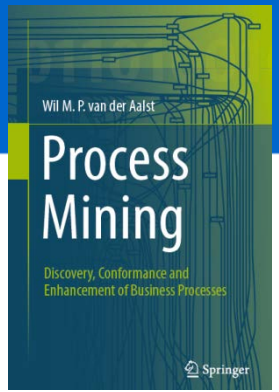
Process Mining: Data Science in Action

On the Representational Bias of Process Mining

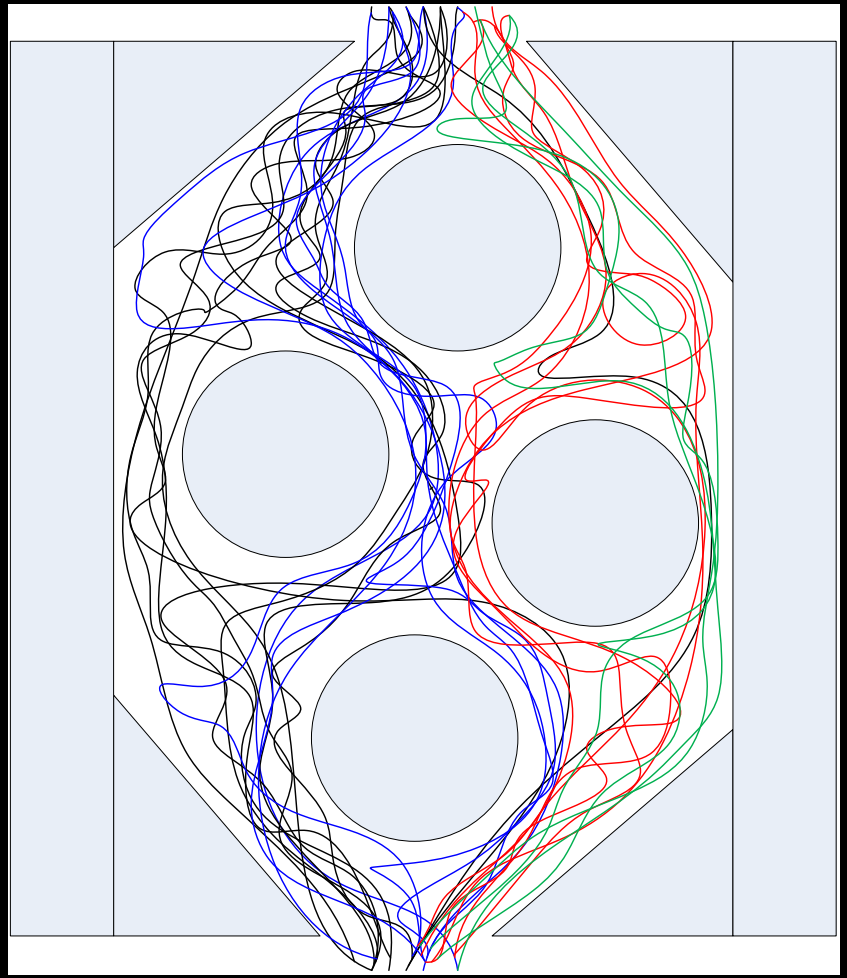
prof.dr.ir. Wil van der Aalst
www.processmining.org

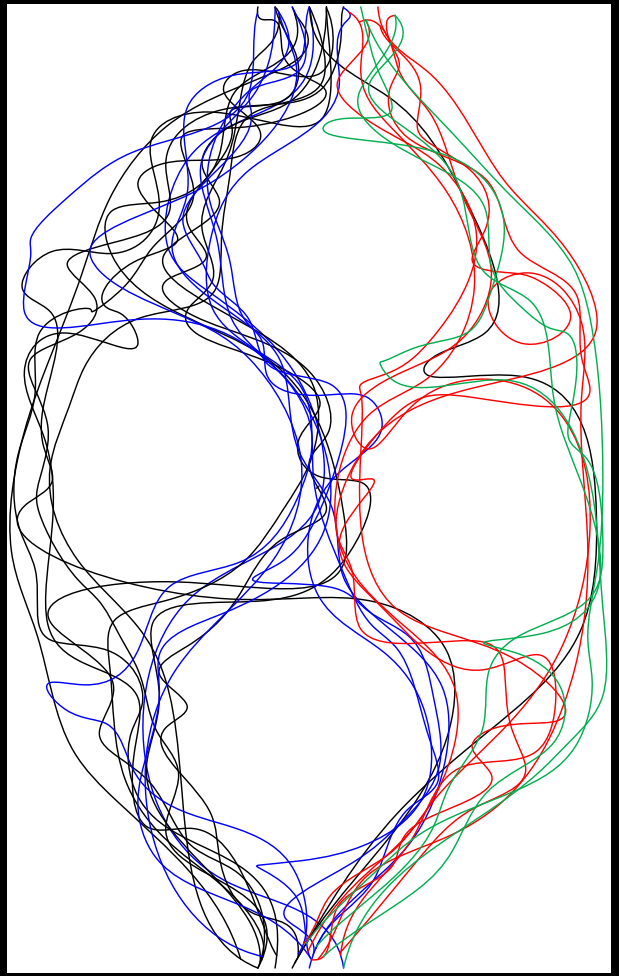


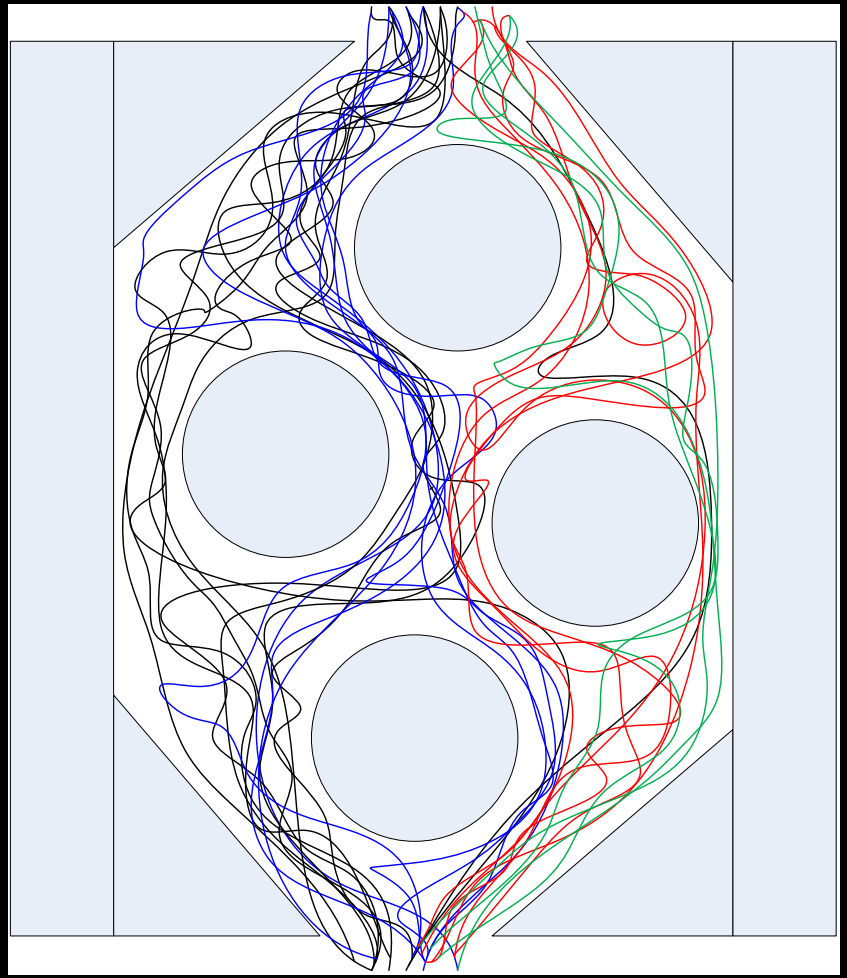
Technische Universiteit
Eindhoven
University of Technology

