Calculation and Applications of Concurrent Rules

Jonas S. Bezerra, Leila Ribeiro

UFRGS
UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

.inf
INSTITUTO DE INFORMÁTICA
UFRGS

VERITES CERTIFIED

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

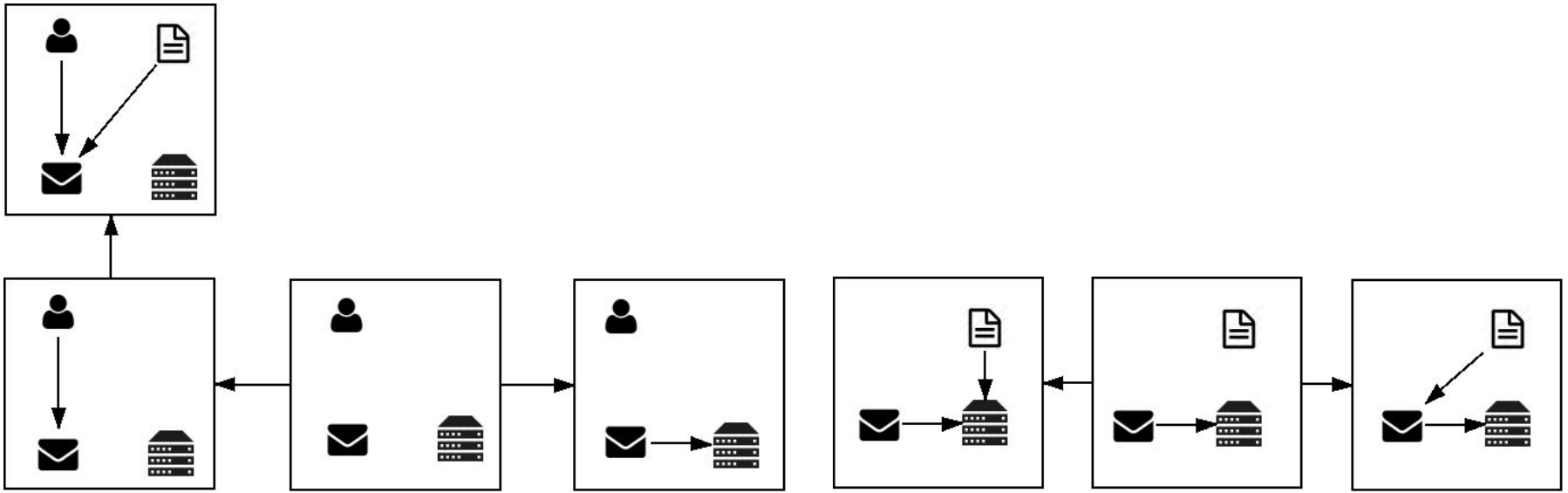
Motivation

- Concurrent Systems (as Graph Transformations)
- Emergent Behaviour

Summarizing Concurrent Behaviour

- Construction of Concurrent Rules
  - Check whether the overall intended behaviour is achieved
  - Show the different behaviours that emerge from the system

