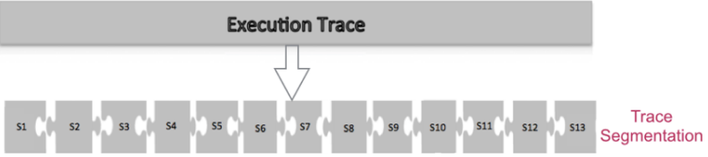
Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?
  - Wilxocon test and Cliff's delta effect size:

☑ Difference of the number of segments;

☑ Values of fitness function;

☑ Computation times.



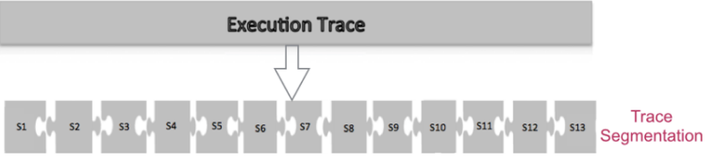
Case Study Results

- RQ1: How do the performances of the GA and DP approaches compare?
  - Wilxocon test and Cliff's delta effect size:

☑ Difference of the number of segments;

☑ Values of fitness function;

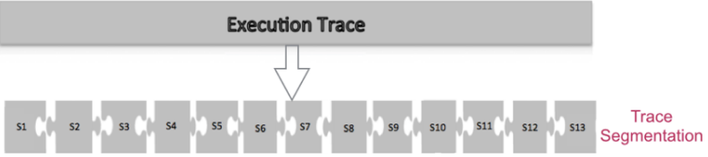
☑ Computation times.



Case Study Results

- RQ2: How do the GA and DP approaches perform?

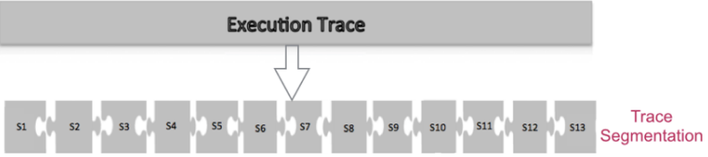
Program	Scenario	Concept	Jaccard		Precision	
			GA	DP	GA	DP
ArgoUML	Create Note	Create Note	0.33	0.87	1	0.99
	Create Class, Create Note	Create Class	0.26	0.53	1	1
	Create Class, Create Note	Create Note	0.34	0.56	1	1
JHotDraw	Draw Rectangle	Draw Rectangle	0.9	0.75	0.9	1
	Add Text, Draw Rectangle	Add Text	0.31	0.33	0.36	0.39
	Add Text, Draw Rectangle	Draw Rectangle	0.62	0.52	0.62	1
	Draw Rectangle, Cut Rectangle	Draw Rectangle	0.74	0.24	0.79	0.24
	Draw Rectangle, Cut Rectangle	Cut Rectangle	0.22	0.31	1	1
	Spawn Window, Draw Circle	Draw Circle	0.82	0.82	0.82	1
	Spawn Window, Draw Circle	Spawn Window	0.42	0.44	1	1



Case Study Results

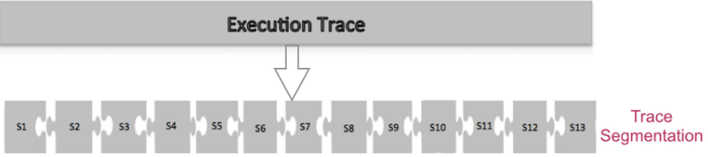
- RQ2: How do the GA and DP approaches perform?

Program	Scenario	Concept	Jaccard		Precision	
			GA	DP	GA	DP
ArgoUML	Create Note	Create Note	0.33	**0.87**	1	0.99
	Create Class, Create Note	Create Class	0.26	**0.53**	1	**1**
	Create Class, Create Note	Create Note	0.34	**0.56**	1	**1**
JHotDraw	Draw Rectangle	Draw Rectangle	0.9	0.75	0.9	**1**
	Add Text, Draw Rectangle	Add Text	0.31	**0.33**	0.36	**0.39**
	Add Text, Draw Rectangle	Draw Rectangle	0.62	0.52	0.62	**1**
	Draw Rectangle, Cut Rectangle	Draw Rectangle	0.74	0.24	0.79	0.24
	Draw Rectangle, Cut Rectangle	Cut Rectangle	0.22	**0.31**	1	**1**
	Spawn Window, Draw Circle	Draw Circle	0.82	**0.82**	0.82	**1**
	Spawn Window, Draw Circle	Spawn Window	0.42	**0.44**	1	**1**



Case Study Results

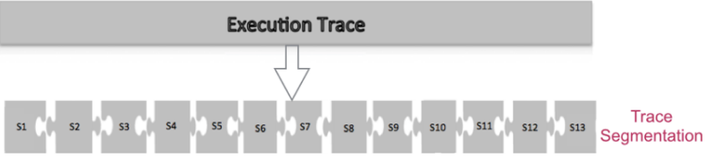
- RQ2: How do the GA and DP approaches perform?
  - Wilxocon test and Cliff's delta effect size:
    - ☑ Jaccad scores
    - ☑ Precision



Case Study Results

- RQ2: How do the GA and DP approaches perform?
  - Wilxocon test and Cliff's delta effect size:

- ☑ Jaccad scores
- ☑ Precision



Case Study Results

- RQ2: How do the GA and DP approaches perform?
  - Wilxocon test and Cliff's delta effect size:

☑ Jaccad scores

☑ Precision

Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work

Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.

Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario draw rectangle:

Original size
Compressed size
Number of segments

First Execution

Second Execution

Third Execution

Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario draw rectangle:

Original size
Compressed size
Number of segments

First Execution

Second Execution

Third Execution

Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario draw rectangle:

Original size
Compressed size
Number of segments

First Execution

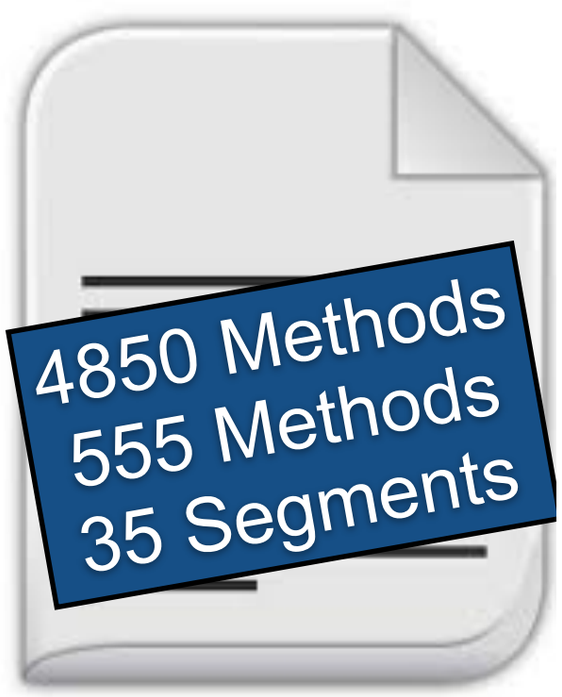
Second Execution

Third Execution

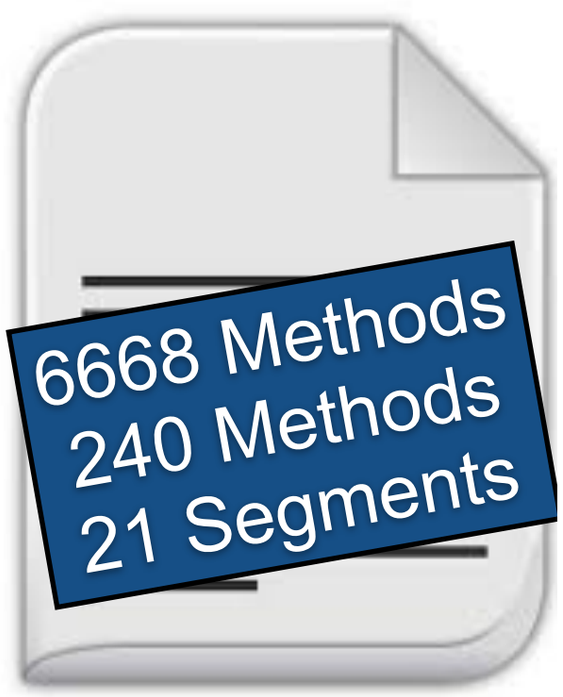
Segments Merging

- Multi-threading: induces variability in traces collected for a given scenario.
- Scenario draw rectangle:

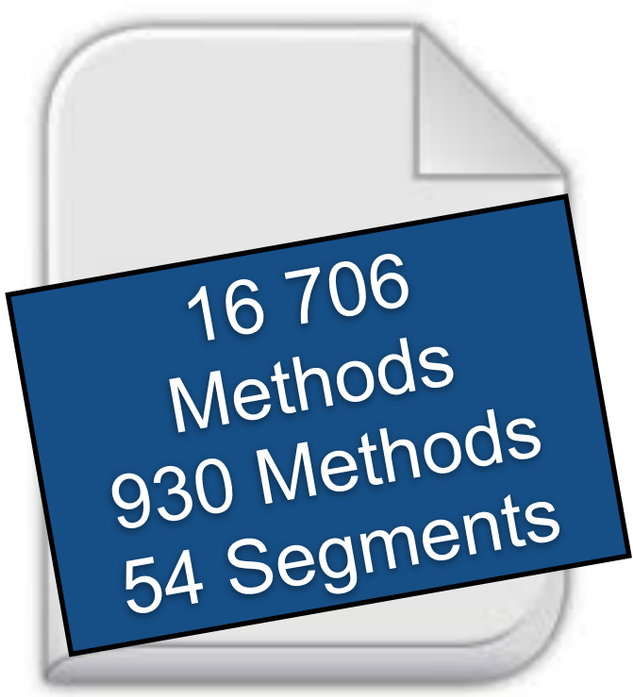
Original size
Compressed size
Number of segments



First Execution



Second Execution



Third Execution

Segments Merging

- We merge segments obtained in multiple executions of the same scenario.
- Similarity:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Segments Merging

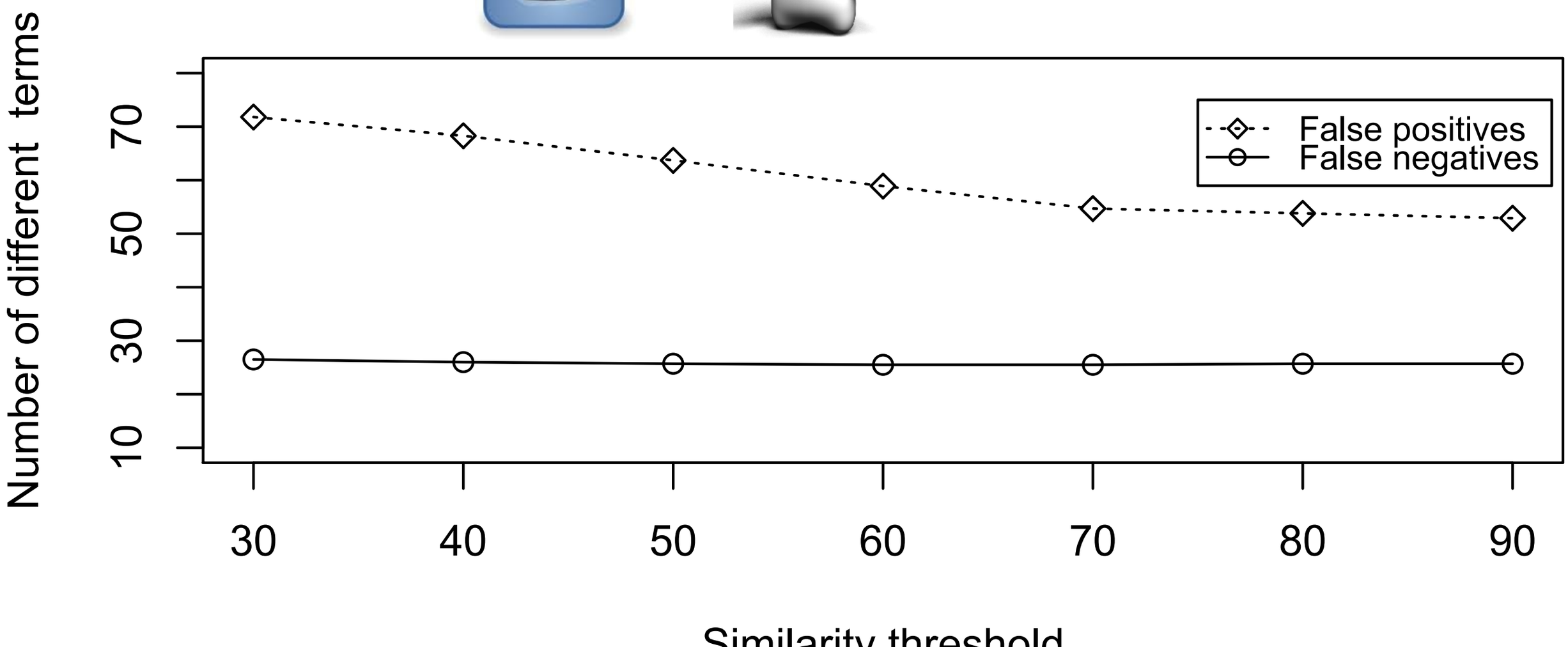
- We merge segments obtained in multiple executions of the same scenario.
- Similarity:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|}$$

Threshold

Similarity Threshold

- Projects:



Segments Merging

- Example:

Trace 1	S1	S2	S3	S4
Trace 2	Z1	Z2	Z3	

Segments Merging

- Example:

Trace 1	S1	S2	S3	S4
Trace 2	Z1	Z2	Z3	

Threshold= 70%

Segments Merging

- Example:

Trace 1
Trace 2

S1	S2	S3	S4
Z1	Z2	Z3	

Threshold= 70%

	S1	S2	S3	S4
Z1	0.9	0.5	0.2	0.3
Z2	0.3	0.9	0.5	0.2
Z3	0.3	0.2	0.5	0.8

Segments Merging

- Example:

Trace 1 S1 S2 S3 S4

Trace 2 Z1 Z2 Z3

Threshold= 70%

	S1	S2	S3	S4
Z1	0.9	0.5	0.2	0.3
Z2	0.3	0.9	0.5	0.2
Z3	0.3	0.2	0.5	0.8

Segments Merging

- Example:

Trace 1
Trace 2

S1	S2	S3	S4
Z1	Z2	Z3	

Threshold= 70%

	S1	S2	S3	S4
Z1	0.9	0.5	0.2	0.3
Z2	0.3	0.9	0.5	0.2
Z3	0.3	0.2	0.5	0.8

Synthetic Trace

S1, Z1	S2, Z2	Z3, S4

Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work

Segments Labelling

- Issue: choice of the most appropriate source of information.

Segments Labelling

- Issue: choice of the most appropriate source of information.
    - Method bodies:

Segments Labelling

- Issue: choice of the most appropriate source of information.
  - Method bodies:
    - ☑ Identifiers;

Segments Labelling

- Issue: choice of the most appropriate source of information.
  - Method bodies:
    - ☑ Identifiers;
    - ☑ Comments;

Segments Labelling

- Issue: choice of the most appropriate source of information.
  - Method bodies:
    - ☑ Identifiers;
    - ☑ Comments;
  - Method signature.

Segments Labelling

- Issue: choice of the most appropriate source of information.
  - Method bodies:
  - ...ure.

Method signatures provide more meaningful terms when labeling software artifacts than other sources. [De Lucia, 2012]

Segments Labelling

- Source of information: terms contained in the signature of methods.
- Hypothesis: a term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- Ranks the terms of the segment by *TF-IDF* and keeps the topmost ones.

Segments Labelling

- To reduce the time and effort: segments are characterized using some unique methods (TF-IDF).
- Small version (5): result in loss of relevant information.
- Medium version(15): preserve better the relevant information.

Experiment Design

- Research Questions:
  - RQ1: How do the labels produced by the participants change when providing them different amount of information?