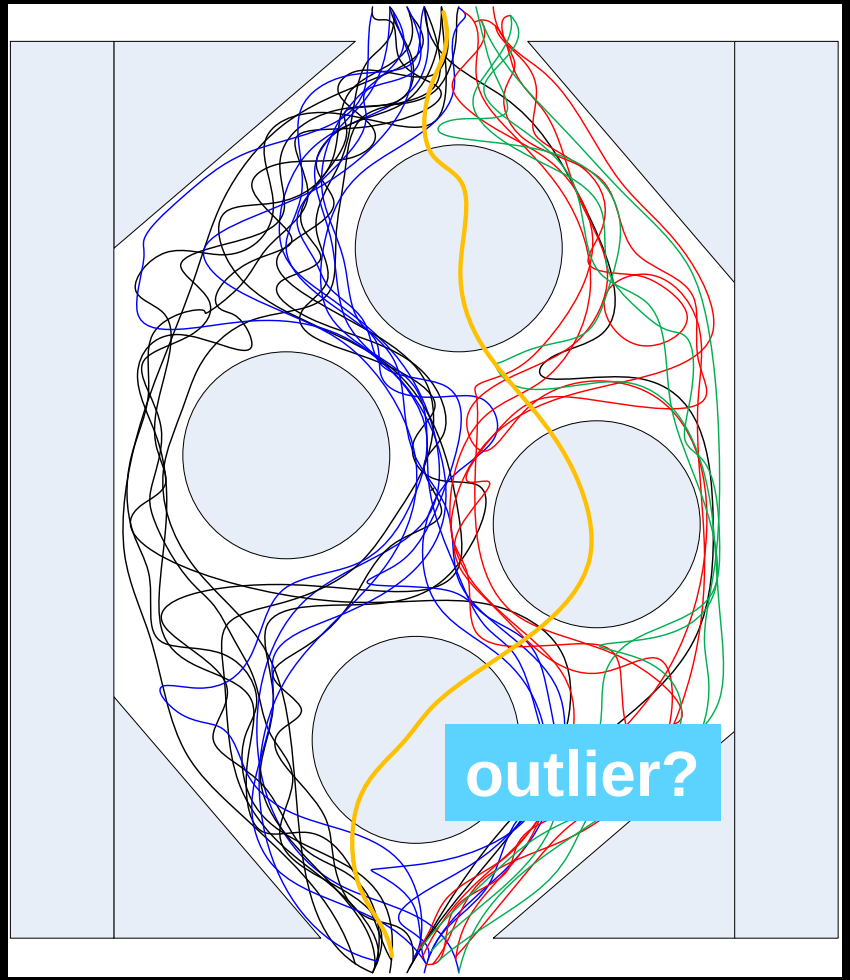
Where innovation starts

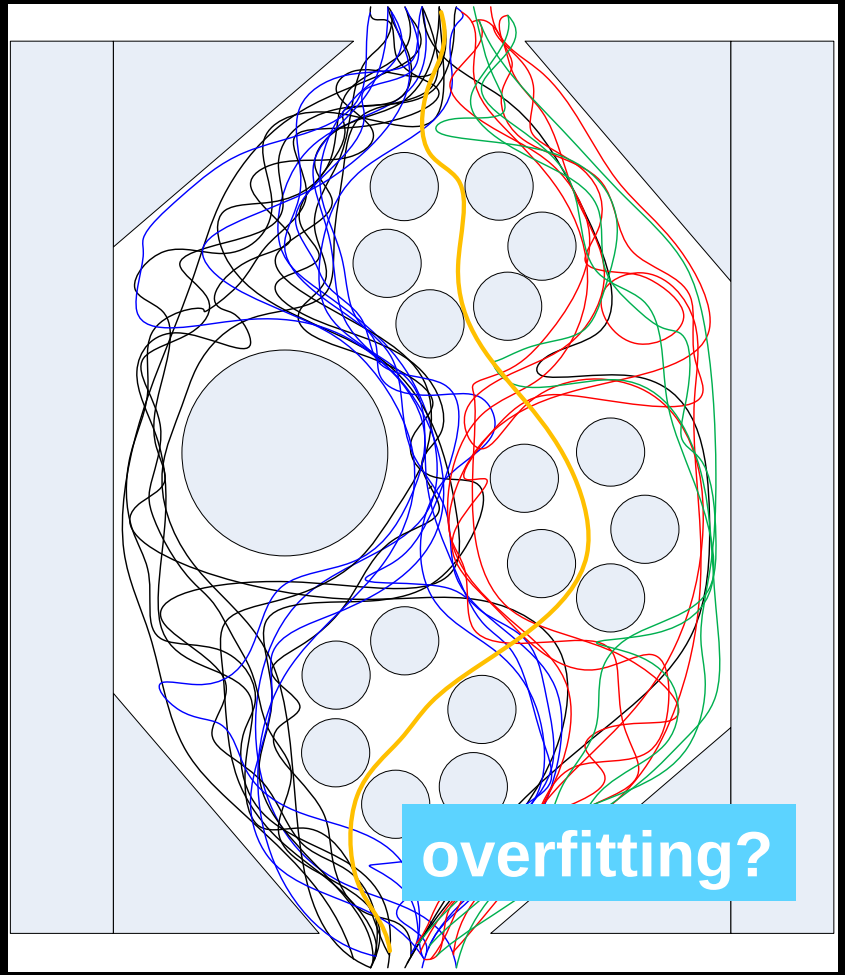

小草对您微微笑
请您把路绕一绕
KEEP OFF GRASS

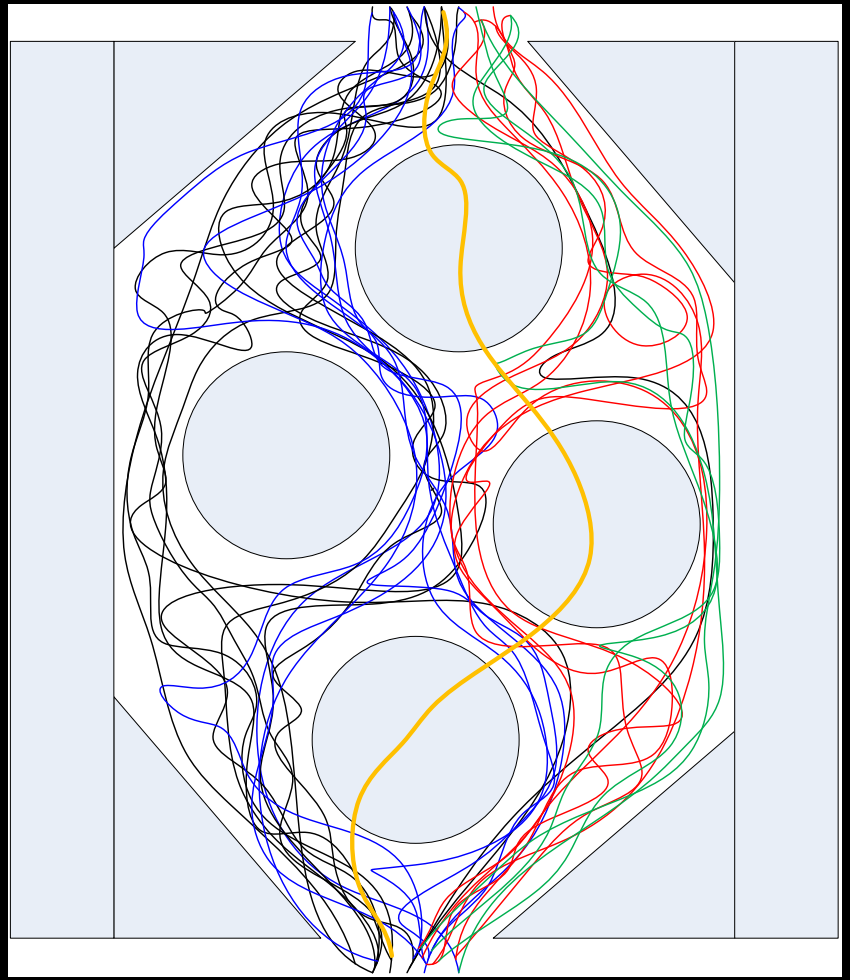