Example Grammar

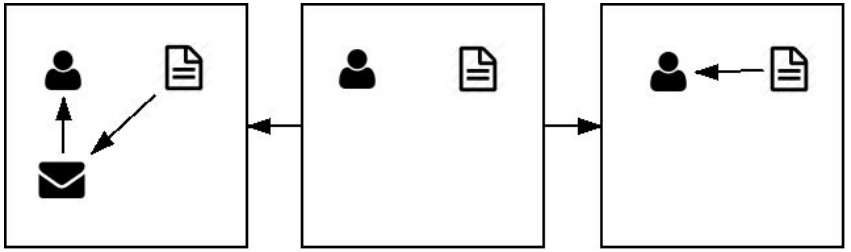


sendMessage

getData

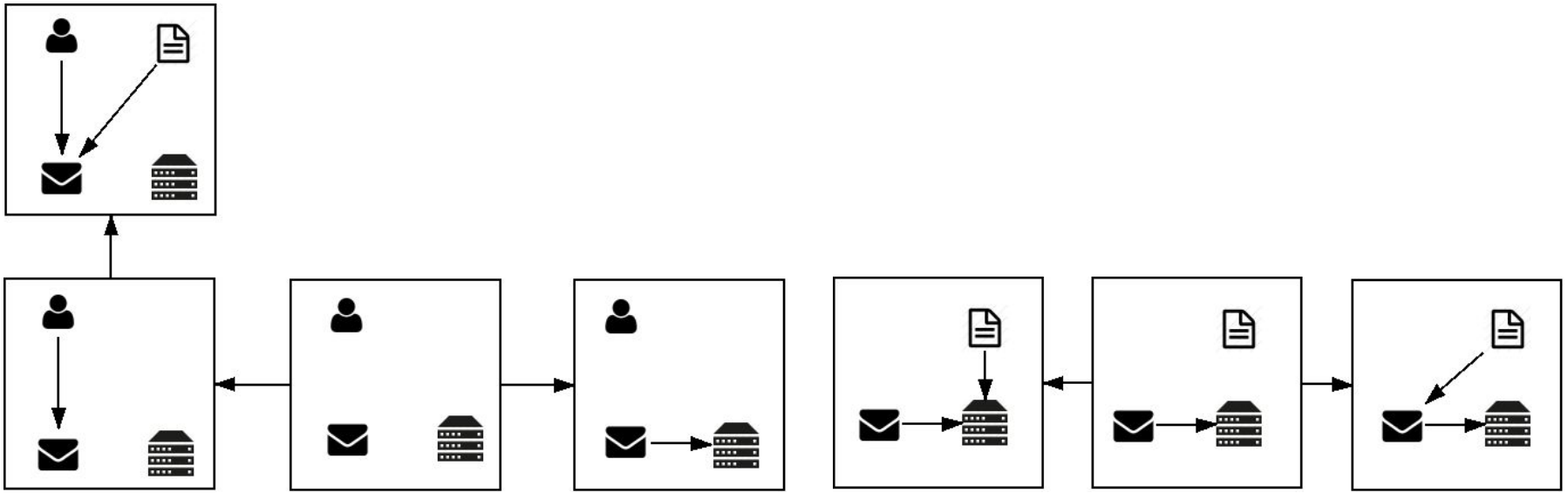
Example Grammar

receiveMessage

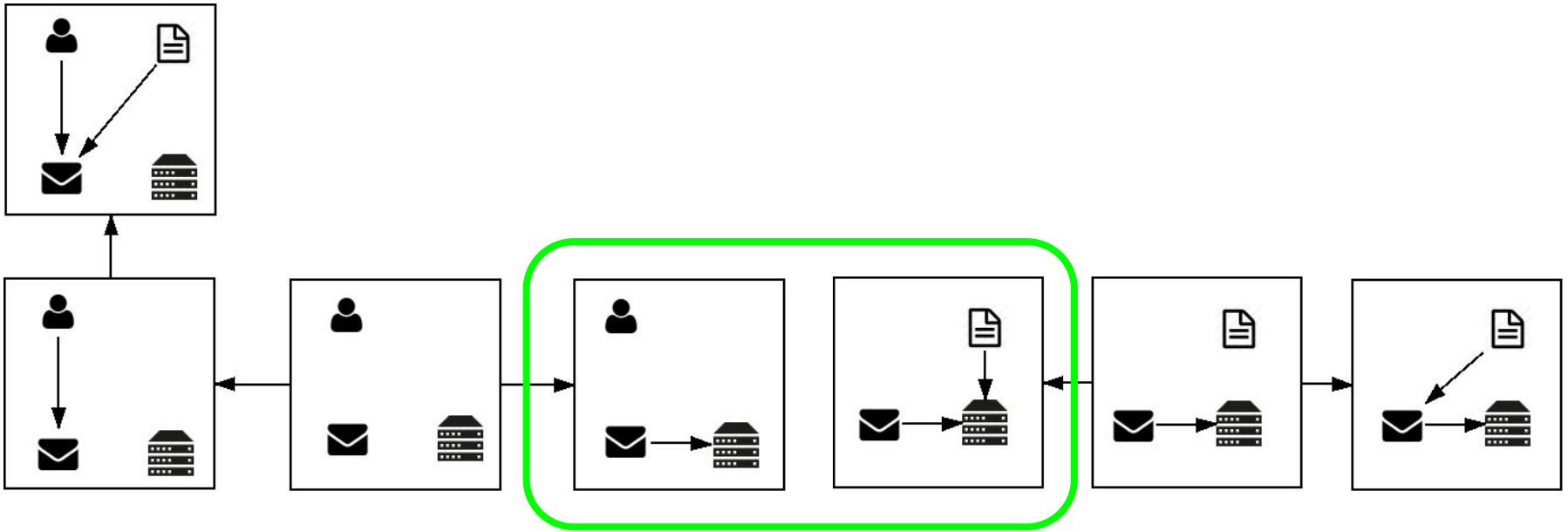


deleteMessage

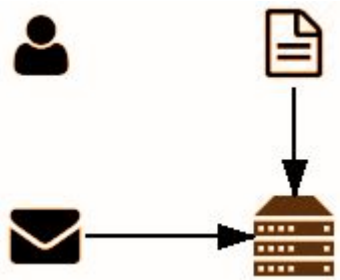
Calculating Concurrent Rules



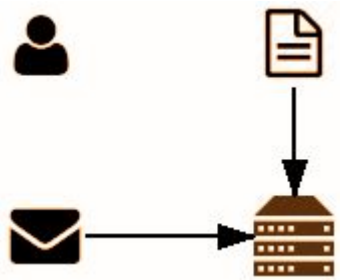
Calculating Concurrent Rules

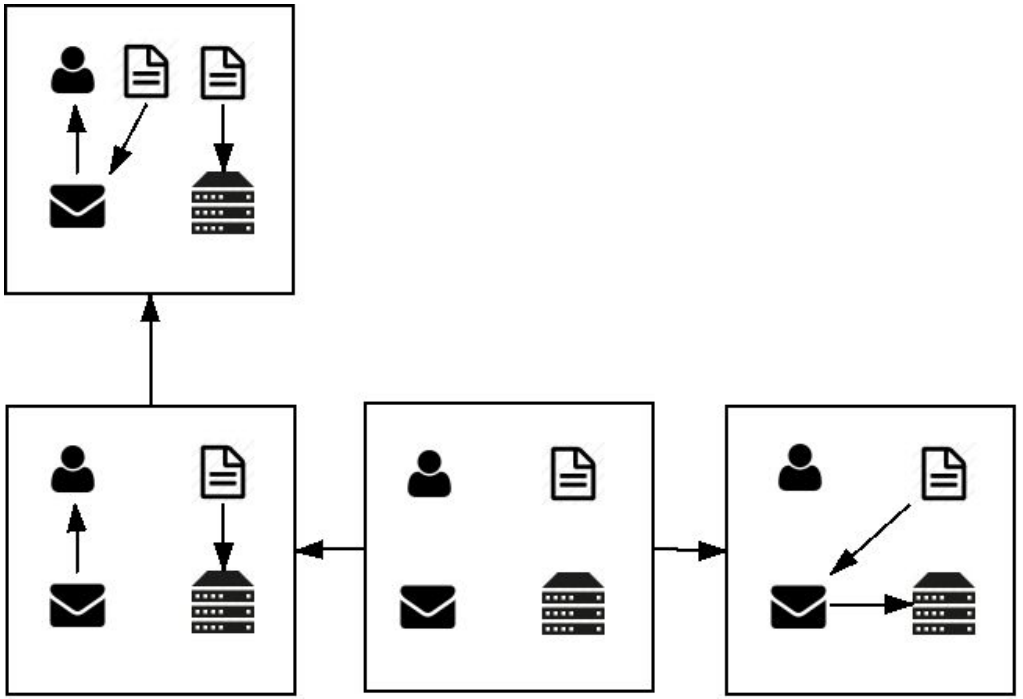