Experiment Design

- Research Questions:
  - RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - RQ2: How do the labels produced by the participants compare to the generated labels?

Experiment Design

- Research Questions:
  - RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - RQ2: How do the labels produced by the participants compare to the generated labels?
- Projects:






Experiment Design

- Research Questions:
  - RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - RQ2: How do the labels produced by the participants compare to the generated labels?
- Projects:






- Participants:

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
- We group participants into 3 groups. Each version is assigned to a different group.

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
- We group participants into 3 groups. Each version is assigned to a different group.

Terms of small segment

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
- We group participants into 3 groups. Each version is assigned to a different group.

Terms of small segment

Terms of medium segment

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
- We group participants into 3 groups. Each version is assigned to a different group.

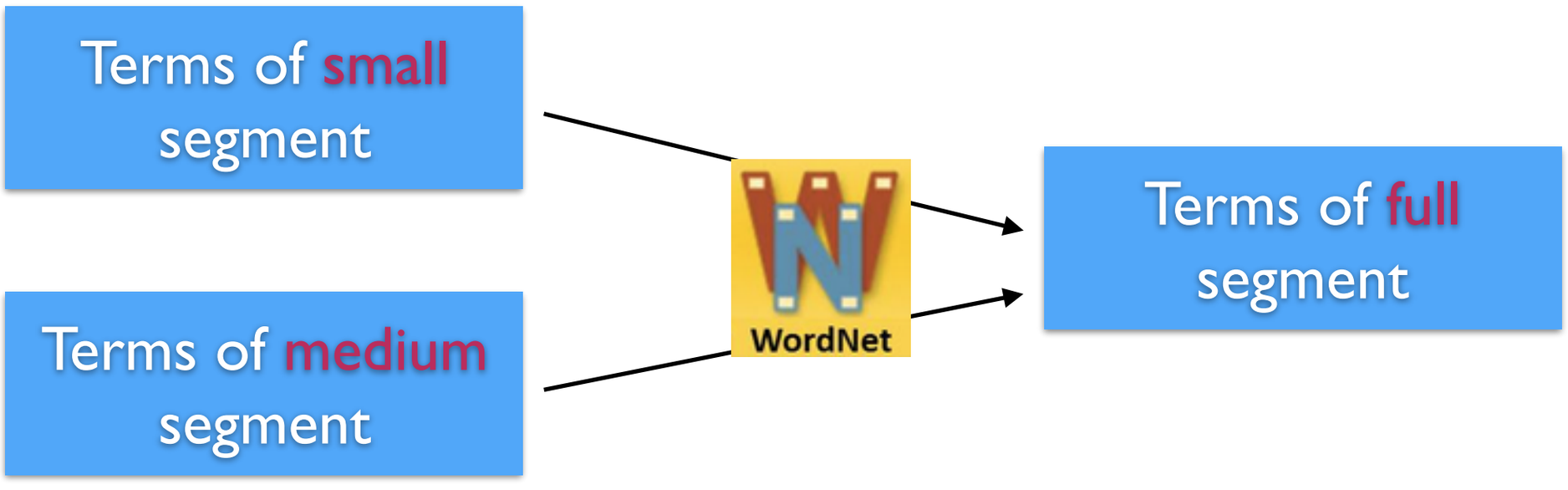
Terms of small segment

Terms of medium segment

Terms of full segment

Experiment Design

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
- We group participants into 3 groups. Each version is assigned to a different group.

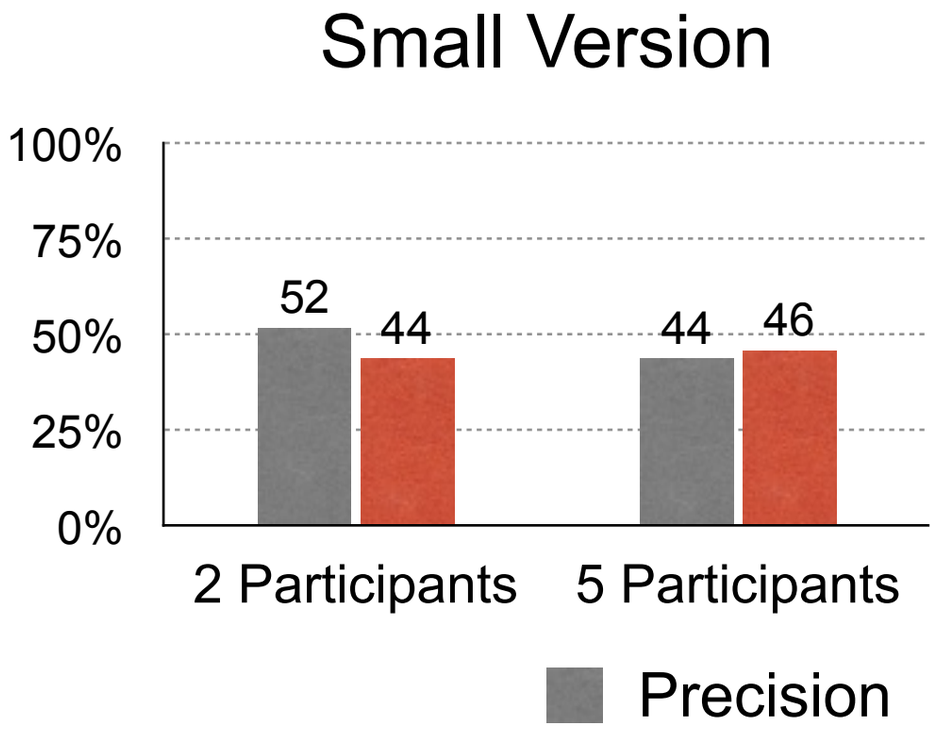


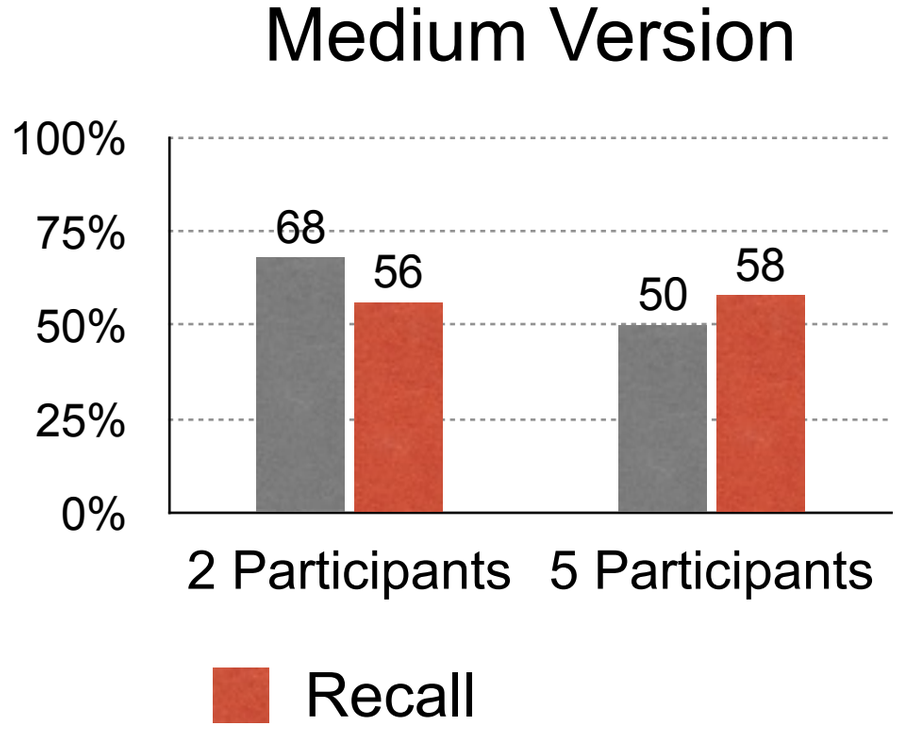
Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?

Experiment Results

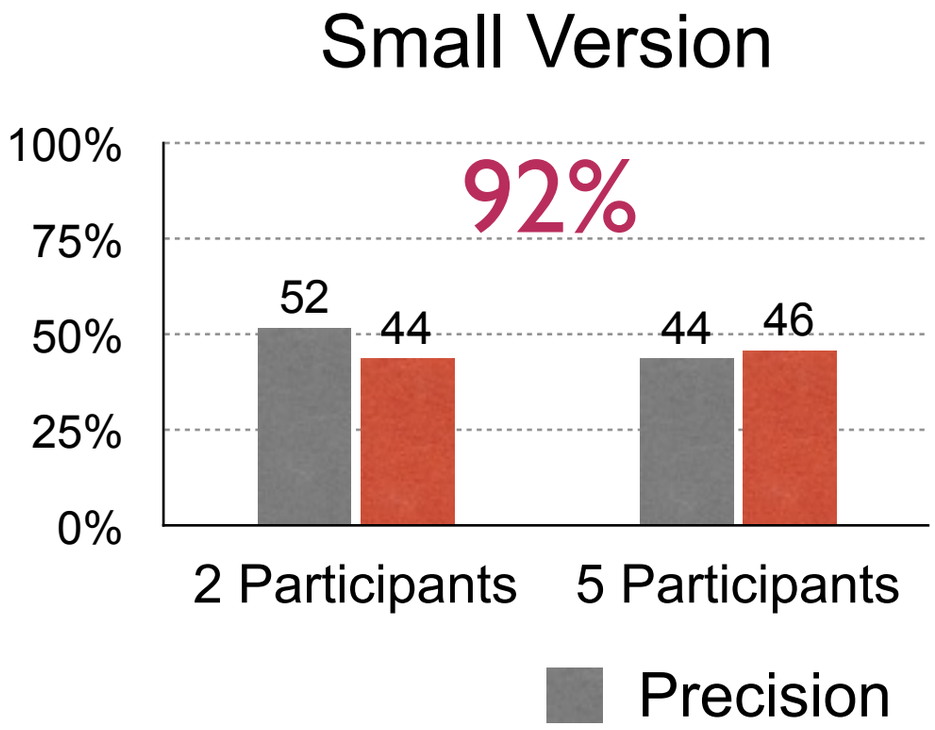
- RQ1: How do the labels produced by the participants change when providing them different amount of information?

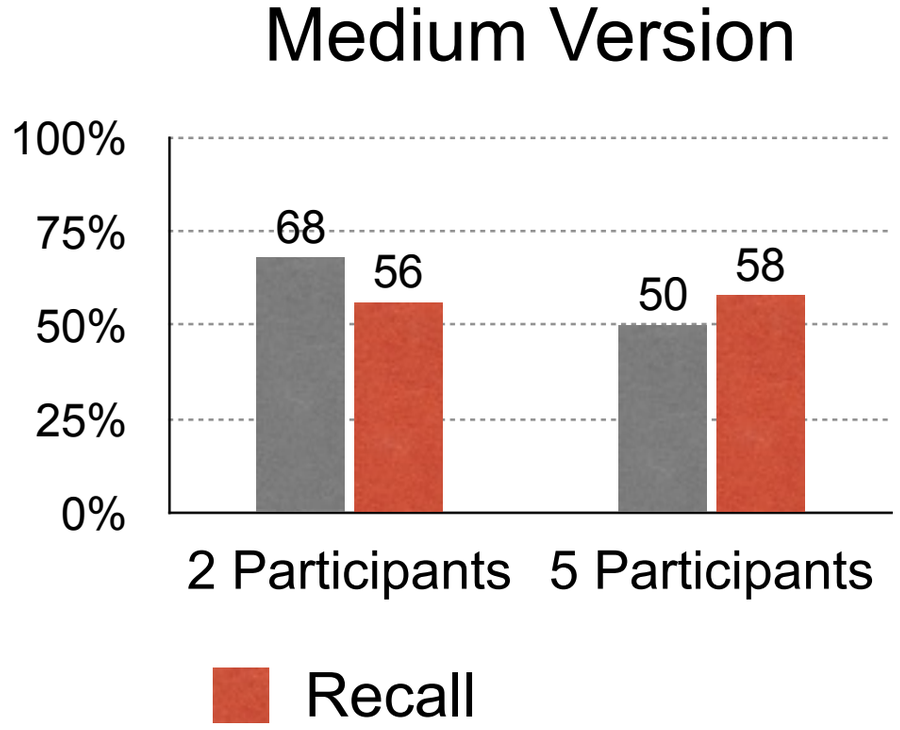




Experiment Results

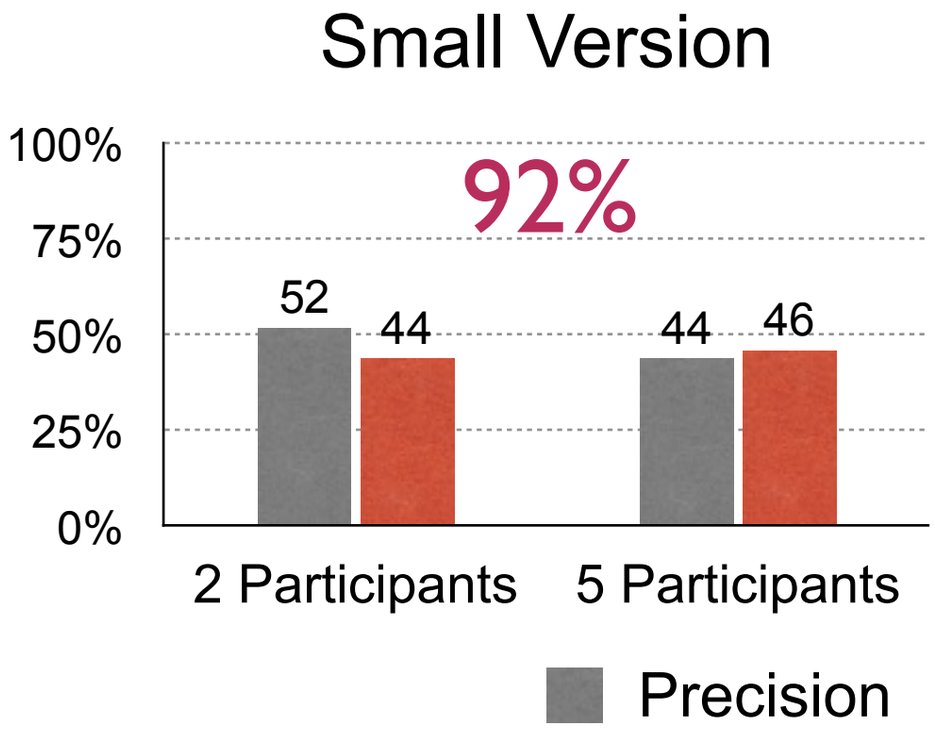
- RQ1: How do the labels produced by the participants change when providing them different amount of information?

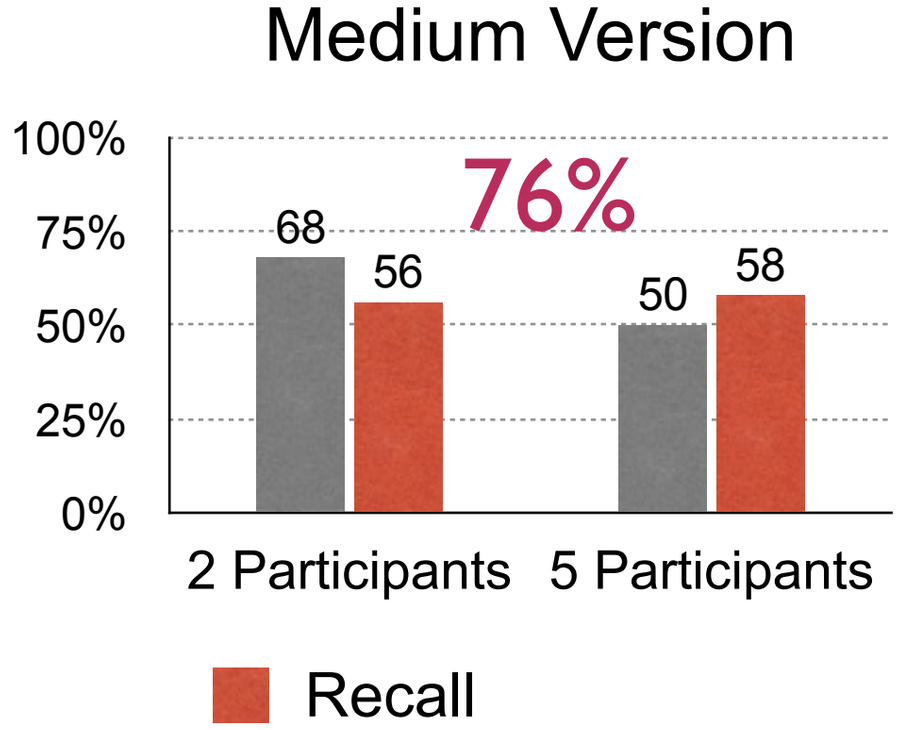




Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?





Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:
    - ☑ Number of participants;
    - ☑ Size of the segment (full, medium, small);
    - ☑ Their interaction;
    - ☑ Years of programming experience.

Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:

  - ☑Number of participants;
  - ☑Size of the segment (full, medium, small);
  - ☑Their interaction;
  - ☑Years of programming experience.

Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:

☑Number of participants;

☑Size of the segment (full, medium, small);

☑Their interaction;

☑Years of programming experience.

Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:

☑Number of participants;

☑Size of the segment (full, medium, small);

☑Their interaction;

☑Years of programming experience.

Experiment Results

- RQ1: How do the labels produced by the participants change when providing them different amount of information?
  - Two-way permutation test:

☑Number of participants;

☑Size of the segment (full, medium, small);

☑Their interaction;

☑Years of programming experience.

Experiment Design

- RQ2: How do the labels produced by the participants compare to the generated labels?

Experiment Design

- RQ2: How do the labels produced by the participants compare to the generated labels?
  - Oracle: 210 segments (less than 100) manually labelled by the participants.

Experiment Design

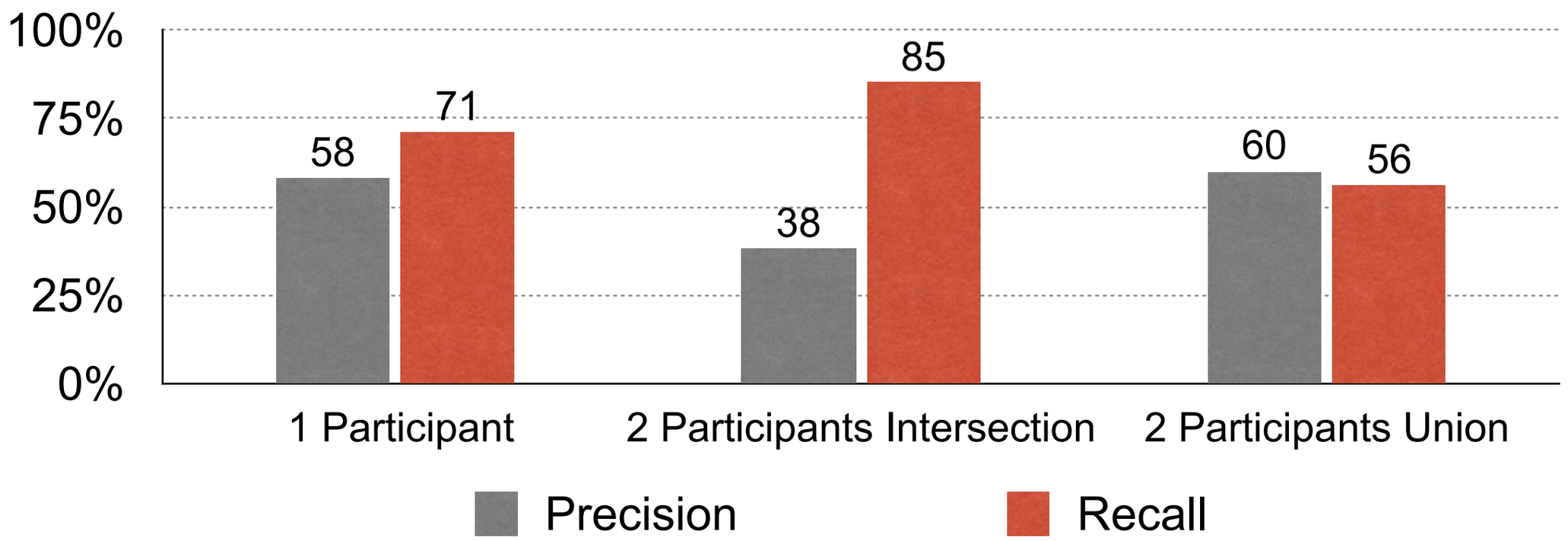
- RQ2: How do the labels produced by the participants compare to the generated labels?
  - Oracle: 210 segments (less than 100) manually labelled by the participants.
  - Evaluation: 1 participant and 2 participants.

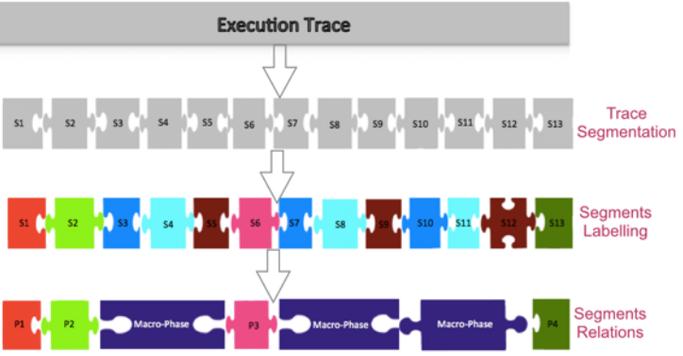
Experiment Results

- RQ2: How do the labels produced by the participants compare to the generated labels?

Experiment Results

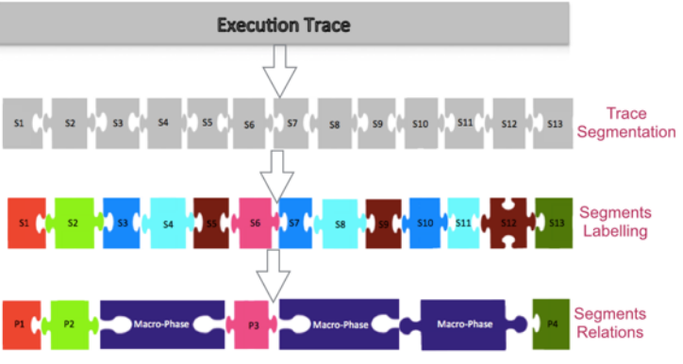
- RQ2: How do the labels produced by the participants compare to the generated labels?





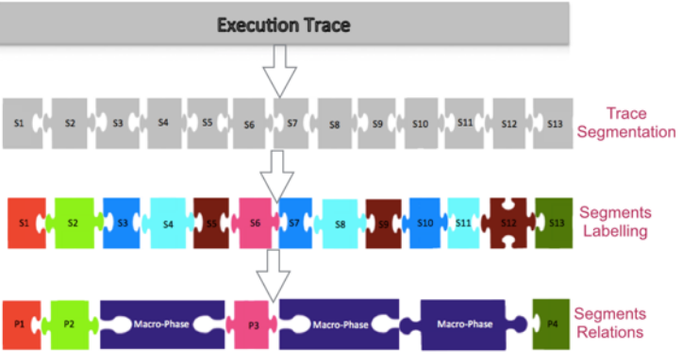
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work



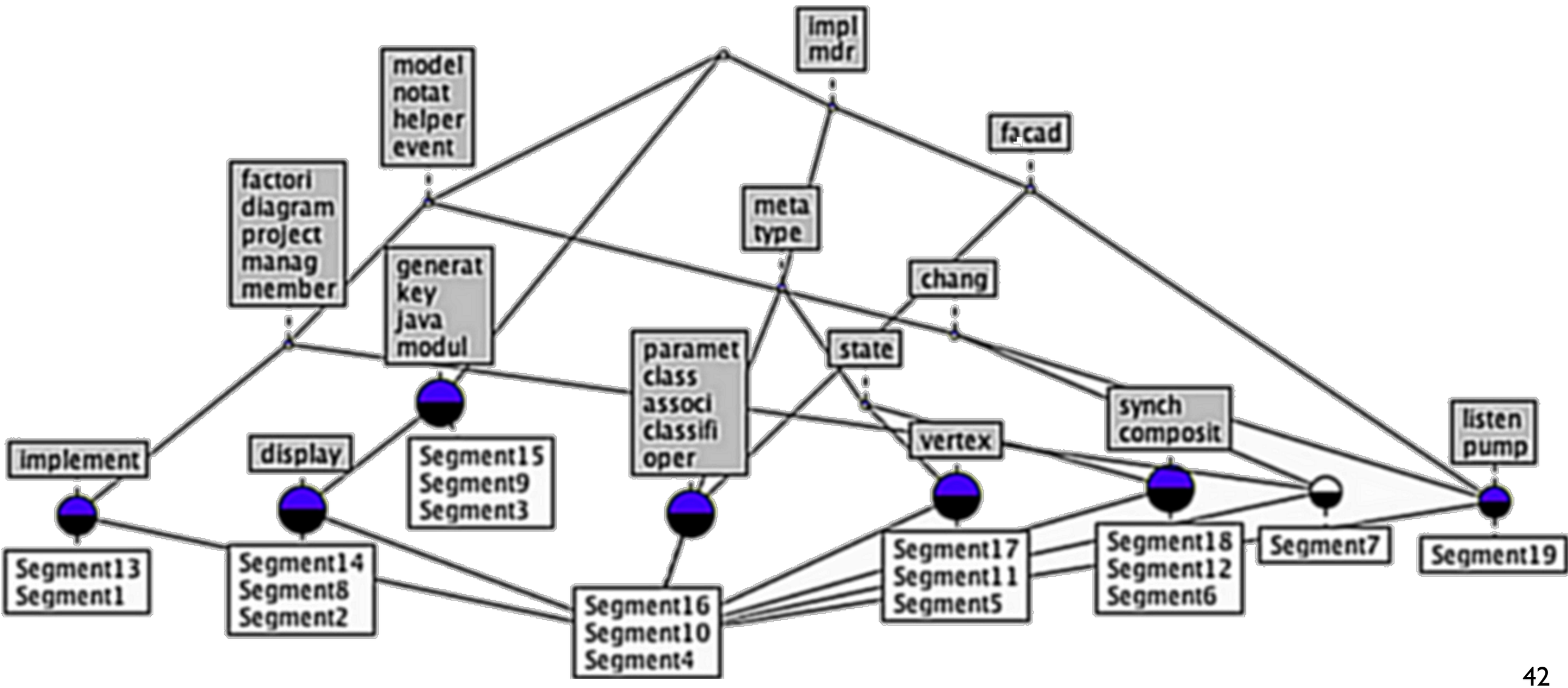
Segments Relations

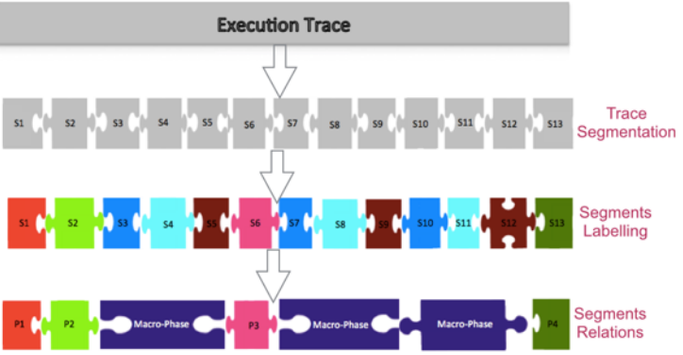
- Formal Concept Analysis: used to identify relations between concepts identified in different segments.
- Groups objects that have common attributes: objects are segments and attributes are terms.
- An FCA concept: maximal collection of objects that have common attributes.



Segments Relations

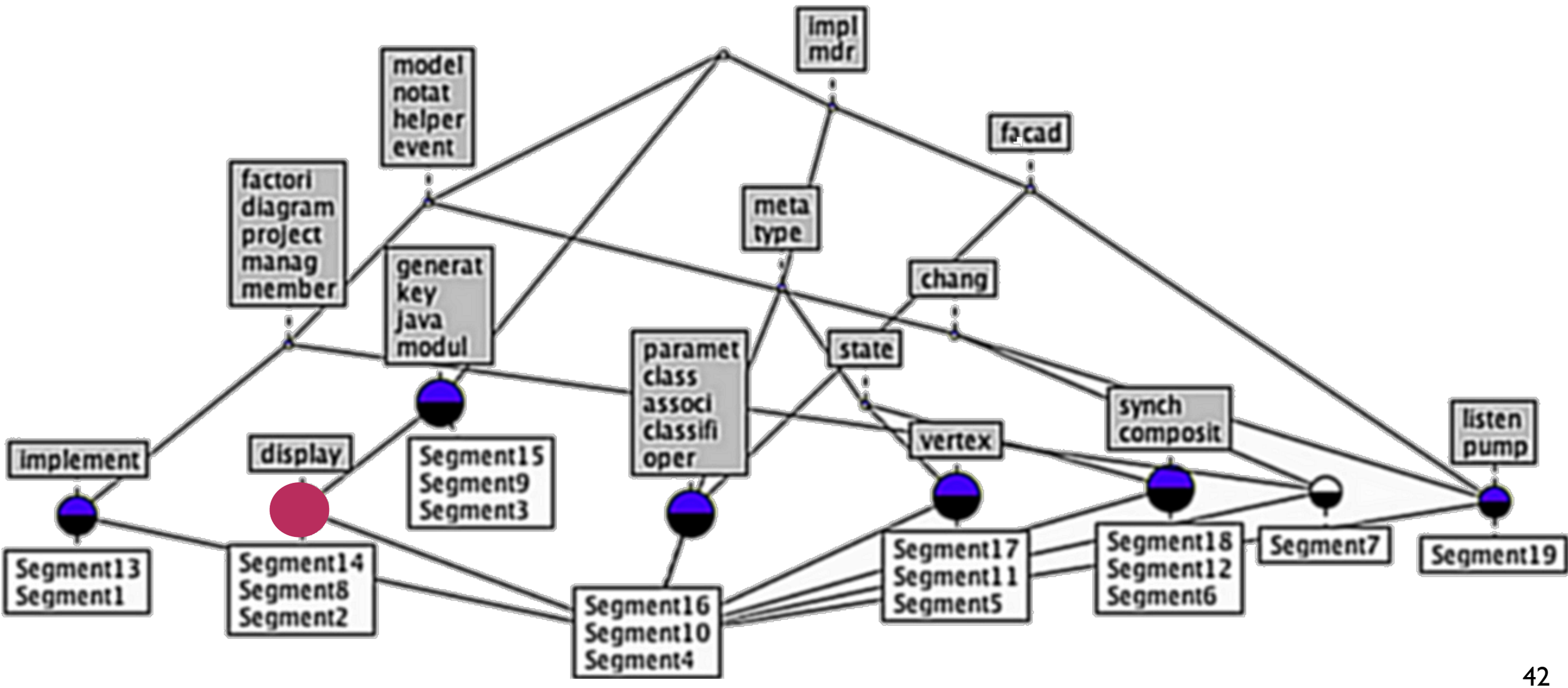
- FCA lattice for the execution trace of the scenario create a class.