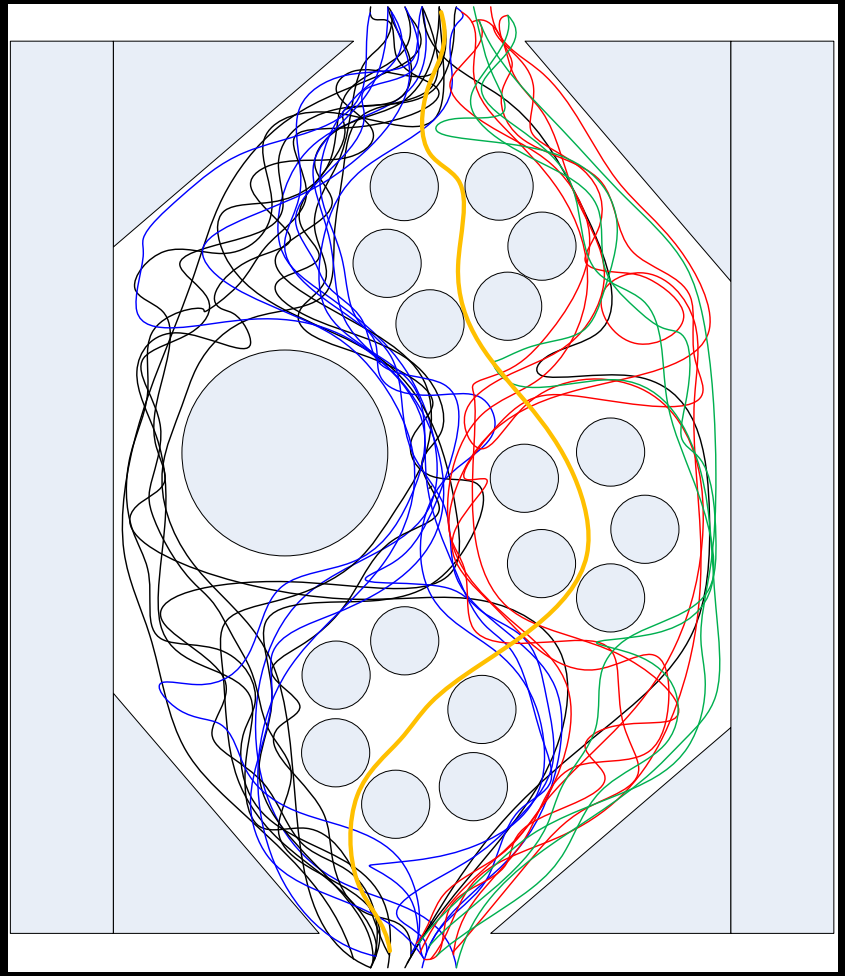


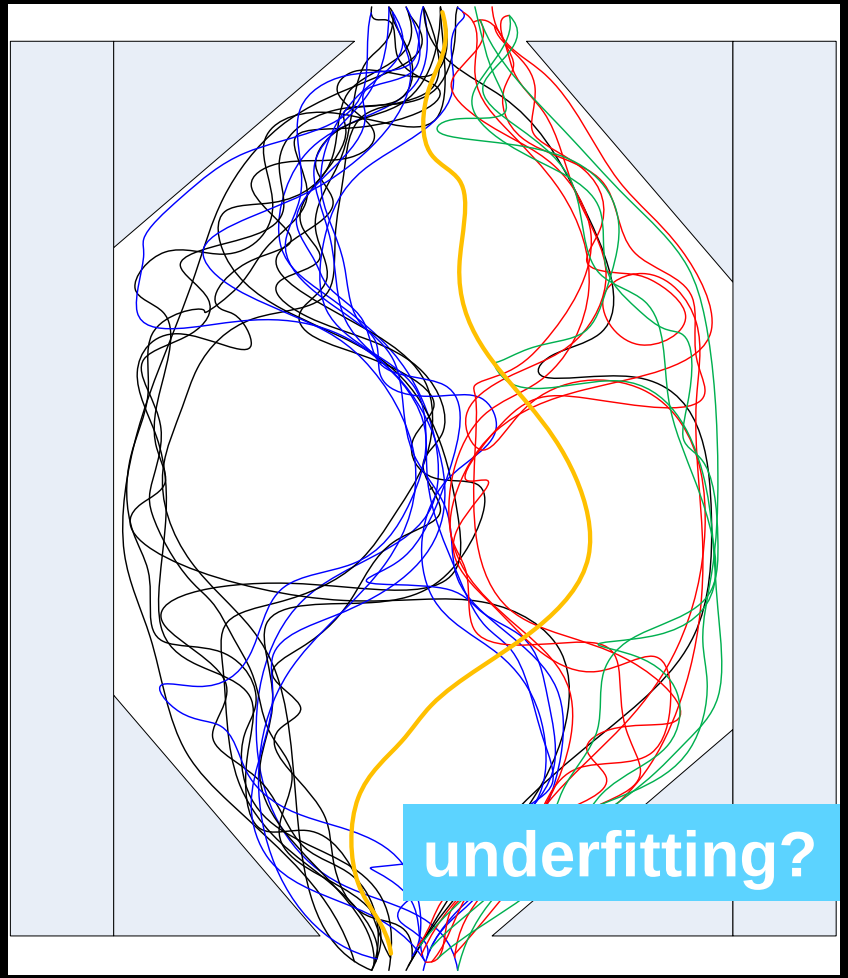


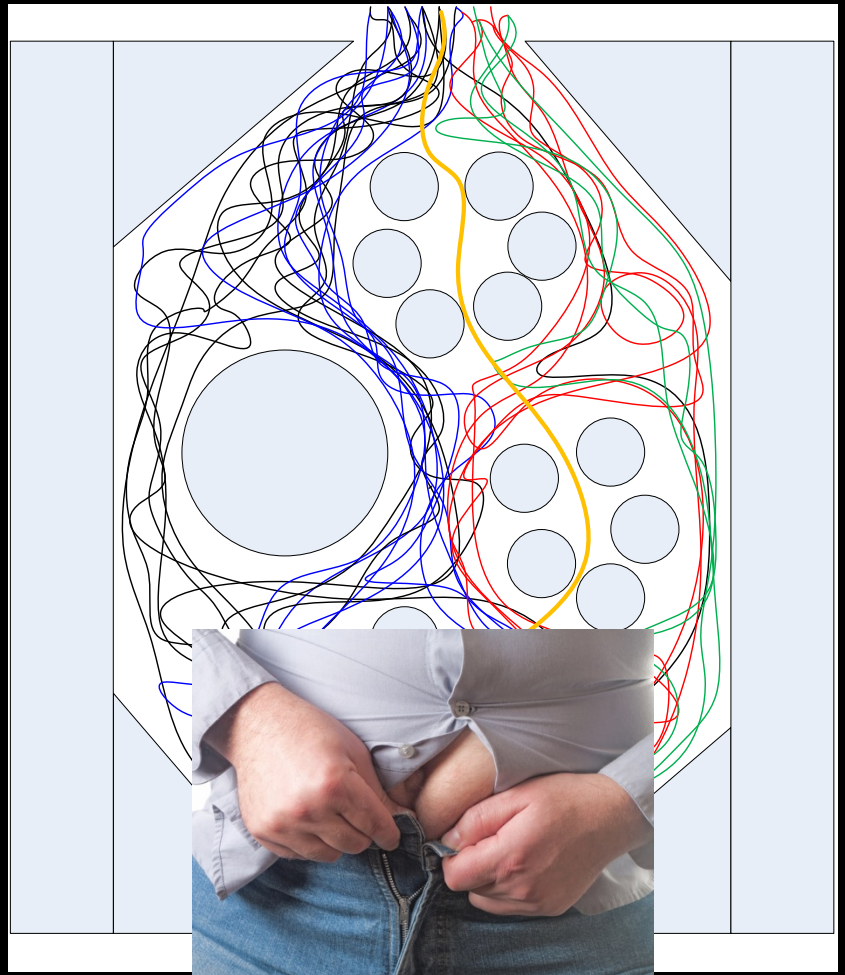

outlier?

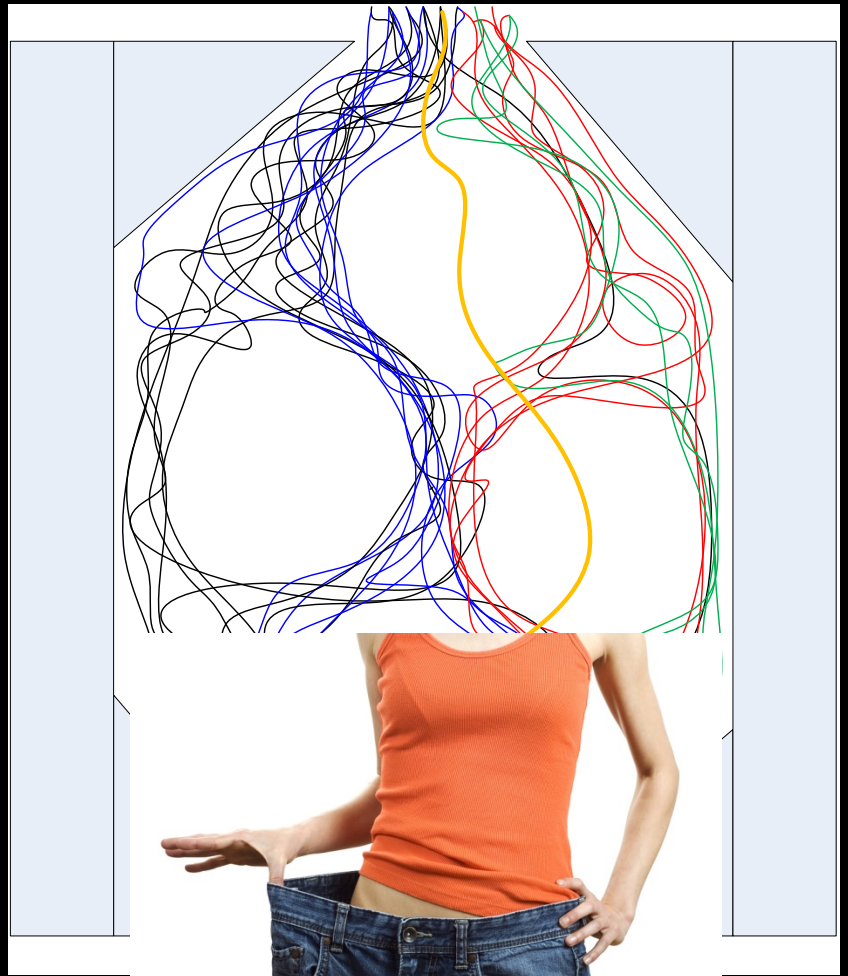

overfitting?