Different Interactions of Rules

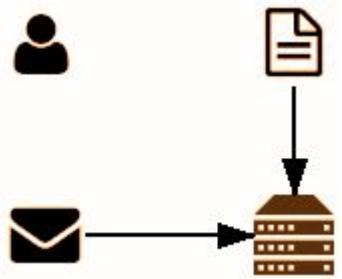


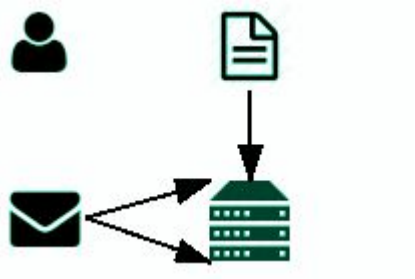
Different Interactions of Rules



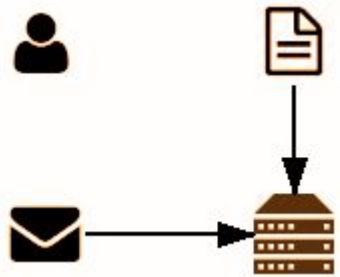


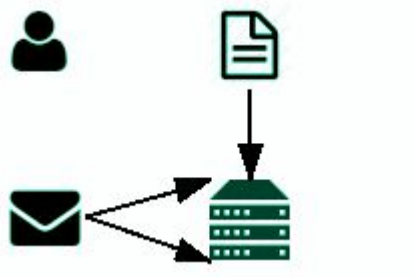
Different Interactions of Rules

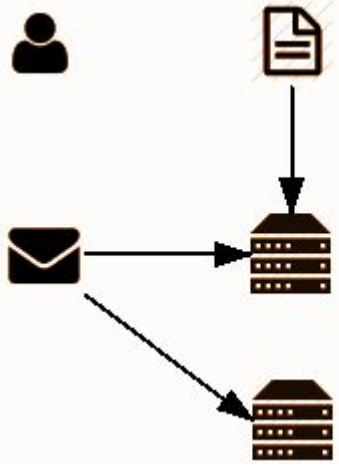




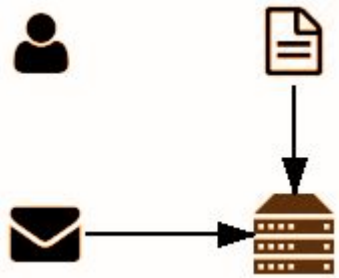
Different Interactions of Rules

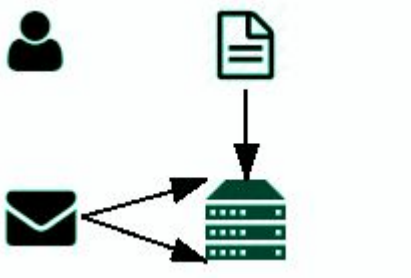