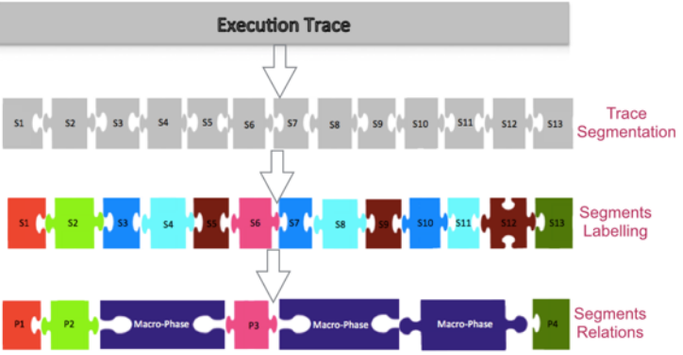




Segments Relations

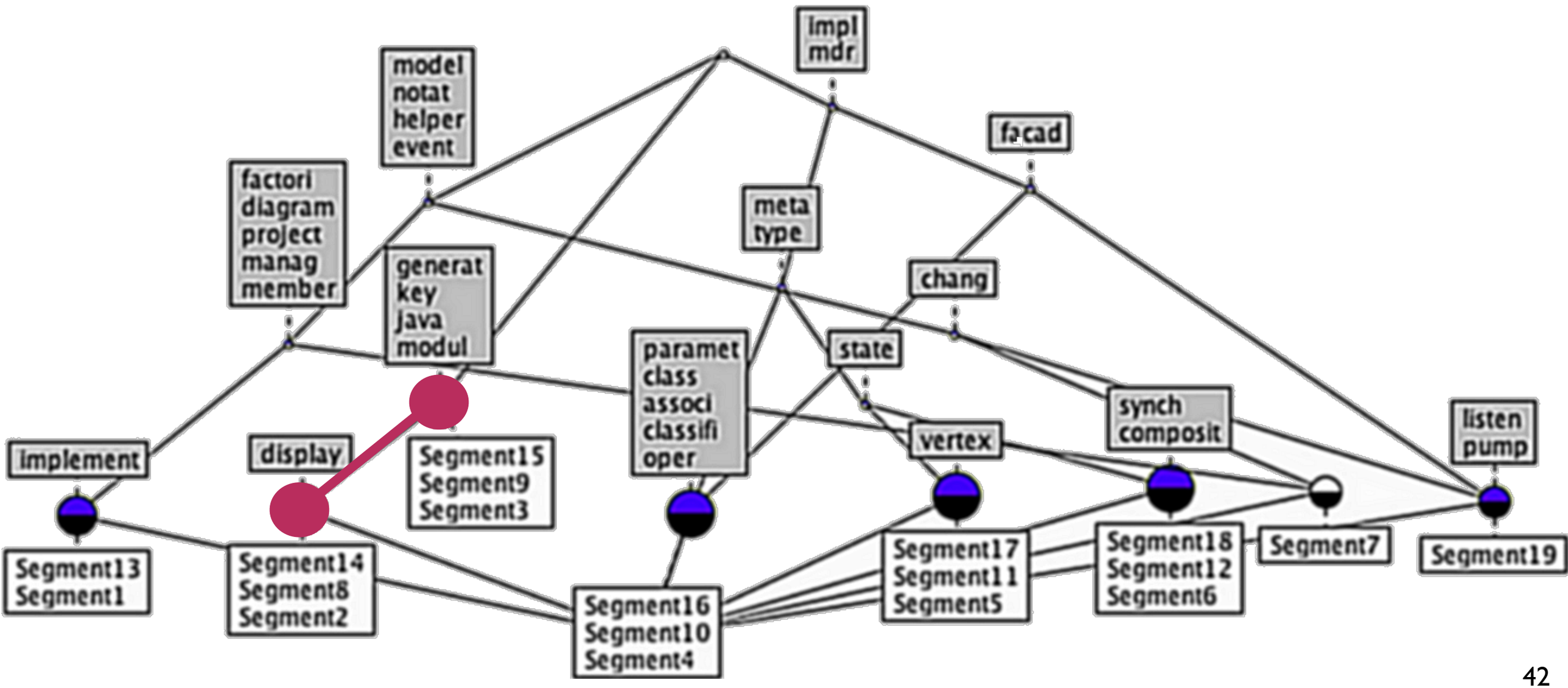
- FCA lattice for the execution trace of the scenario create a class.

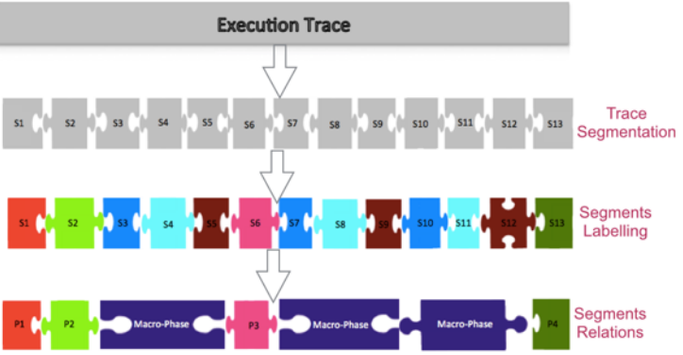




Segments Relations

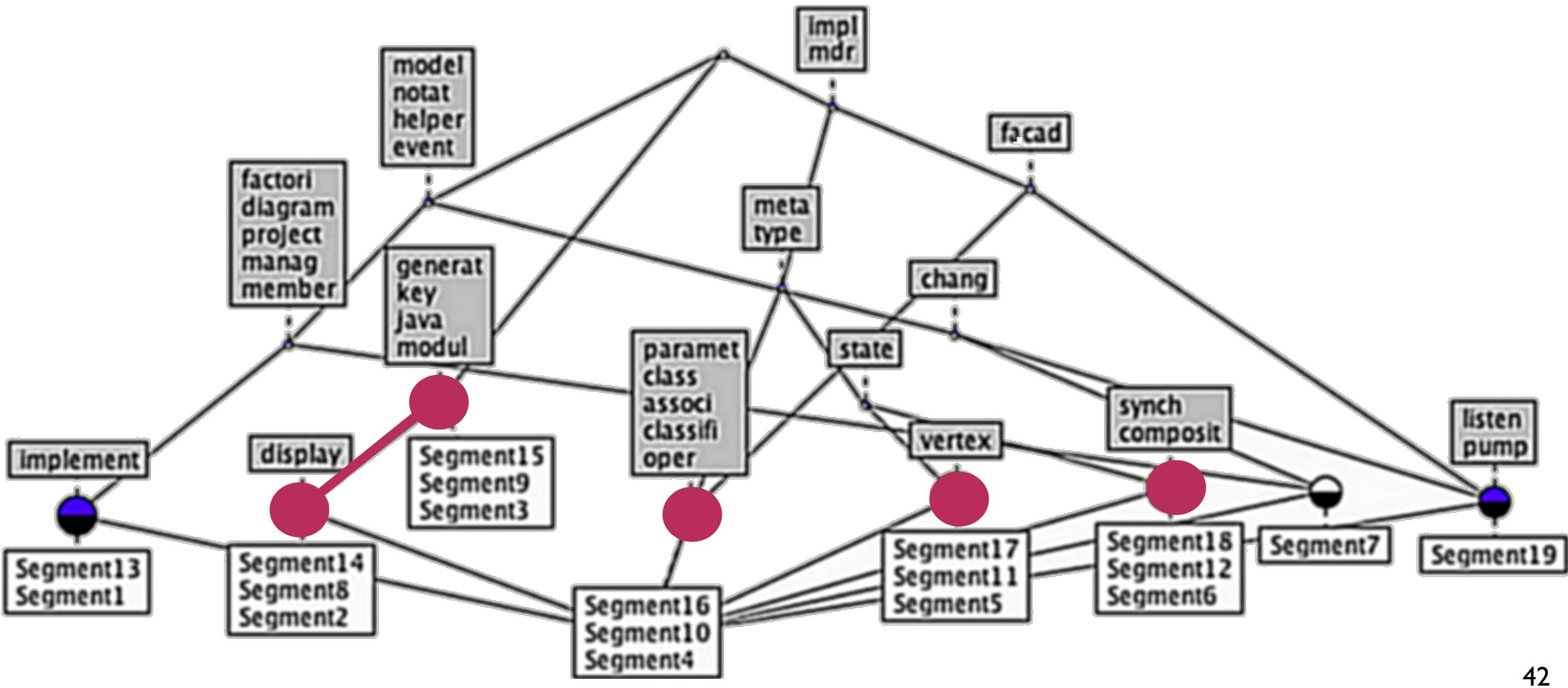
- FCA lattice for the execution trace of the scenario create a class.

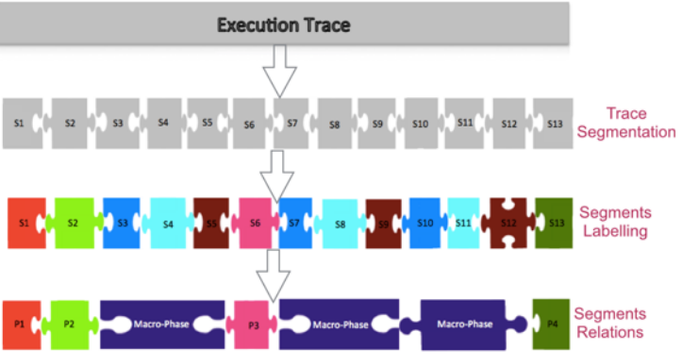




Segments Relations

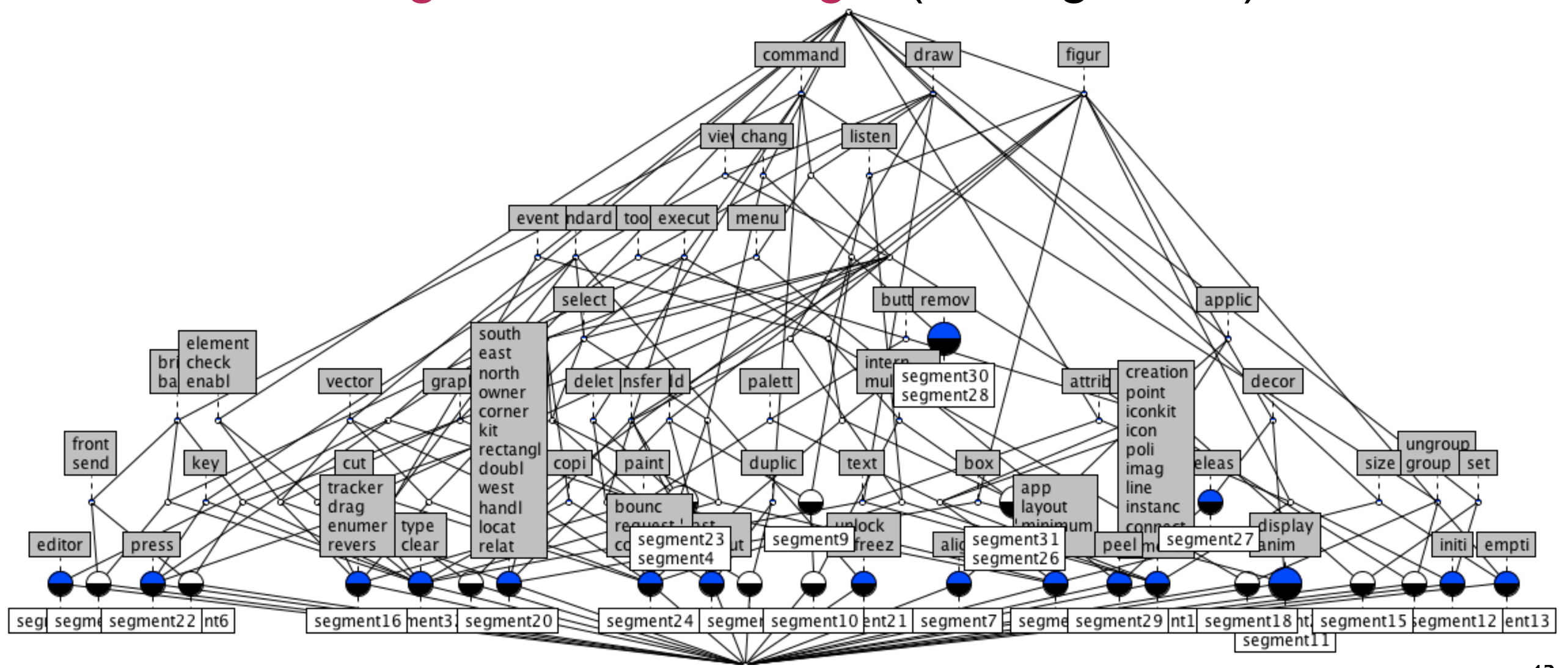
- FCA lattice for the execution trace of the scenario create a class.

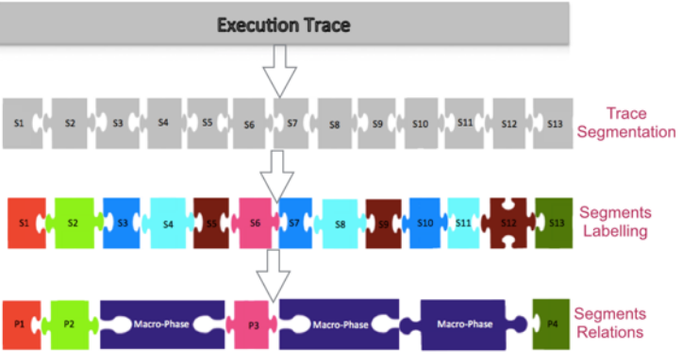




Segments Relations

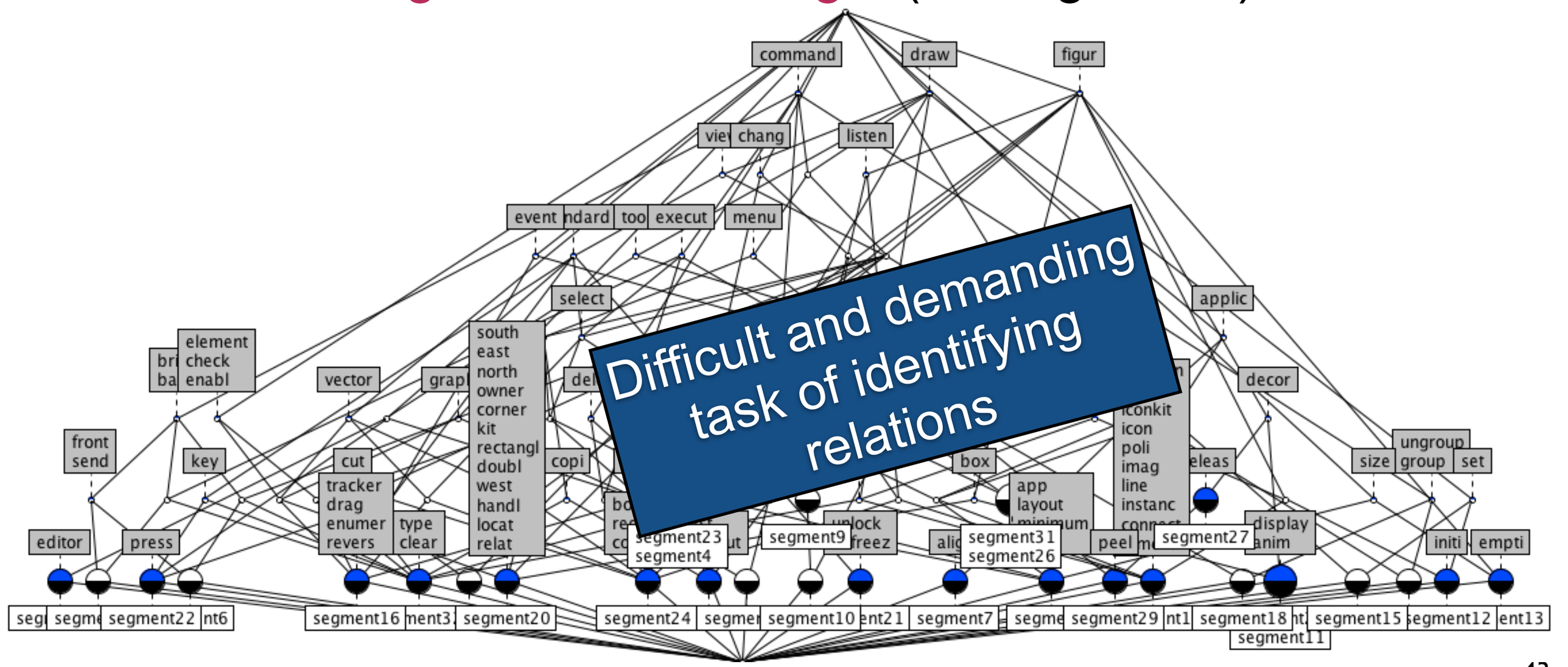
- FCA lattice for the trace of the scenario draw rectangle delete rectangle (32 segments).