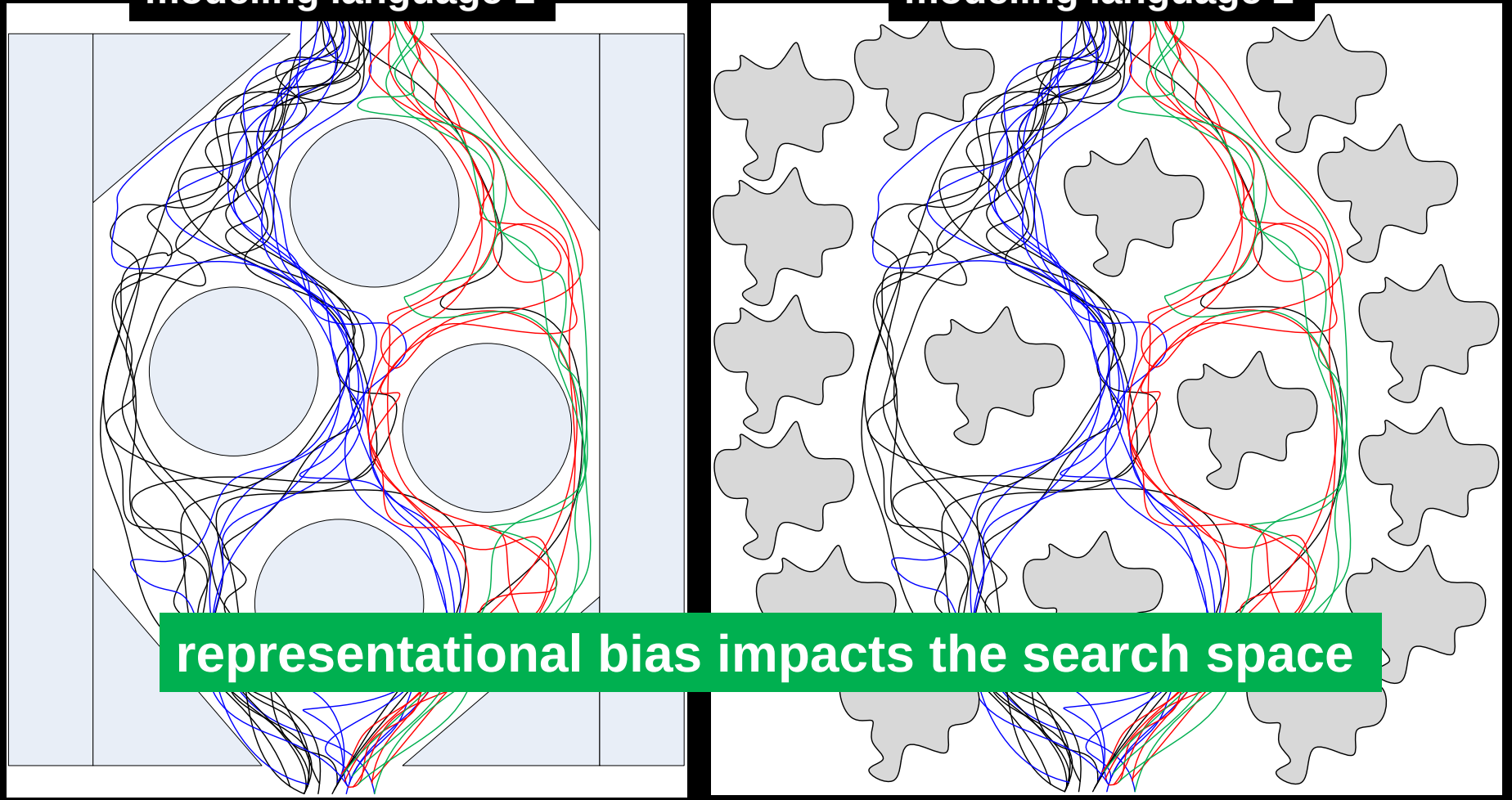




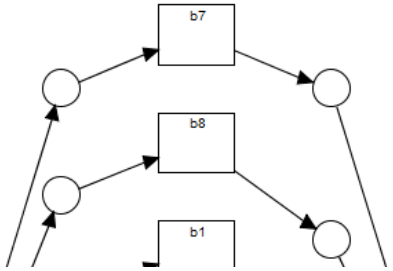

underfitting?