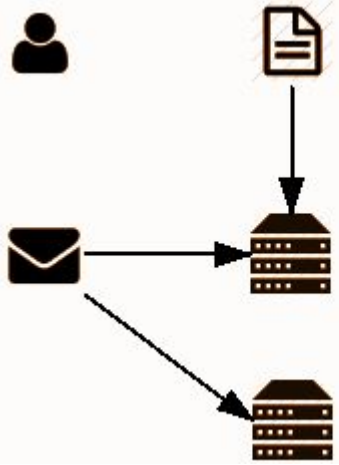


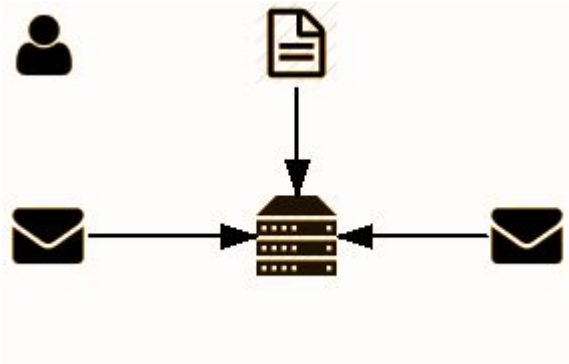


Different Interactions of Rules

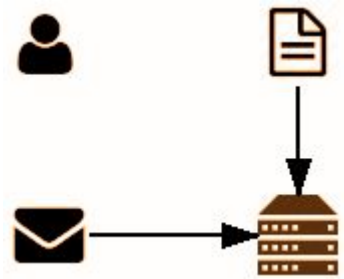


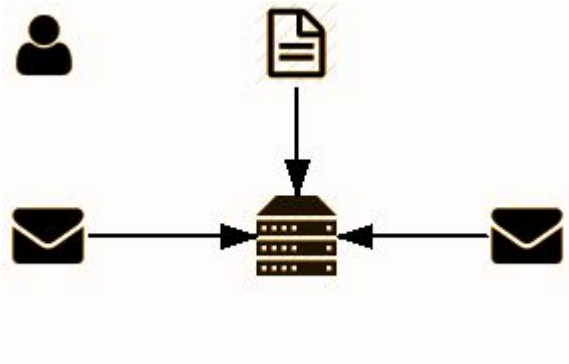


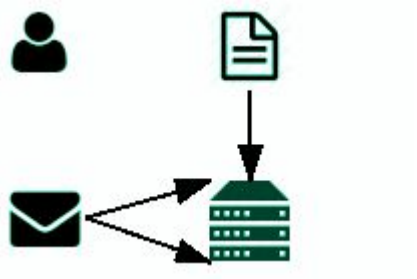


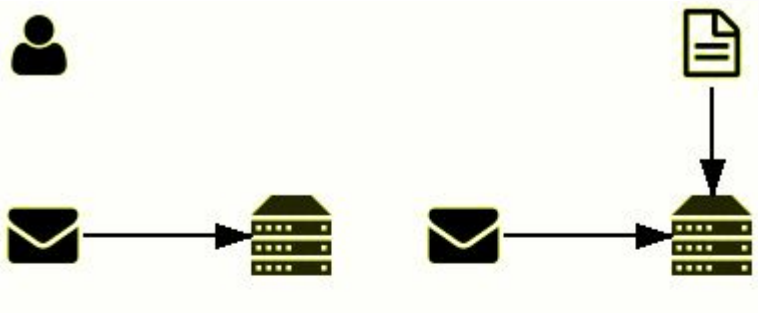


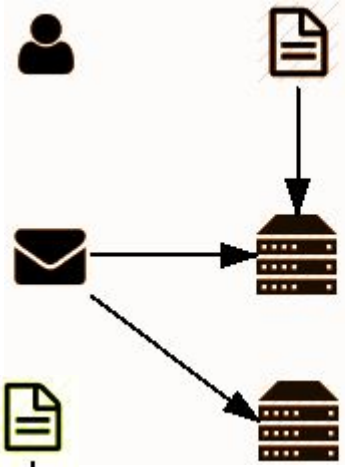
Different Interactions of Rules











Combinatorial Explosion

- There is one concurrent rule for each possible interaction
- We need to calculate a concurrent rule for each generated concurrent rule of a sequence (n-1) and their interactions with n

Combinatorial Explosion

- There is one concurrent rule for each possible interaction
- We need to calculate a concurrent rule for each generated concurrent rule of a sequence (n-1) and their interactions with n