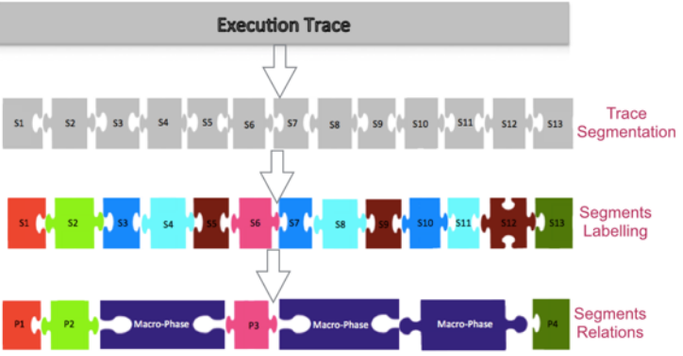




Segments Relations

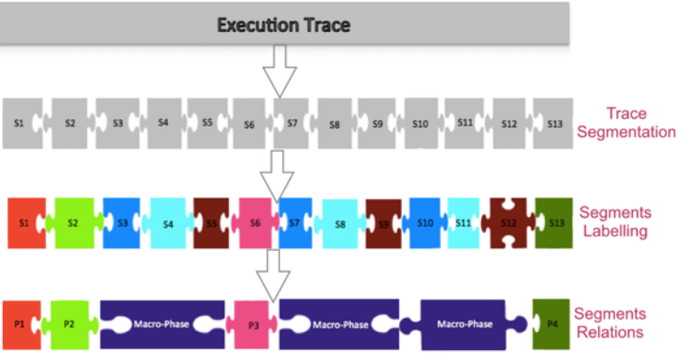
- FCA lattice for the trace of the scenario draw rectangle delete rectangle (32 segments).





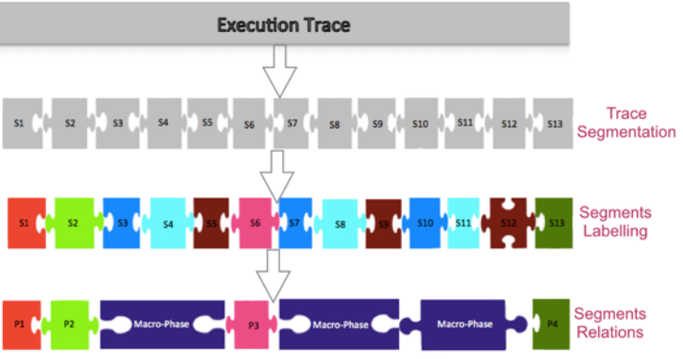
Experiment Design

- Research Questions:
  - RQ1: To what extent does our approach correctly identify relations among segments?



Experiment Design

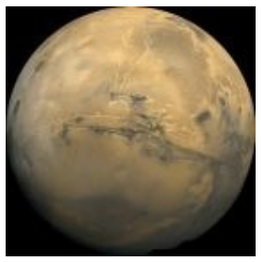
- Research Questions:
  - RQ1: To what extent does our approach correctly identify relations among segments?
- Projects:



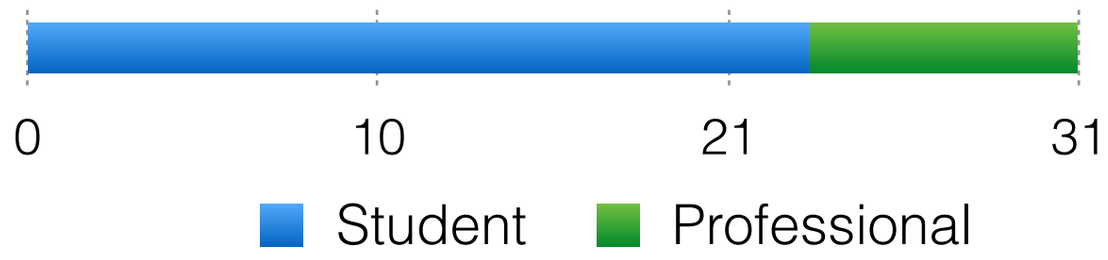
Experiment Design

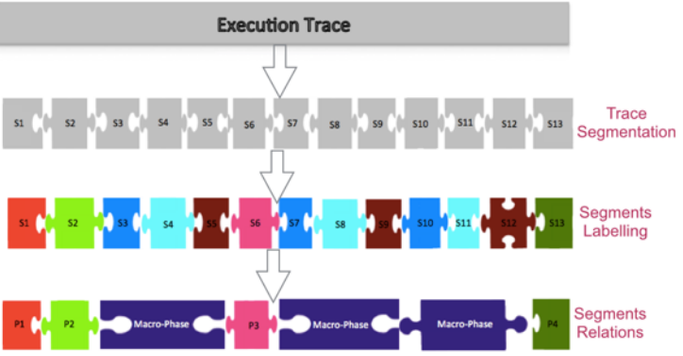
- Research Questions:
  - RQ1: To what extent does our approach correctly identify relations among segments?
- Projects:





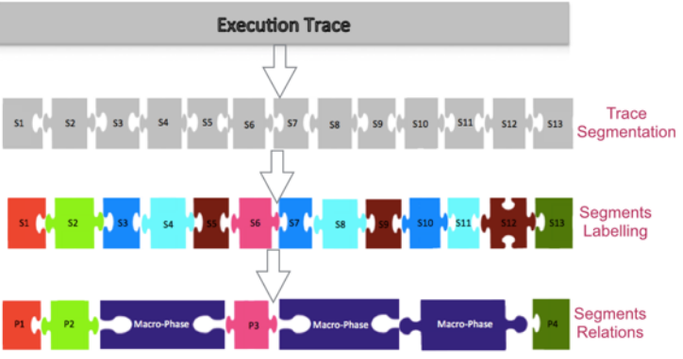
- Participants:





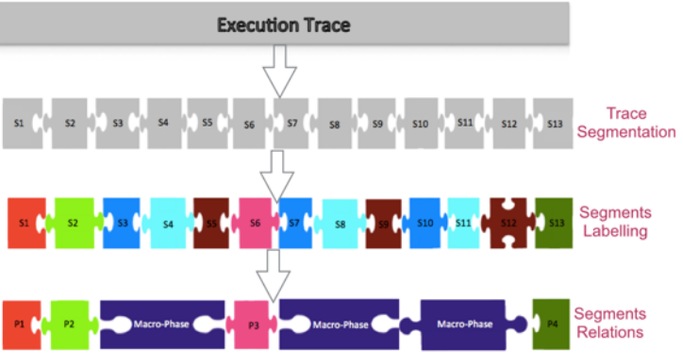
Experiment Design

- RQ1: To what extent does our approach correctly identify relations among segments?



Experiment Design

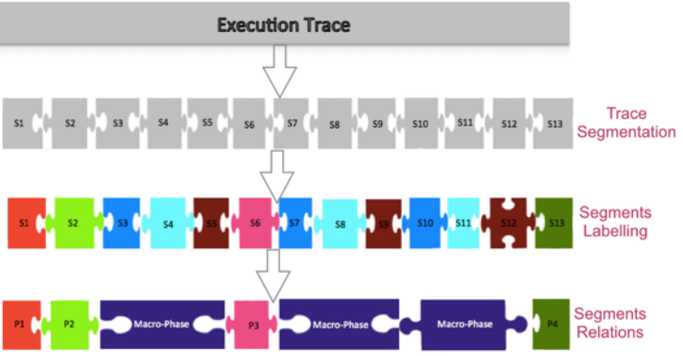
- RQ1: To what extent does our approach correctly identify relations among segments?
  - 100 relations are validated by participants.



Experiment Design

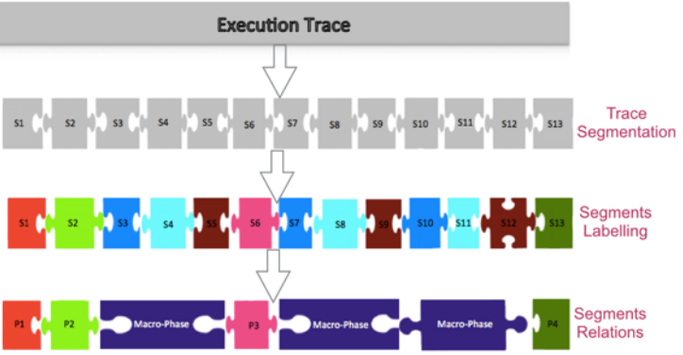
- RQ1: To what extent does our approach correctly identify relations among segments?
  - 100 relations are validated by participants.

	Segments	Labels	Relations
Our Approach	9	listener, add, change, figure	sub/super concept
	10	figure, listener, add, internal, multicaster, event, change	
Participant 1	9	composite, figure, trigger, event	sub/super concept
	10	manage, figure, change, event, trigger	
Participant 2	9	abstract, figure, change, add, listener	same concept
	10	figure, change, event, multicaster, add, listener	



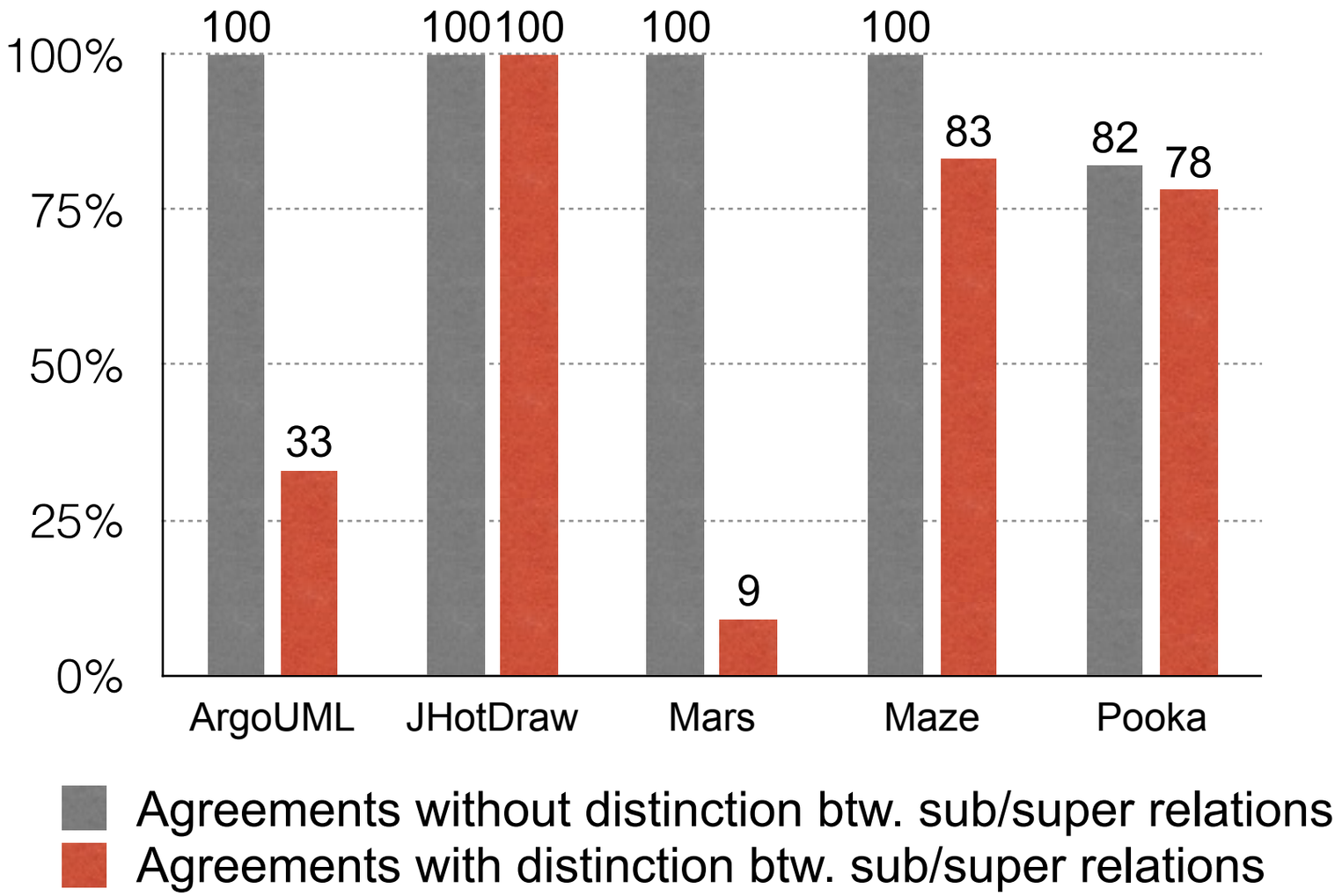
Experiment Results

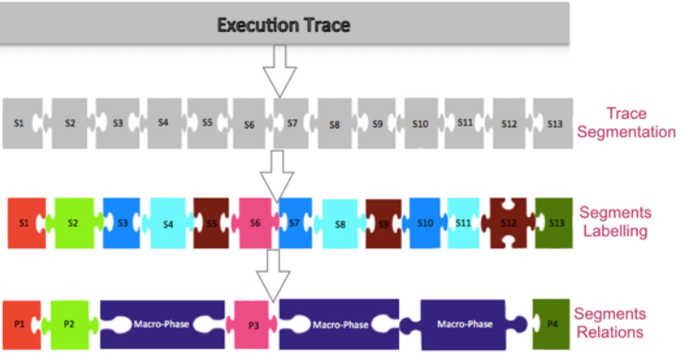
- RQ1: To what extent does our approach correctly identify relations among segments?



Experiment Results

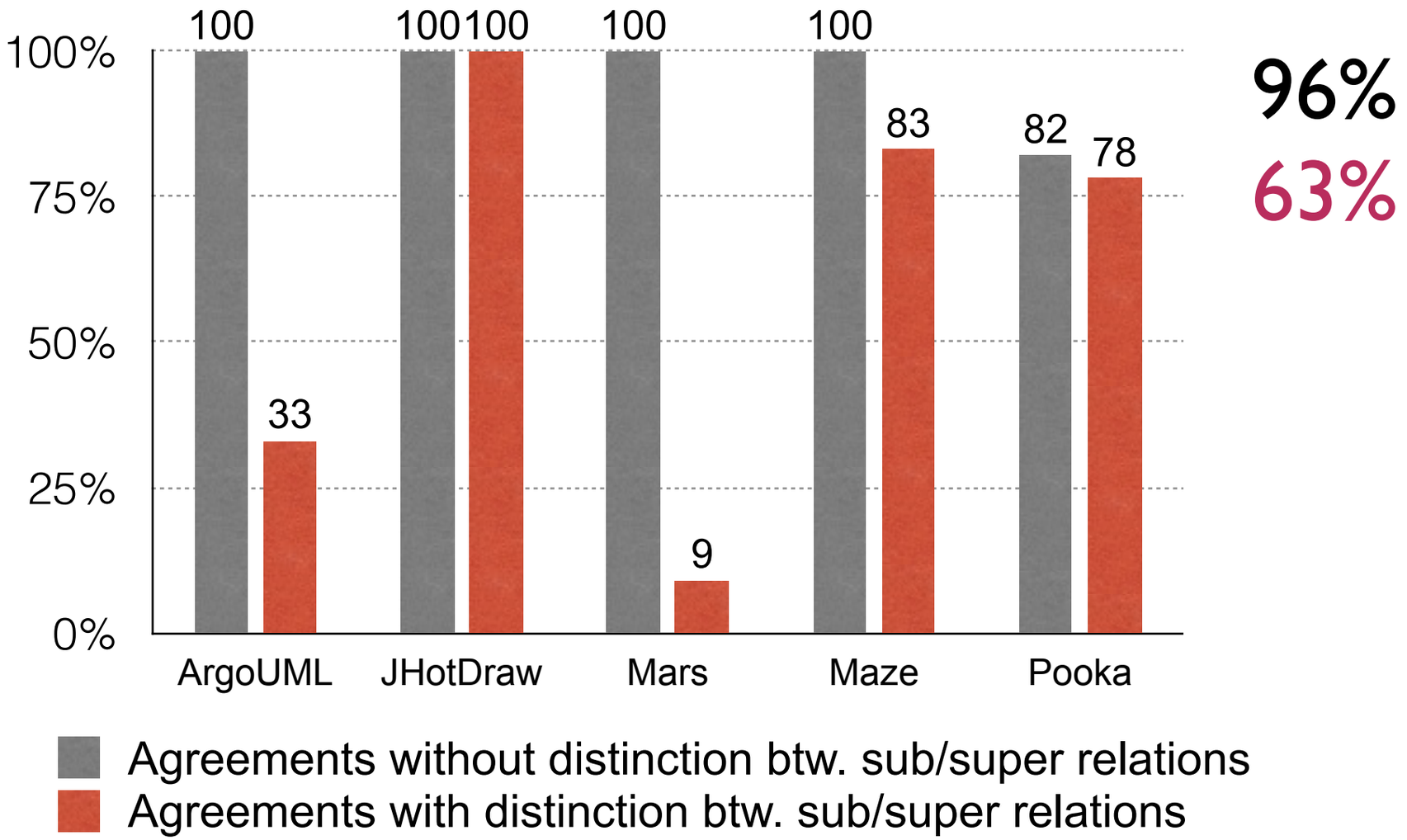
- RQ1: To what extent does our approach correctly identify relations among segments?