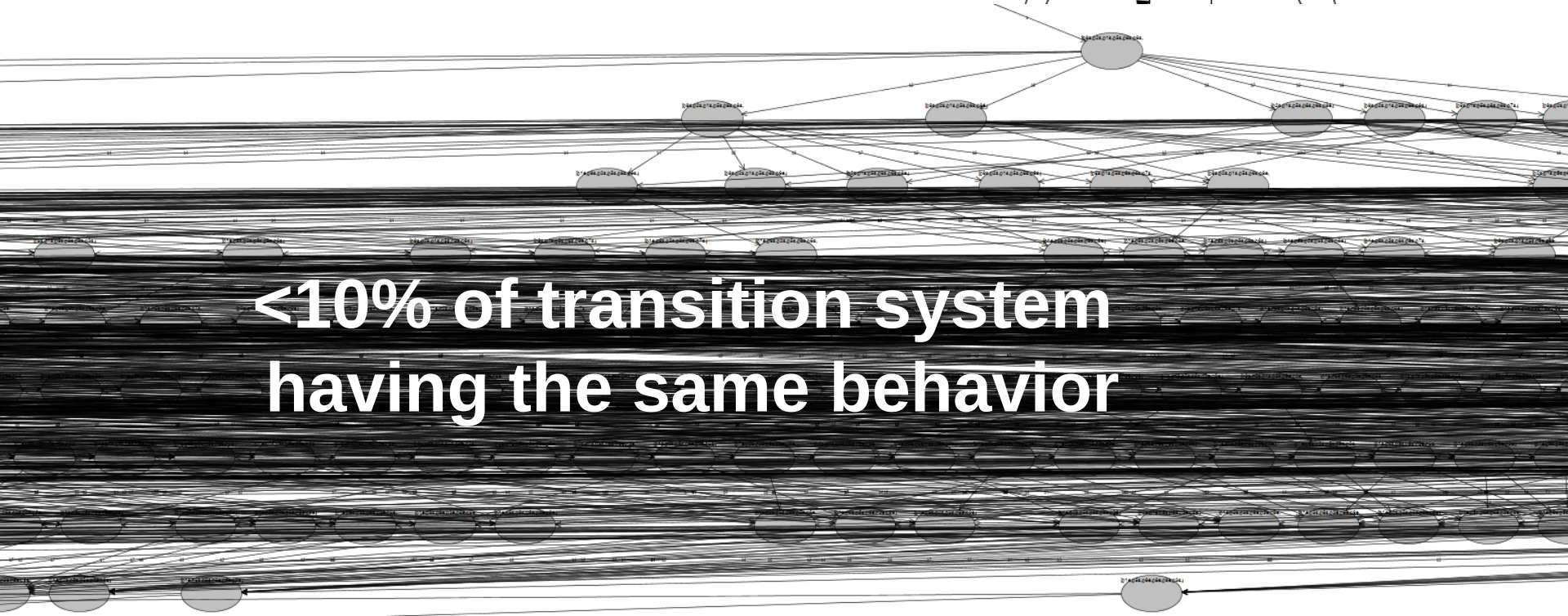




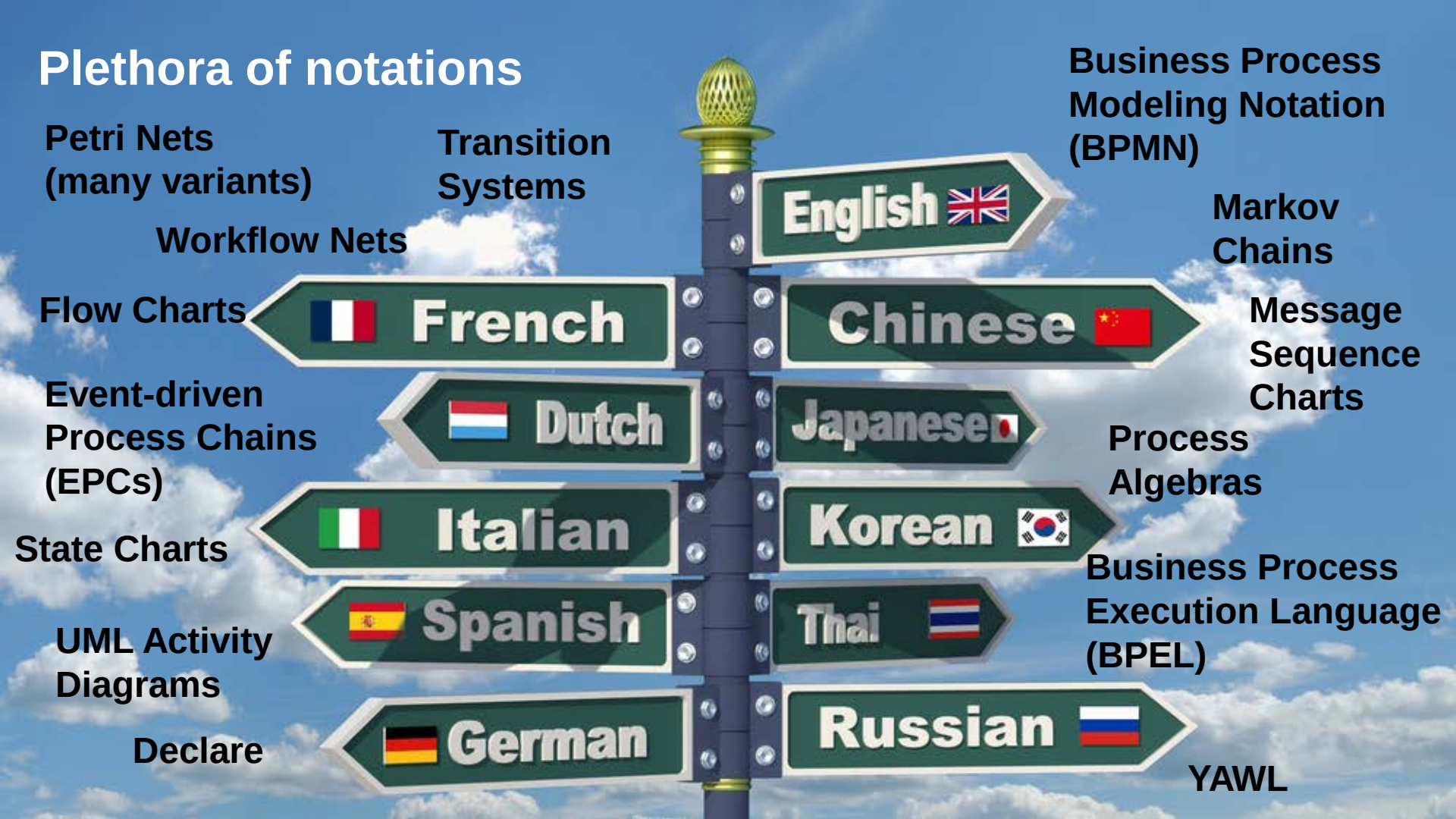

modeling language 1
modeling language 2
representational bias impacts the search space