# Rules	2	3	4
# Concurrent Rules	5	189	1021
# NACs	11	533	3961

Combinatorial Explosion

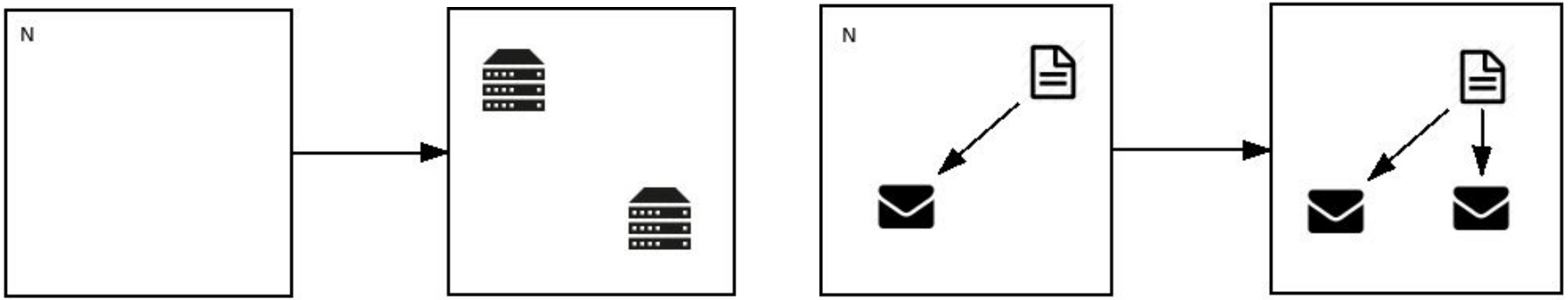
- We can reduce the problem focusing on more “interesting” rules:
  - Constraints
  - Rules by dependencies
  - Maximal Rules

Constraints

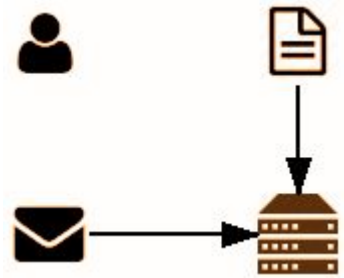
- Restrictions on the model domain
  - Forbid certain structures to exist
  - Ensures that certain structures exist

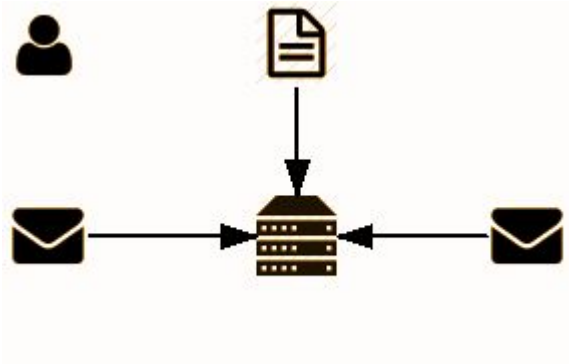
Constraints

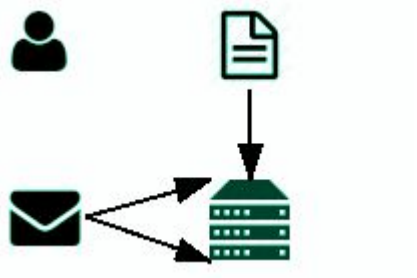
- Restrictions on the model domain
  - Forbid certain structures to exist
  - Ensures that certain structures exist