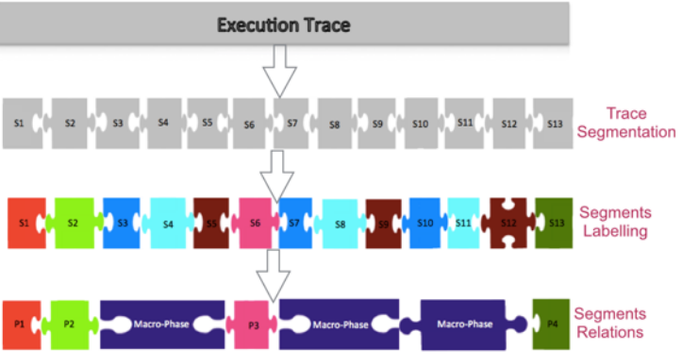




Experiment Results

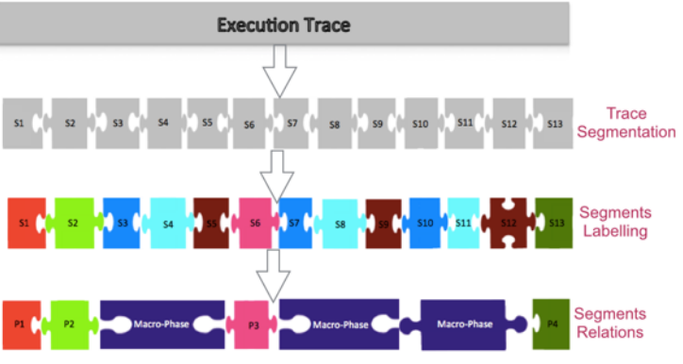
- RQ1: To what extent does our approach correctly identify relations among segments?





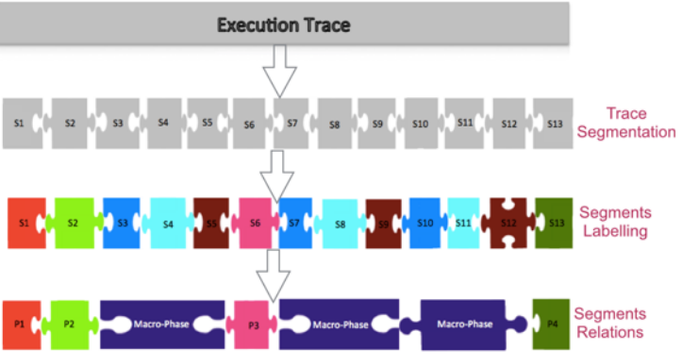
Outline

- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work



Usefulness Evaluation

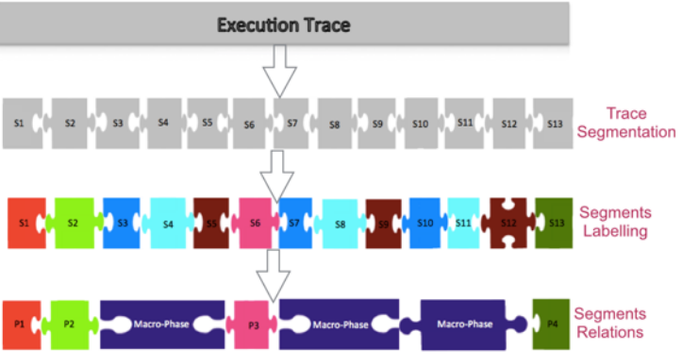
- During maintenance, developers are interested to understand some segments of a trace that implement some concepts of interest.
- Trace Segmentation approach groups these concepts in few segments.
- Labelling and relating segments approach guide developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.



Empirical Study

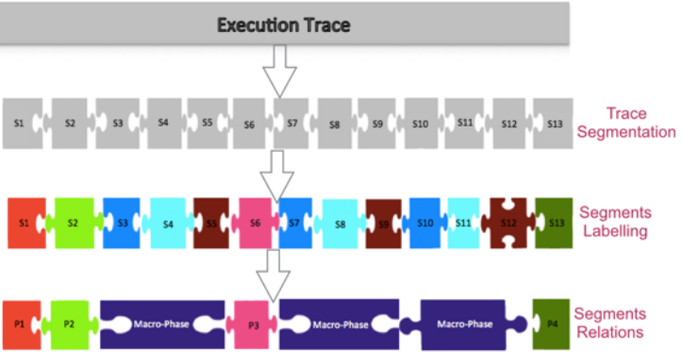
- Research Questions:
  - RQ1: Does our trace segmentation has a potential to support concept location?
  - RQ2: To what extent does our approach support concept location tasks?
- Projects:

The dataset was made publicly available by Dit et al., [2013]



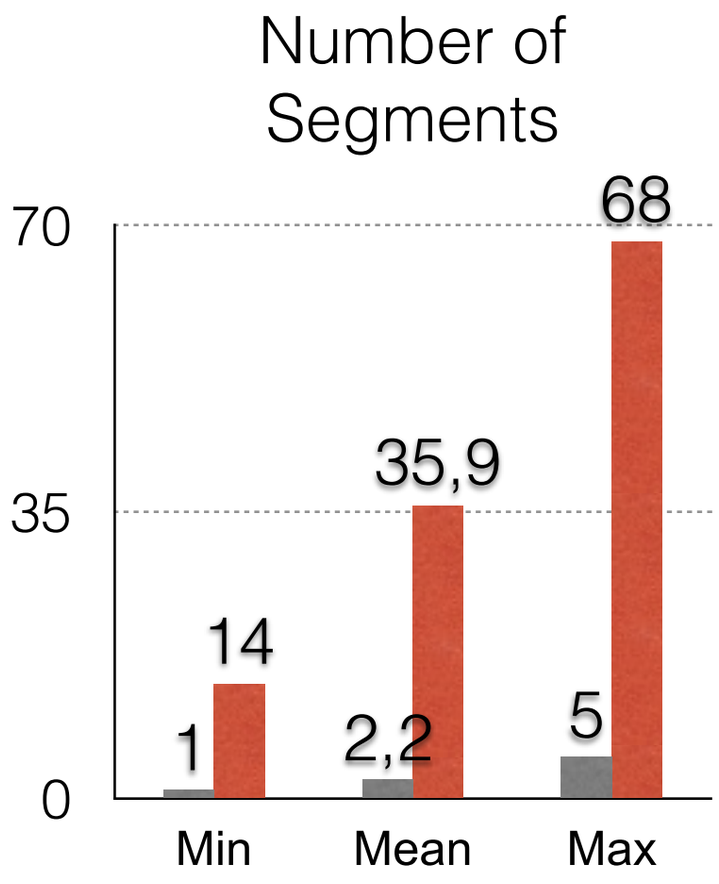
Empirical Results

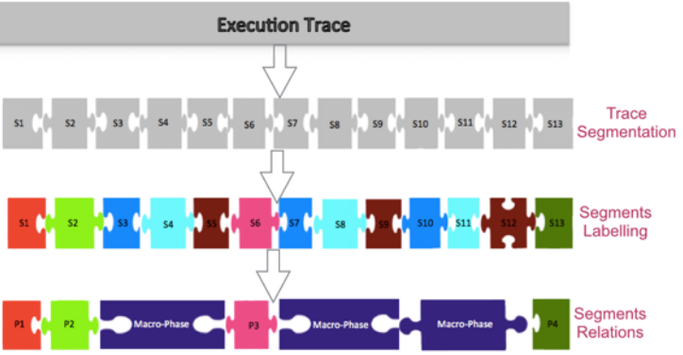
- RQ1: Does our trace segmentation approach has a potential to support concept location?



Empirical Results

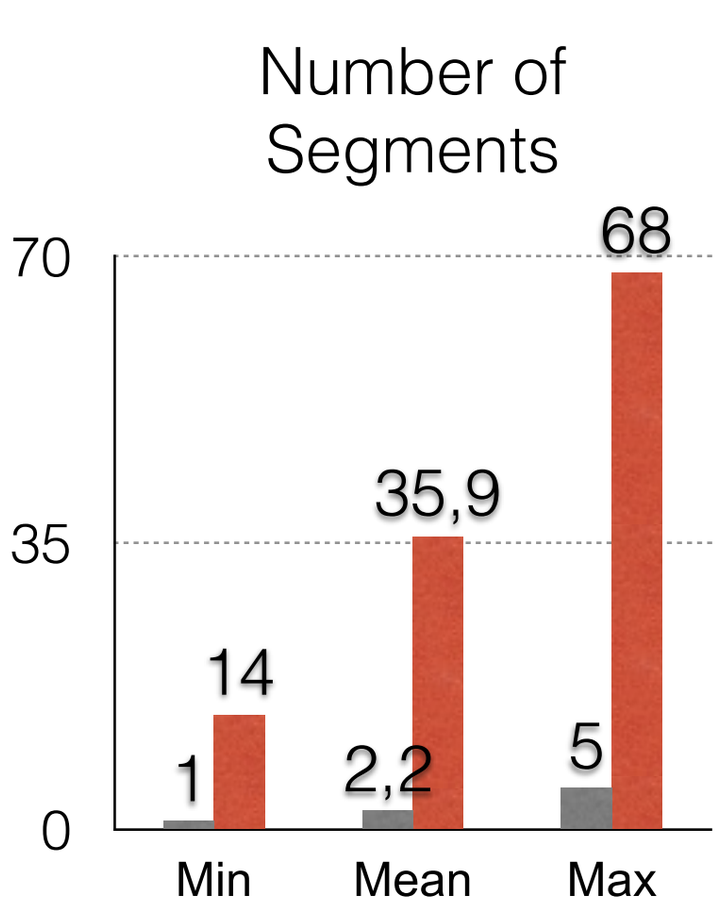
- RQ1: Does our trace segmentation approach has a potential to support concept location?

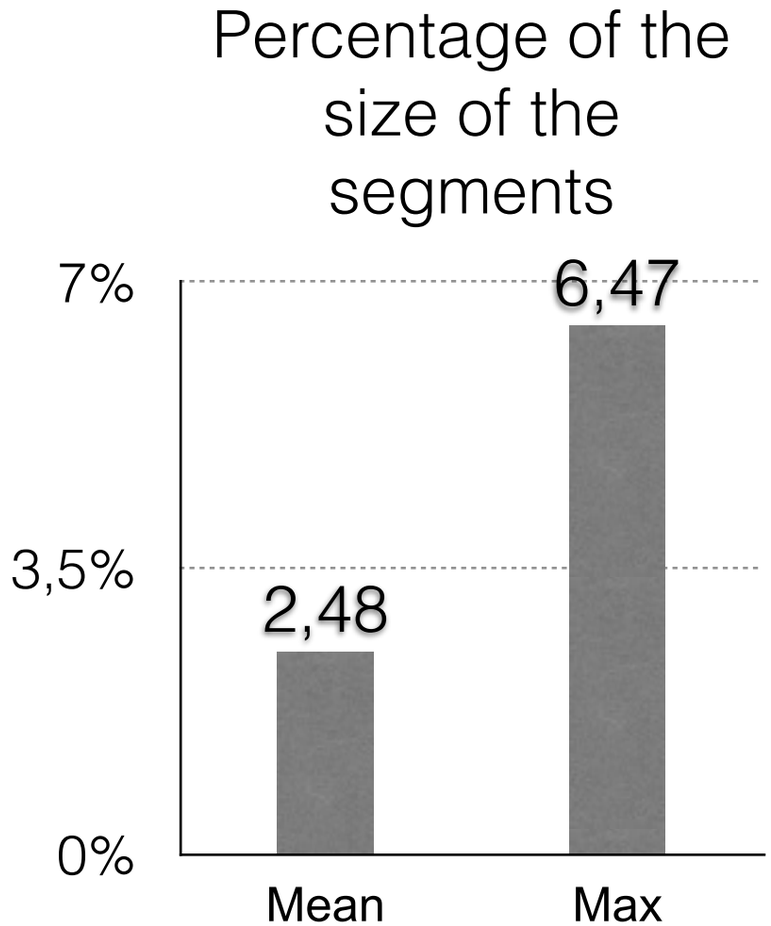


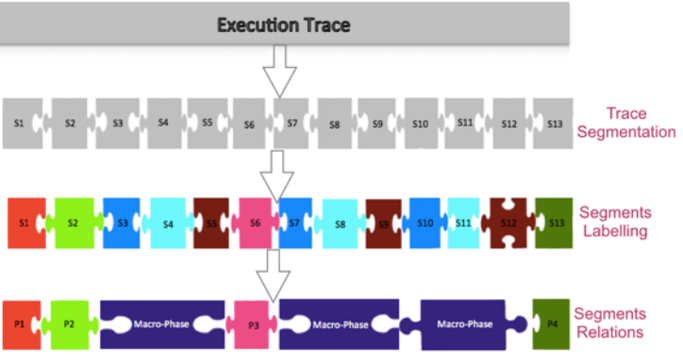


Empirical Results

- RQ1: Does our trace segmentation approach has a potential to support concept location?