Representation

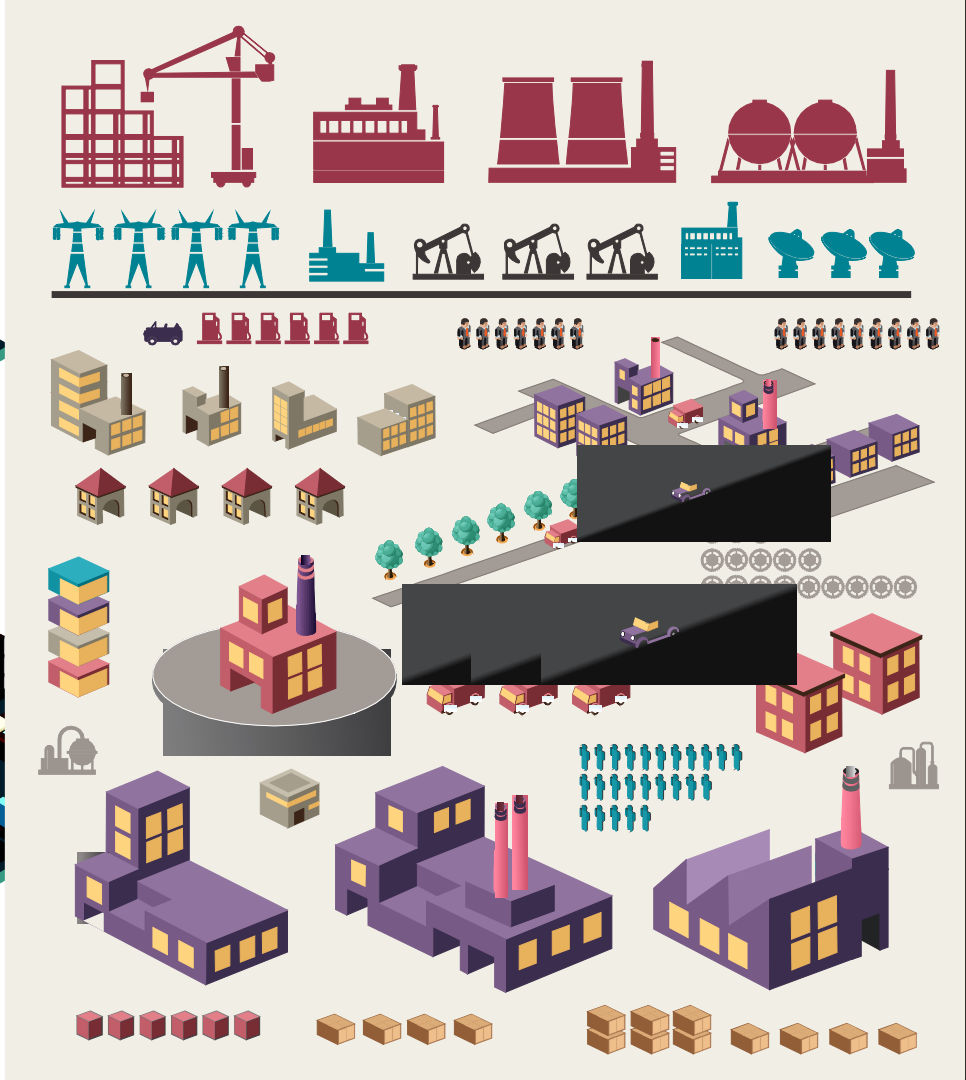


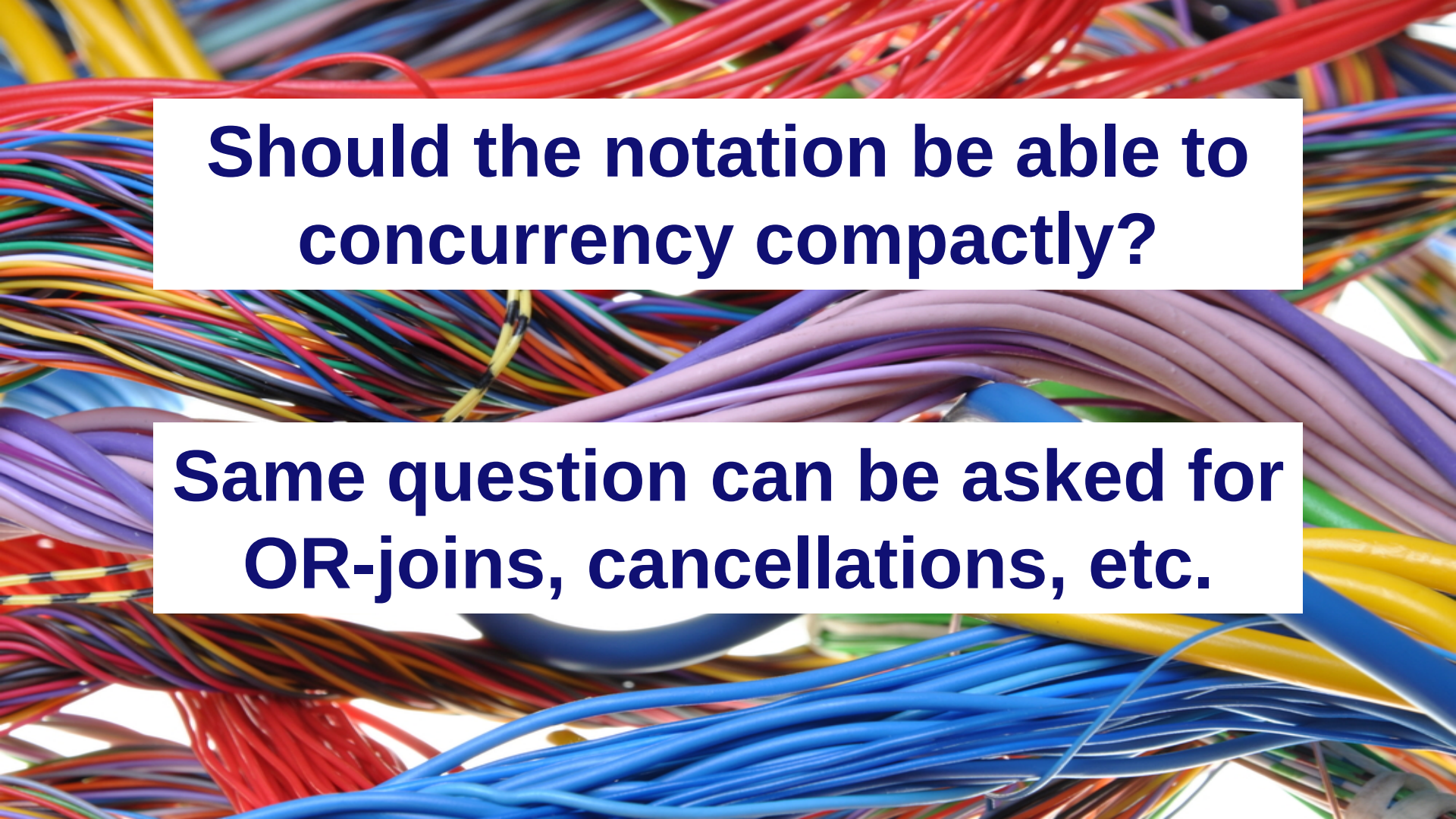


<10% of transition system having the same behavior


Plethora of notations
Petri Nets (many variants)
Transition Systems
Business Process Modeling Notation (BPMN)
Workflow Nets
Markov Chains
Flow Charts
Message Sequence Charts
Event-driven Process Chains (EPCs)
Process Algebras
State Charts
Business Process Execution Language (BPEL)
UML Activity Diagrams
Declare
YAWL
English
French
Chinese
Dutch
Japanese
Italian
Korean
Spanish
Thai
German
Russian

let us consider concurrency in more detail …



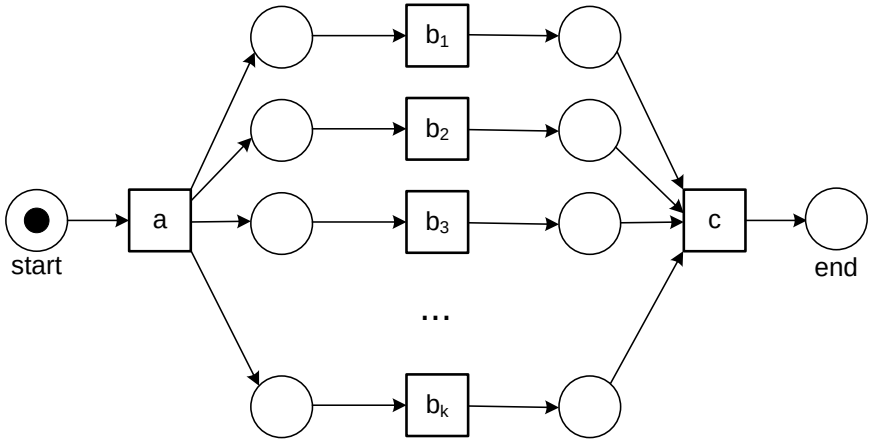

Should the notation be able to concurrency compactly?
Same question can be asked for OR-joins, cancellations, etc.

Question
How many traces in case of concurrency?

- Consider a process model with a start activity and end activity and in the middle ***k*** **parallel activities**.
- How many traces are possible (say ***k*** **= 10**)?
- Does the Alpha algorithm need to see all of these to rediscover the original model?

TU/e

Model with *k* parallel activities



k	number of different traces: k!
1	1
2	2
5	120
10	3628800
20	2432902008176640000

Alpha algorithm only needs to see k(k-1) direct successions.

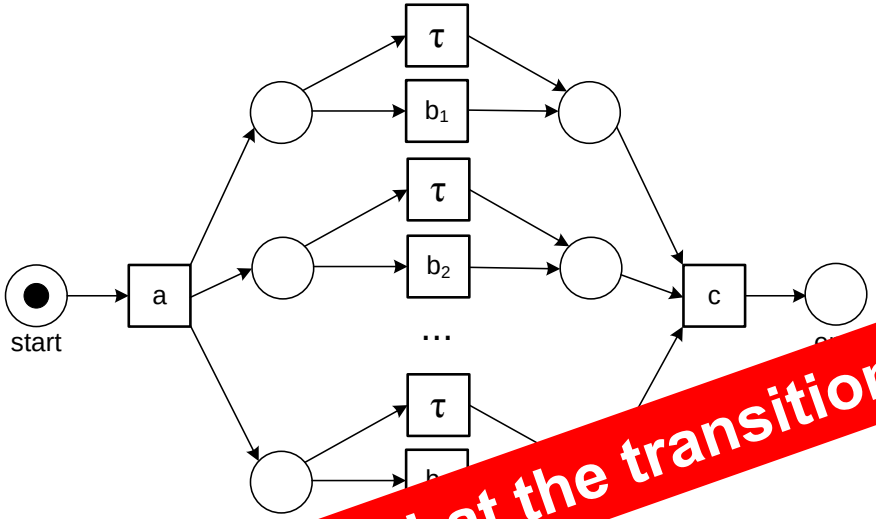
TU/e

Question
How many traces in case of an OR-join?

- Suppose now that the *k* parallel activities are all optional.
- Is the Alpha algorithm able to discover such constructs?
- How many traces are possible?

TU/e

Model with *k* optional activities



k	number of different traces
1	
	5
	326
10	9864101
20	66133133192480800001

imagine what the transition system looks like ....

Alpha algorithm will not discover this model.

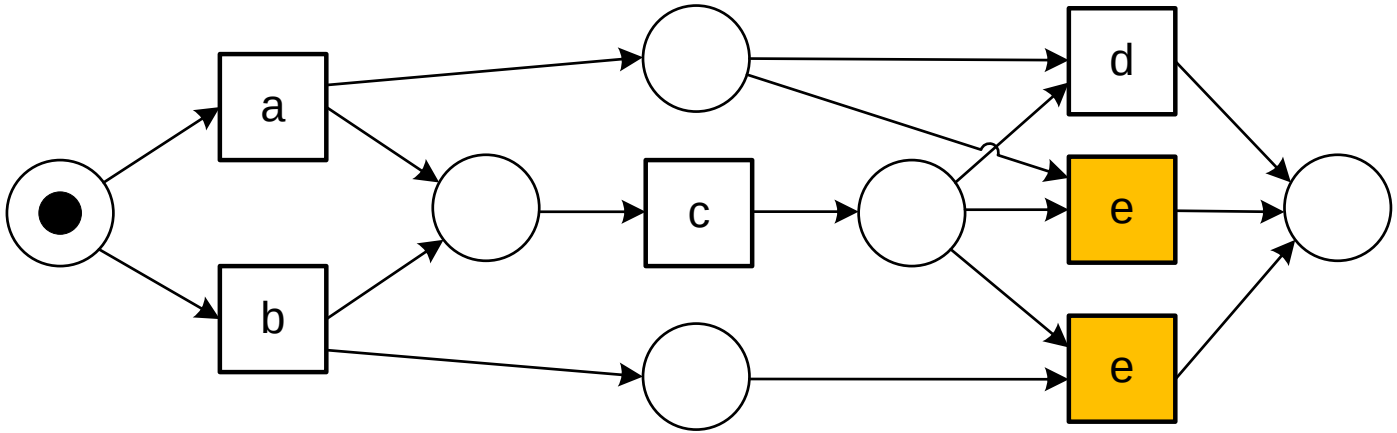
$$\sum_{i=0}^{k} \binom{k}{i} i!$$

TU/e

- **Highly unlikely to discover concurrency or OR-joins if such behaviors cannot be represented easily.**
- **Often the discovery process is guided by the representational bias!**