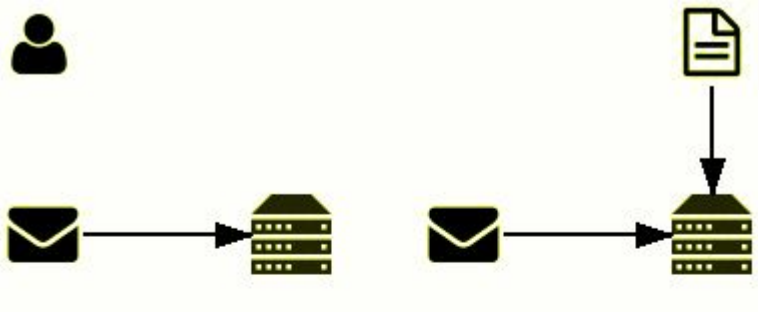


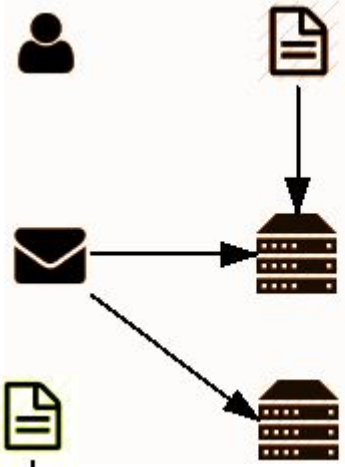
Constraints on Overlappings



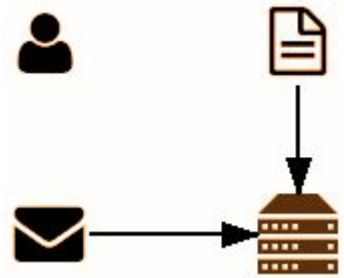


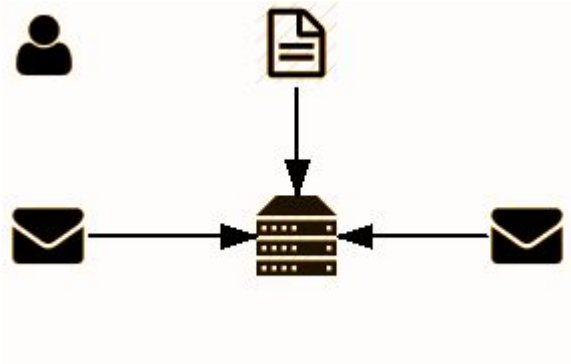


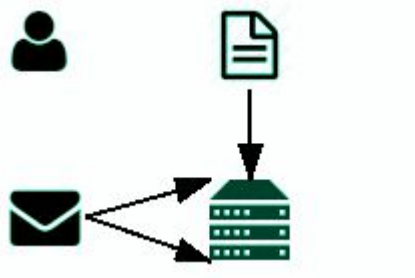


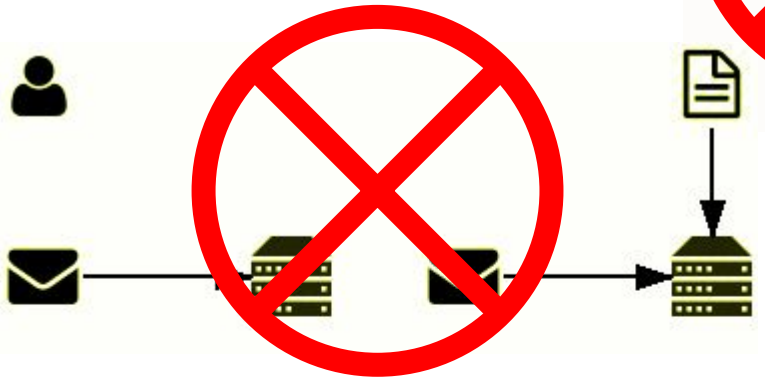


Constraints on Overlappings