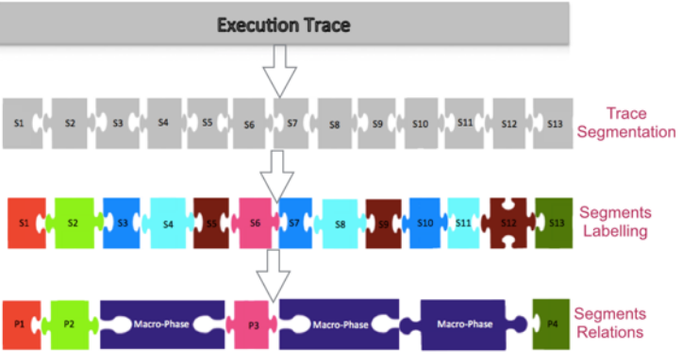






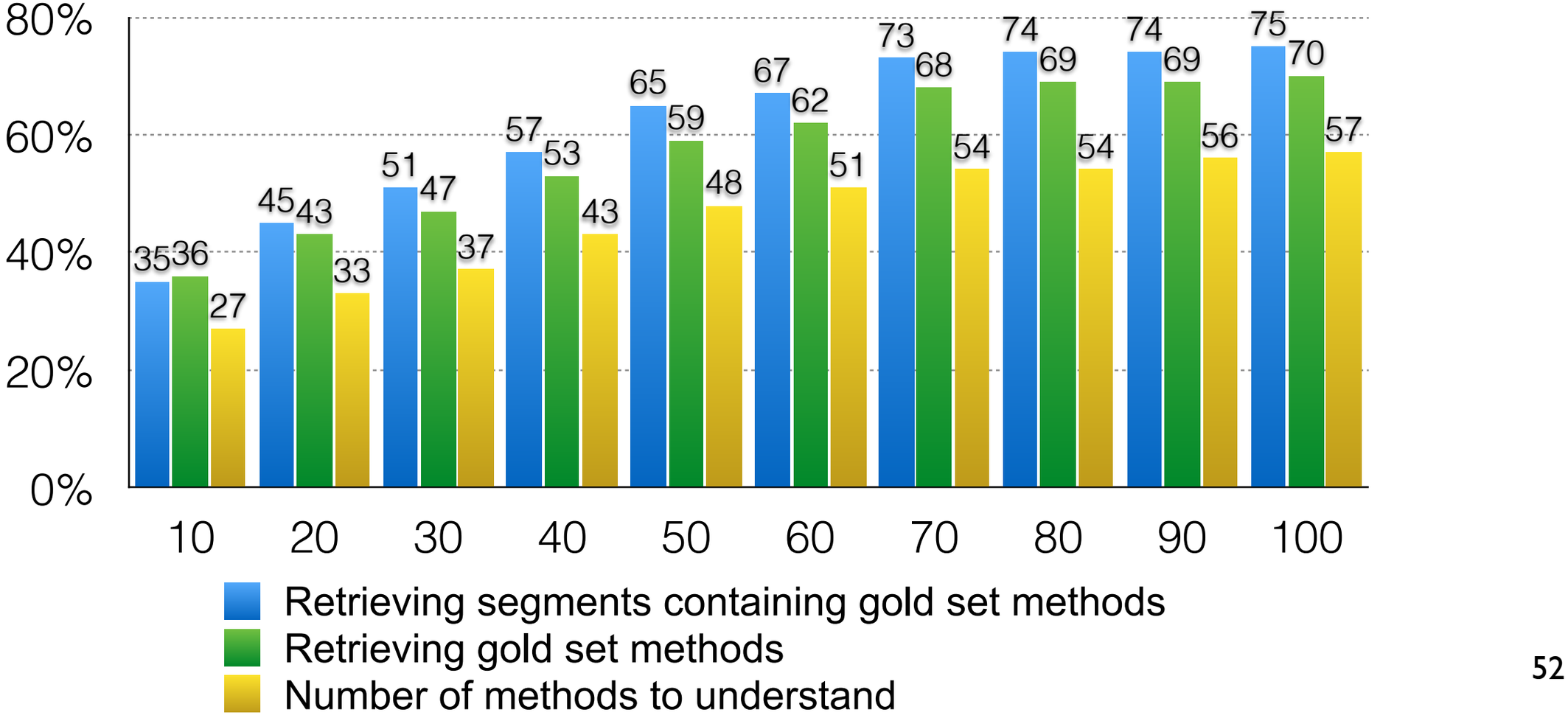
Empirical Results

- RQ2: To what extent does our approach support concept location tasks if used as a standalone technique?
  - Title of the bug report;
  - Labels of the segments;
  - FCA lattice.



Empirical Results

- RQ2: To what extent does our approach support concept location tasks if used as a standalone technique?

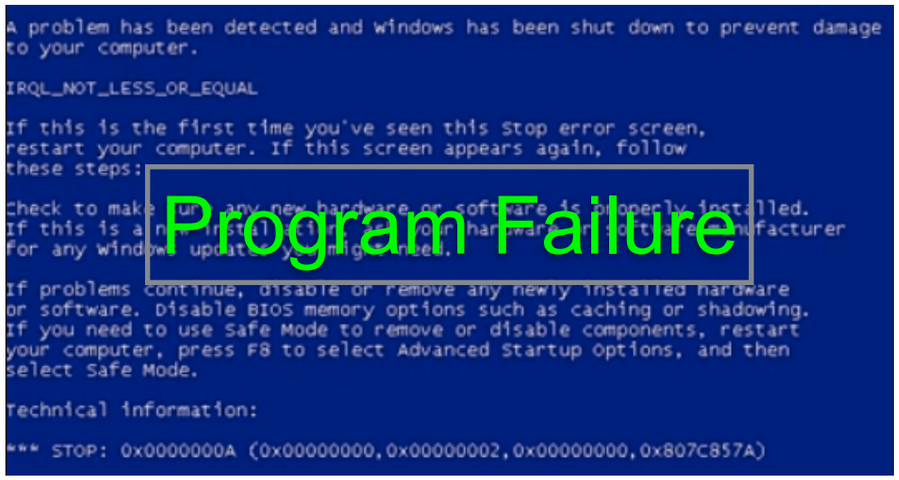


Outline

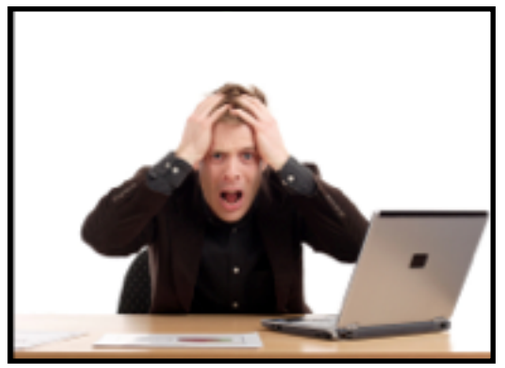
- Context
- Problem Statement
- Trace Segmentation
- Segments Merging
- Segments Labelling
- Segments Relations
- Usefulness Evaluation
- Conclusion and Future Work

Conclusion

- A typical scenario in which concept location takes part:

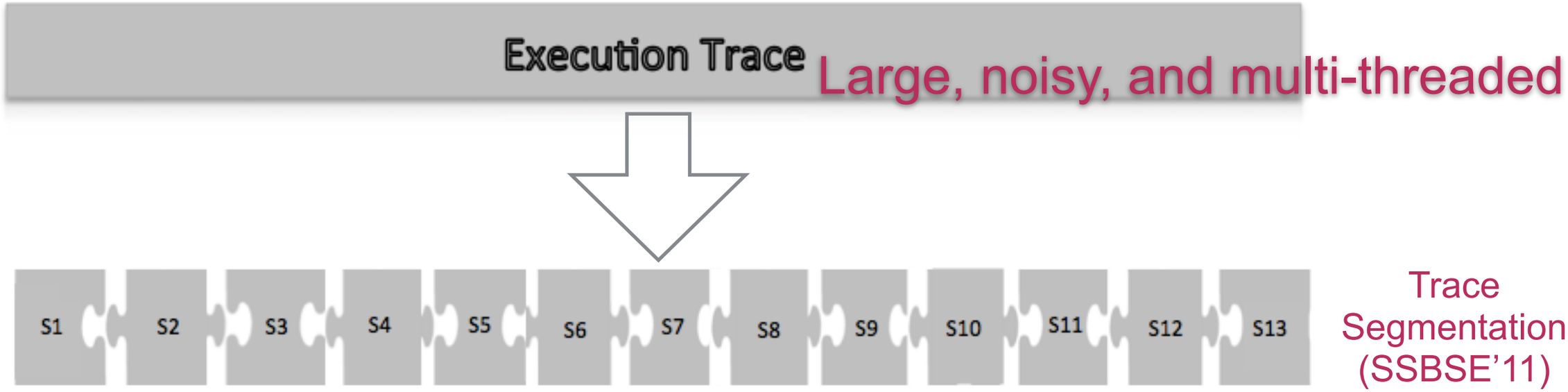


Execution Trace



Execution Trace

Large, noisy, and multi-threaded


Execution Trace
Large, noisy, and multi-threaded
S1
S2
S3
S4
S5
S6
S7
S8
S9
S10
S11
S12
S13
Trace Segmentation (SSBSE'11)

Large, noisy, and multi-threaded

Dynamic Programming (DP) Approach

Trace Segmentation (SSBSE'11)

- Example of trace segmentation using DP.

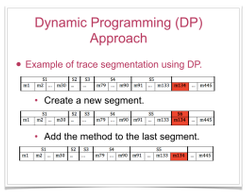
S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

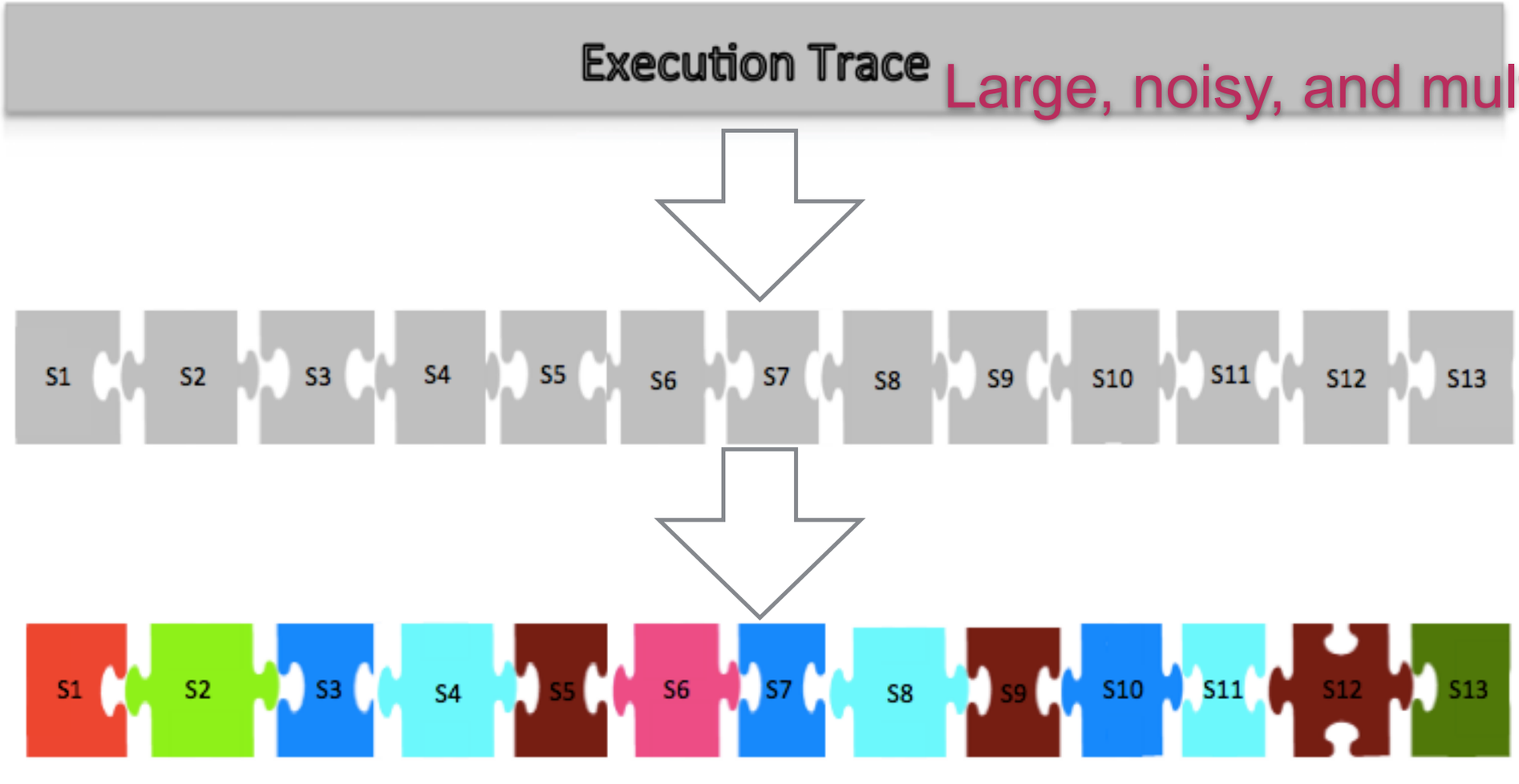
- Create a new segment.

S1				S2	S3	S4			S5			S6		
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445

- Add the method to the last segment.

S1				S2	S3	S4			S5					
m1	m2	...	m30	...	...	m79	...	m90	m91	...	m133	m134	...	m445





Large, noisy, and multi-threaded

Trace Segmentation (SSBSE'11) ✓

Labeling Segments (WCRE'12, JSEP14)

Execution Trace

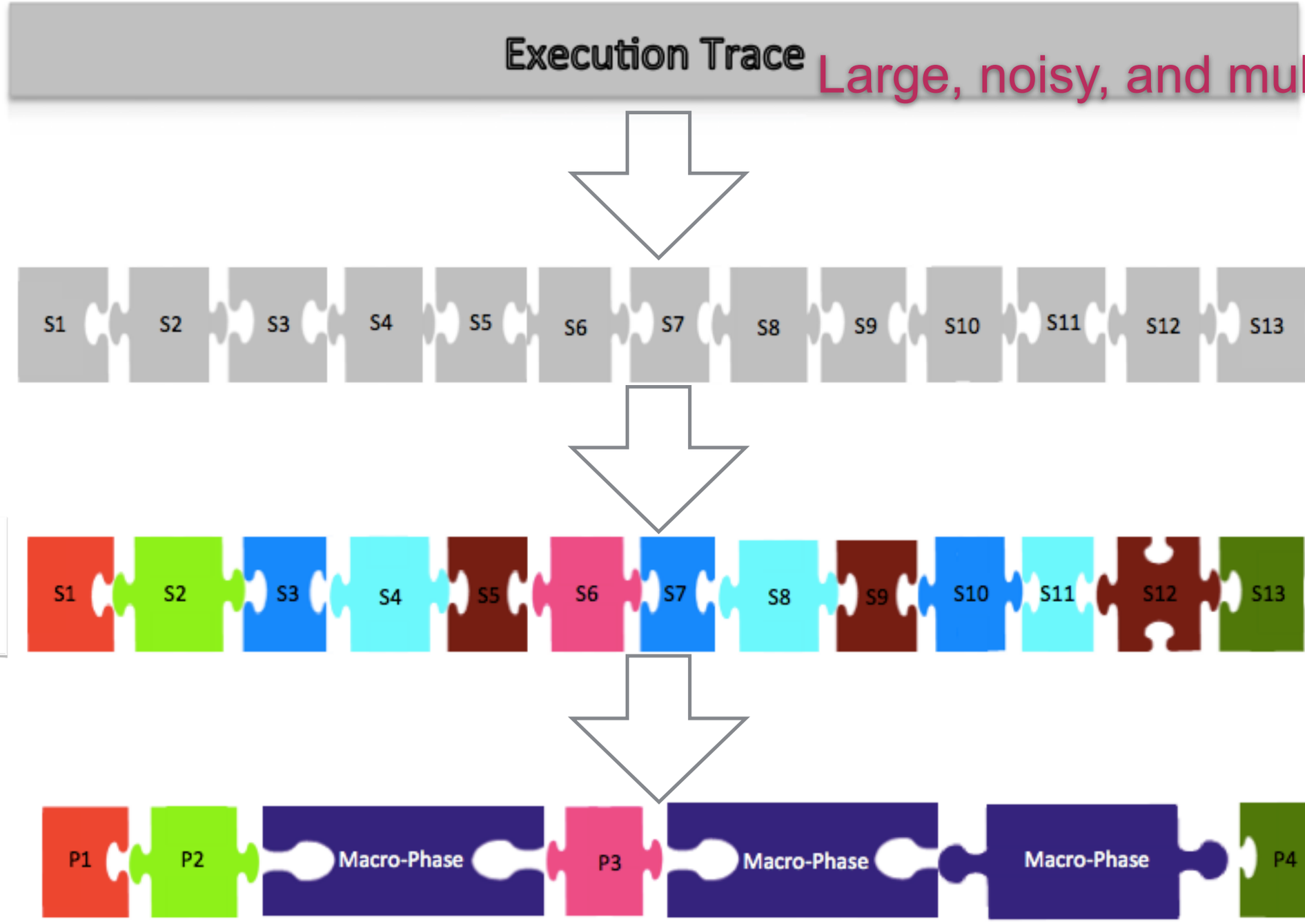
Large, noisy, and multi-threaded

Segments Labelling

- Source of information: terms contained in the signature of methods.
- Hypothesis: A term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- Ranks the terms of the segment by *TF-IDF* and keeps the topmost ones.