Assumed Bias: WF-nets with unique labels

$$L = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}, \langle a, c, e \rangle^{20}]$$



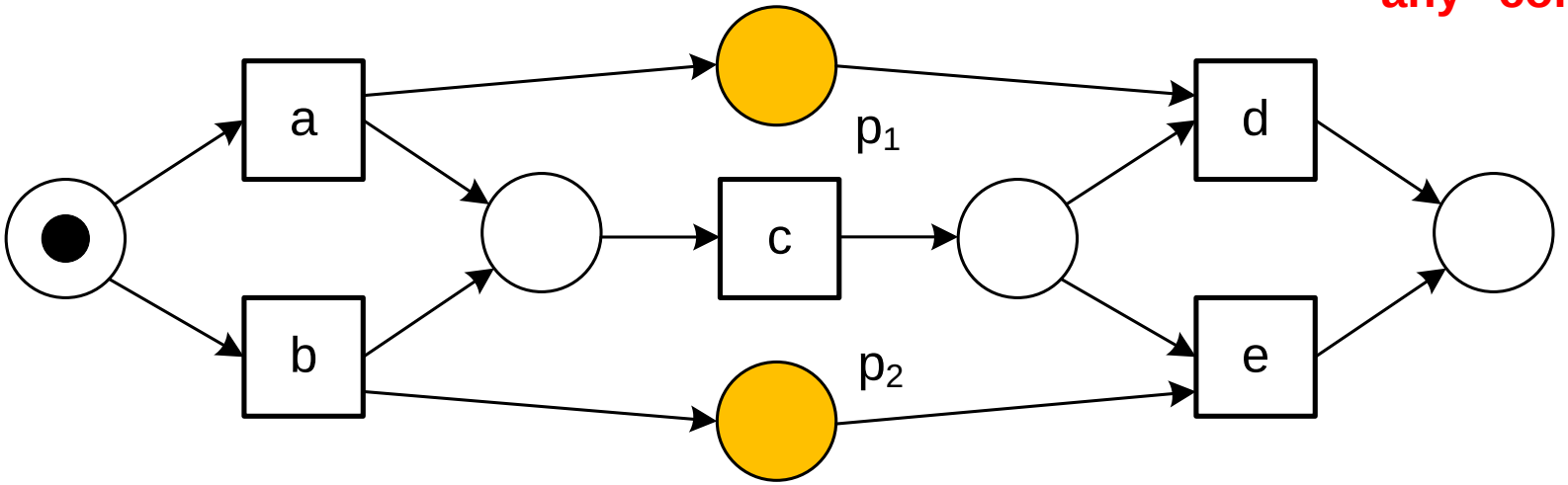
No WF-net with unique labels can be discovered that has precisely this behavior!

TU/e

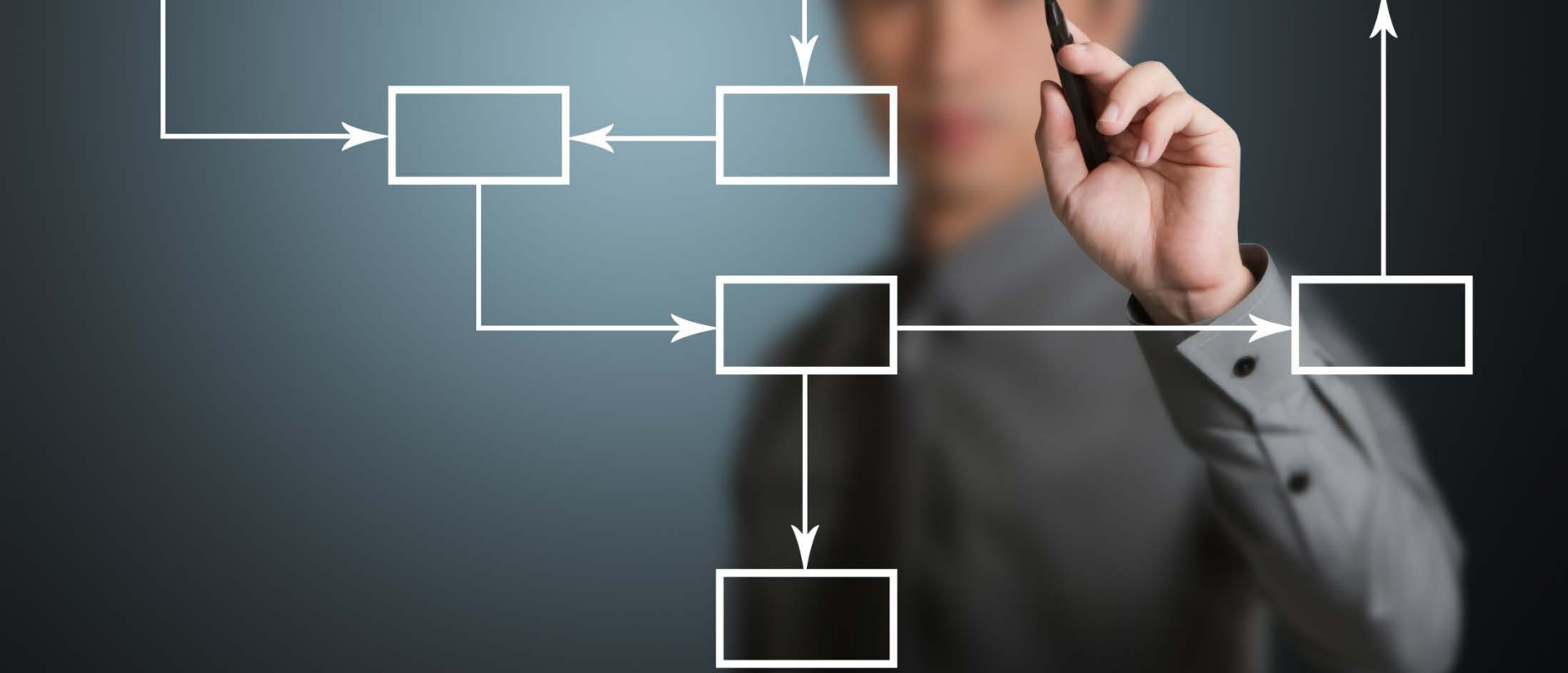
Assumed Bias:
No indirect dependencies

$$L_9 = [\langle a, c, d \rangle^{45}, \langle b, c, e \rangle^{42}]$$

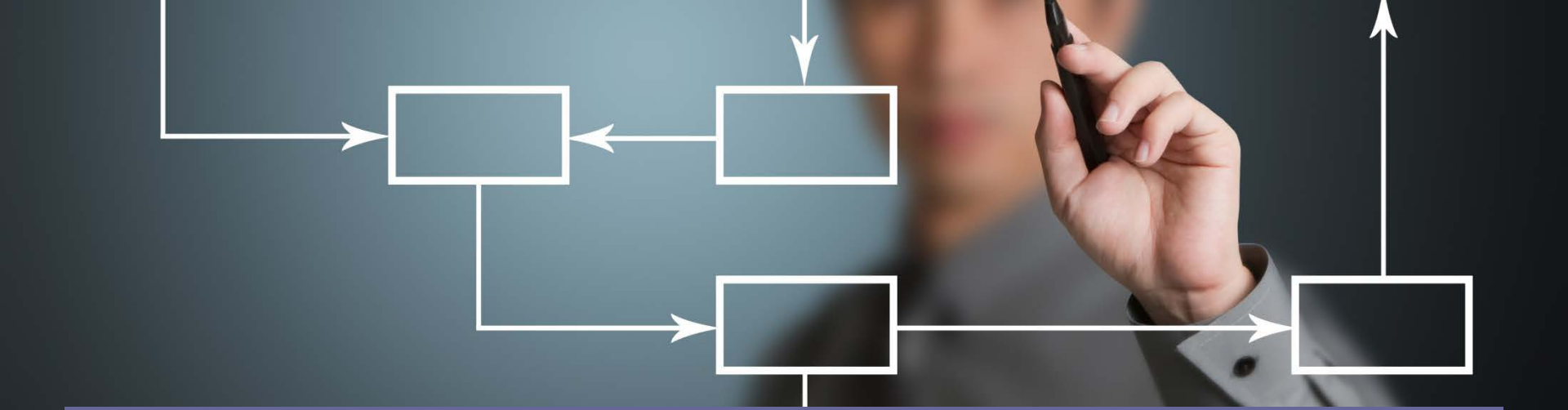
Activities a and b influence the choice for d and e but there is never any "contact"!