Trace Segmentation (SSBSE'11) ✓

Labeling Segments (WCRE'12, JSEP14)



Large, noisy, and multi-threaded

Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.
- Create a new segment.
- Add the method to the last segment.

Trace Segmentation (SSBSE'11) ✓

Segments Labelling

- Source of information: terms contained in the signature of methods.
- Hypothesis: A term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- Ranks the terms of the segment by *TF-IDF* and keeps the topmost ones.

Labeling Segments (WCRE'12, JSEP14) ✓

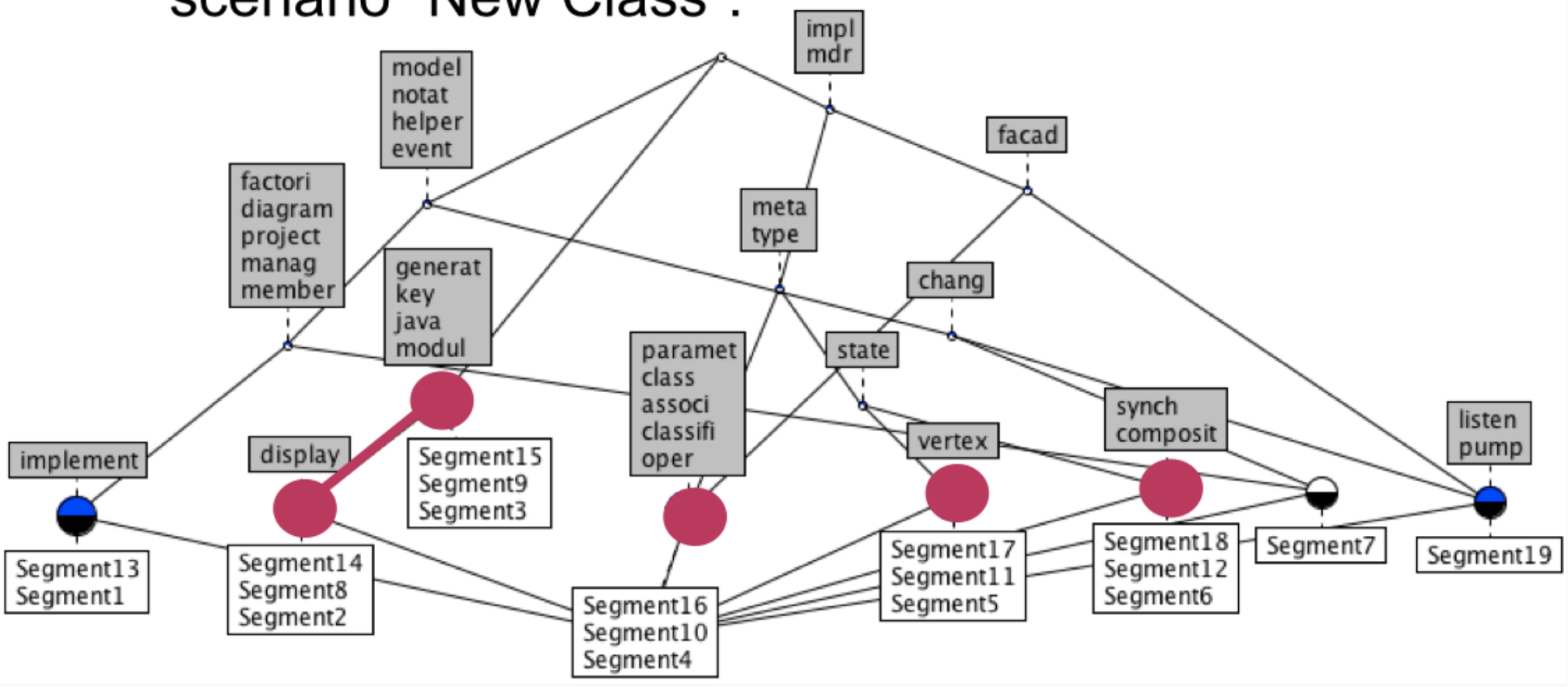
Relating Segments (WCRE'12, JSEP14) ✓

Execution Trace

Large, noisy, and multi-threaded

Segments Relations

- FCA lattice for the execution trace of the scenario “New Class”.



Trace Segmentation (SSBSE’11) ✓

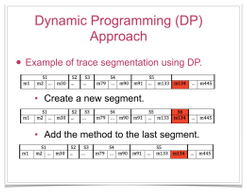
Labeling Segments (WCRE’12, JSEP14)

✓

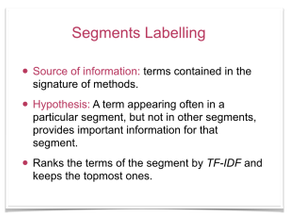
Relating Segments (WCRE’12, JSEP14)

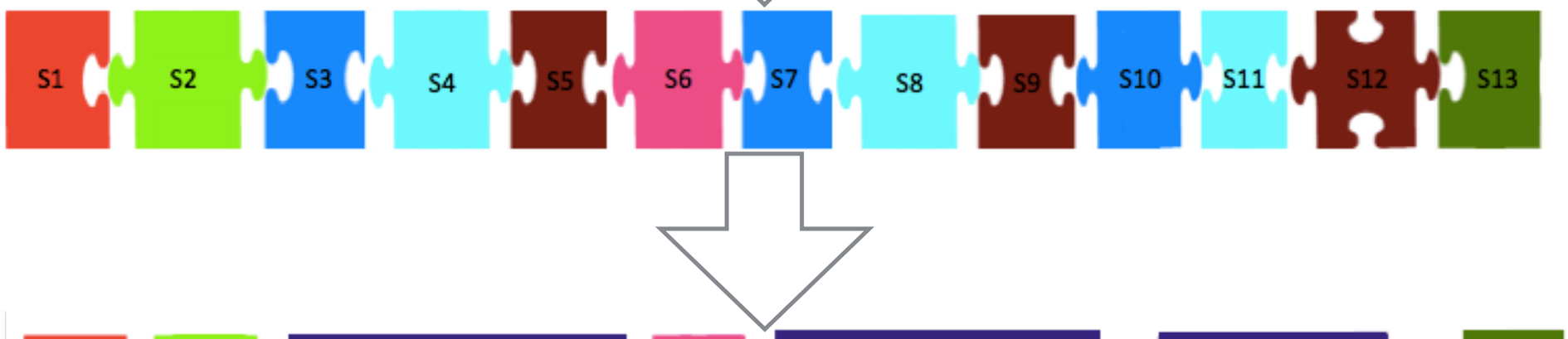
Execution Trace

Large, noisy, and multi-threaded

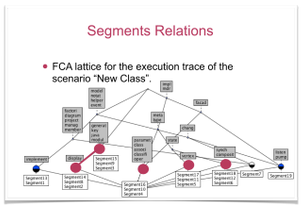


Trace Segmentation (SSBSE'11) ✓





Labeling Segments (WCRE'12, JSEP14) ✓



Relating Segments (WCRE'12, JSEP14) ✓

Execution Trace

Large, noisy, and multi-threaded

Usefulness Evaluation

- During maintenance, developers are interested to understand some segments of a trace that implement some concepts of interest.
- Our approach groups these concepts in few segments.
- Our approach guide developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.

Trace Segmentation (SSBSE'11) ✓

Labeling Segments (WCRE'12, JSEP14)

✓

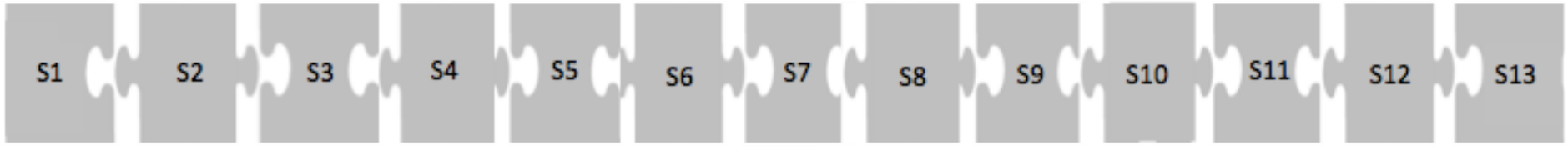
Relating Segments (WCRE'12, JSEP14)

✓

Usefulness Evaluation (JSEP14)

Execution Trace

Large, noisy, and multi-threaded



Trace Segmentation (SSBSE'11) ✓

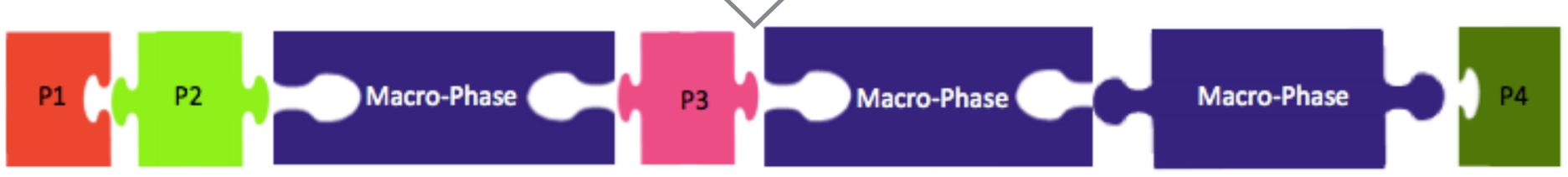
Dynamic Programming (DP) Approach

- Example of trace segmentation using DP.
- Create a new segment.
- Add the method to the last segment.

Labeling Segments (WCRE'12, JSEP14) ✓

Segments Labelling

- Source of information: terms contained in the signature of methods.
- Hypothesis: A term appearing often in a particular segment, but not in other segments, provides important information for that segment.
- Ranks the terms of the segment by *TF-IDF* and keeps the topmost ones.



Relating Segments (WCRE'12, JSEP14) ✓

Segments Relations

- FCA lattice for the execution trace of the scenario "New Class".

Usefulness Evaluation

- During maintenance, developers are interested to understand some segments of a trace that implement some concepts of interest.
- Our approach groups these concepts in few segments.
- Our approach guide developers towards segments that implement the concepts to maintain and reduce the number of methods to investigate.

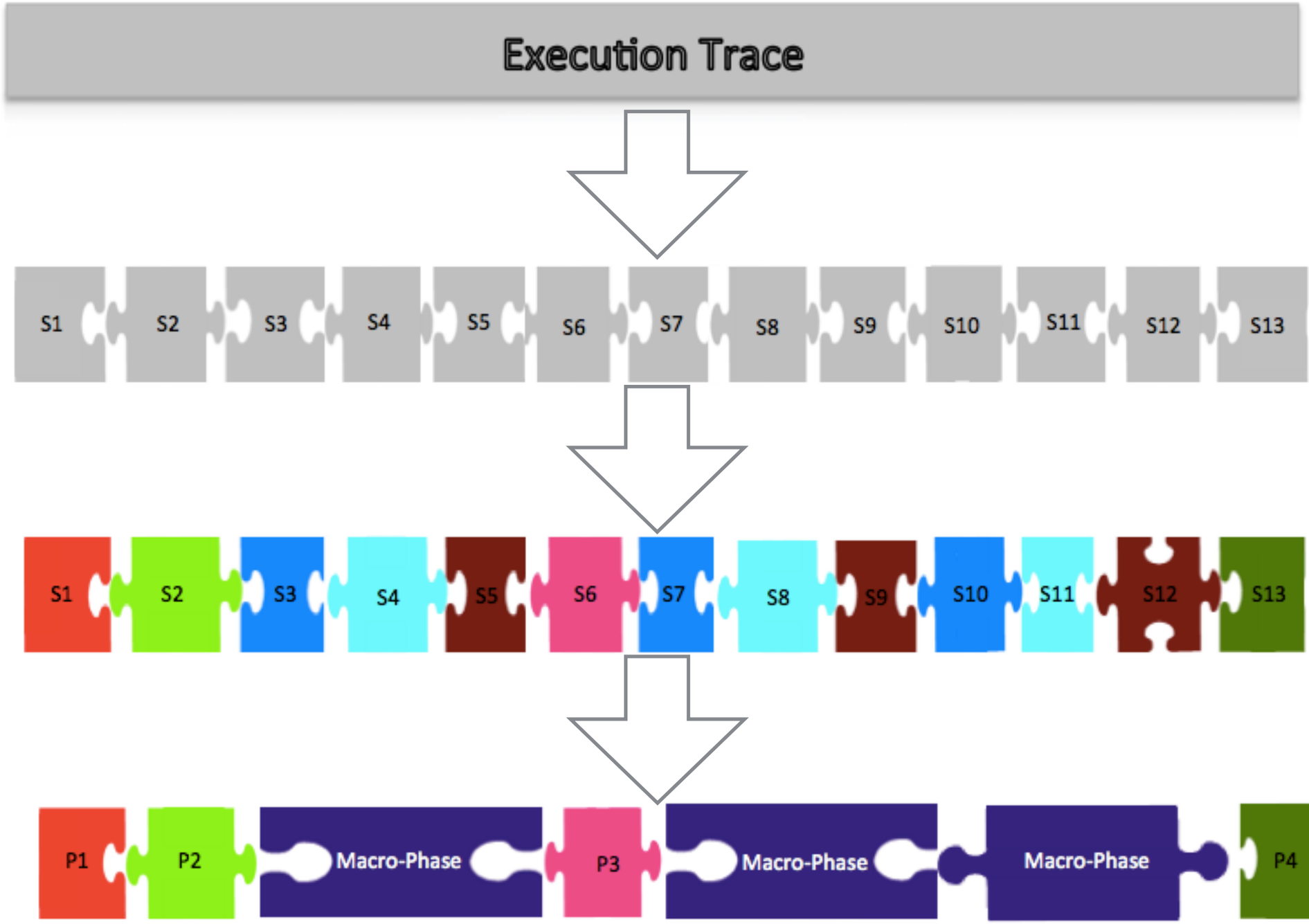
Usefulness Evaluation (JSEP14) ✓

Thesis

Identify concepts and facilitate the analysis of large execution traces for maintenance tasks.

Future Work

- A tool to visualize the identified relations among segments.
- Adapting our approach to online labelling of traces while they are being generated.
- Trace segmentation of distributed systems.



Trace Segmentation (SSBSE'11) ✓

Labeling Segments (WCRE'12, JSEP14)

✓

Relating Segments (WCRE'12, JSEP14)

✓

Usefulness Evaluation (JSEP14)

✓

References

- Ian Sommerville. Software Engineering. Addison-Wesley, 6th edition, 2000.
- Dehaghani S. M. H et Hajrahimi N., 2013 . Which factors affect software projects maintenance cost more? Acta Informatica Medica, 21, 63–66.
- REISS, S. P. et RENIERIS, M. (2001). Encoding program executions.(ICSE), 221–230.
- HAMOU-LHADJ, A. et LETHBRIDGE, T. (2006). Summarizing the content of large traces to facilitate the understanding of the behaviour of a software system. (ICPC), 181–190.
- HAMOU-LHADJ, A., BRAUN, E., AMYOT, D. et LETHBRIDGE, T. (2005). Recovering behavioral design models from execution traces.(CSMR), 112–121.
- PIRZADEH, H. et HAMOU-LHADJ, A. (2011). A novel approach based on gestalt psychology for abstracting the content of large execution traces for program comprehension. (ICECCS), 221–230.

References

- ASADI, F., ANTONIOL, G. et GUÉHÉNEUC, Y.-G. (2010). Concept locations with genetic algorithms: A comparison of four distributed architectures. (SSBSE), 153–162.
- DE LUCIA, A., DI PENTA, M., OLIVETO, R., PANICHELLA, A. et PANICHELLA, S. (2012). Using IR methods for labeling source code artifacts: Is it worthwhile? (ICPC), 193–202.
- DIT, B., HOLTZHAUER, A., POSHYVANYK, D. et KAGDI, H. (2013a). A dataset from change history to support evaluation of software maintenance tasks. (MSR), 131–134.