TU/e



- **Representational bias matters!**
- **Impacts search space of process discovery**



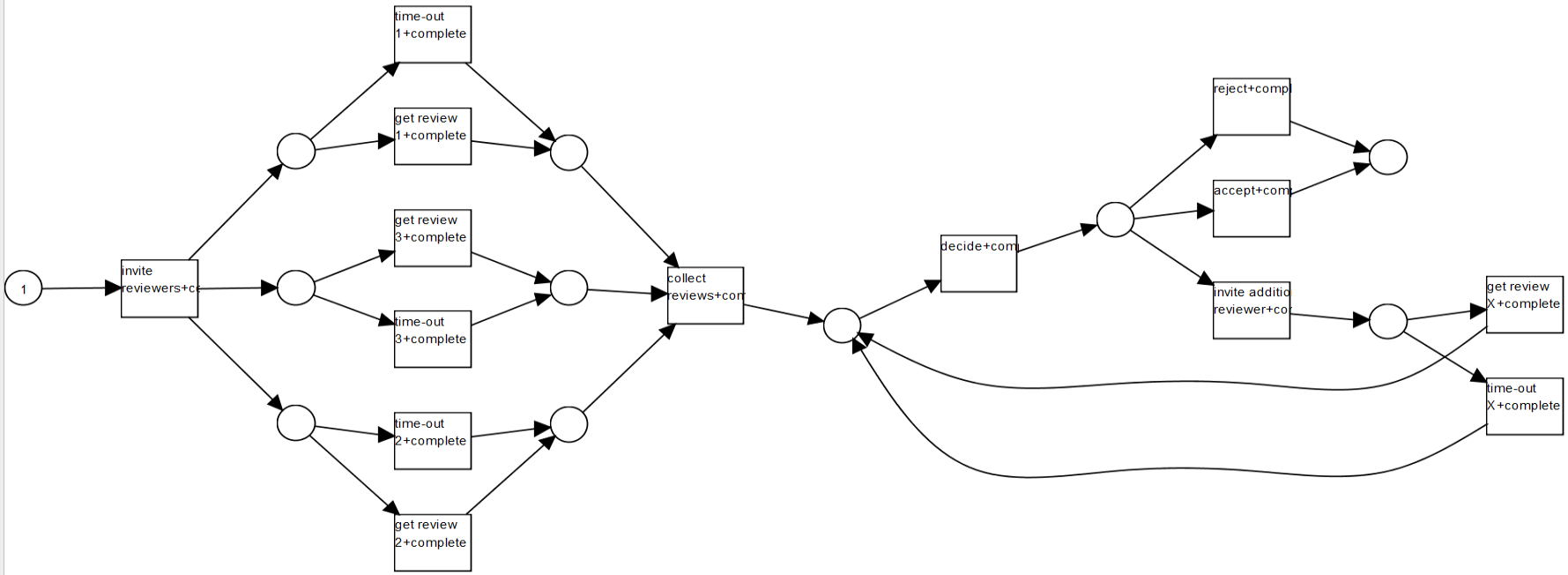
visualization of discovered model

≠

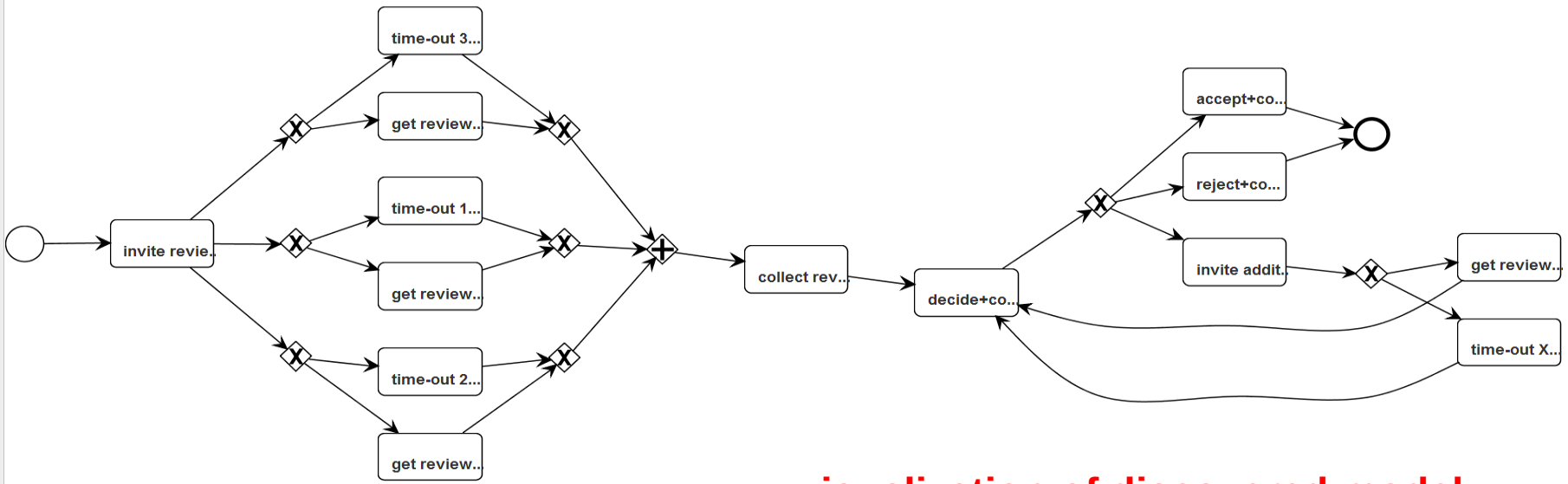
representational bias

Petrinet from reviewing_with_fewer_errors_and_more_data.zip , mined with AlphaMiner

Discovered using Alpha algorithm …



… visualized as a BPMN diagram!



visualization of discovered model
≠
representational bias

More important than visualization …

Why is process discovery so difficult?

- There are no negative examples (i.e., a log shows what has happened but does not show what could not happen).
- Due to concurrency, loops, and choices the search space has a complex structure and the log typically contains only a fraction of all possible behaviors.
- There is no clear relation between the size of a model and its behavior (i.e., a smaller model may generate more or less behavior although classical analysis and evaluation methods typically assume some monotonicity property).
- Careful consideration of representational bias is needed !!

Part I: Preliminaries

- Chapter 1: Introduction
- Chapter 2: Process Modeling and Analysis
- Chapter 3: Data Mining

Part II: From Event Logs to Process Models

- Chapter 4: Getting the Data
- Chapter 5: Process Discovery: An Introduction
- Chapter 6: Advanced Process Discovery Techniques

Part III: Beyond Process Discovery

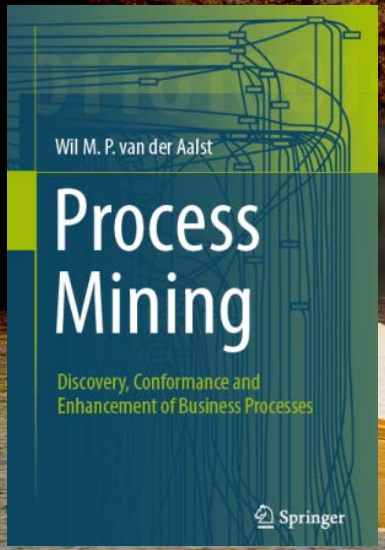
- Chapter 7: Conformance Checking
- Chapter 8: Mining Additional Perspectives
- Chapter 9: Operational Support

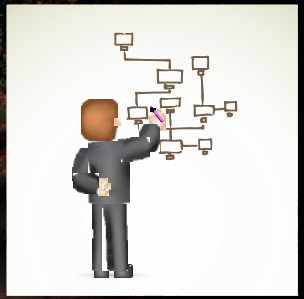
Part IV: Putting Process Mining to Work

- Chapter 10: Tool Support
- Chapter 11: Analyzing "Lasagna Processes"
- Chapter 12: Analyzing "Spaghetti Processes"

Part V: Reflection

- Chapter 13: Cartography and Navigation
- Chapter 14: Epilogue





TU/e