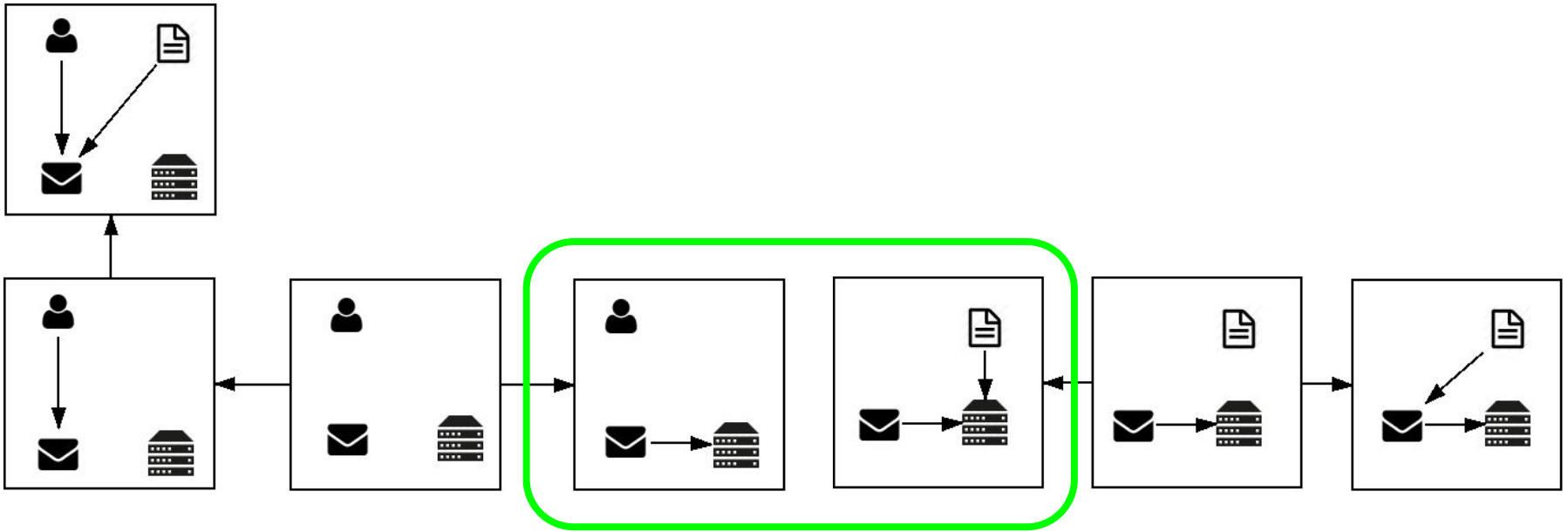
Constraints on Rules and NACs

- Even if the overlapping satisfy the constraints, the LHS and RHS may not
- NACs of the concurrent rules may forbid something already forbidden

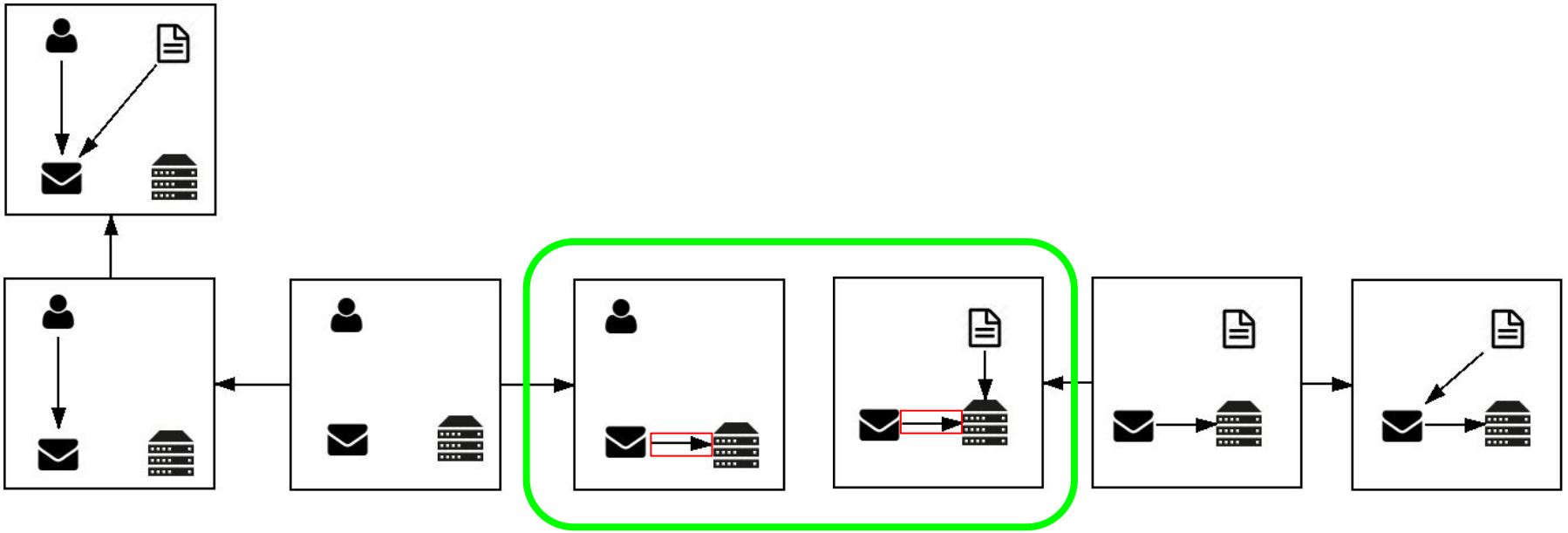
Concurrent-Rules induced by dependency

- Restrict the calculation of rules only where the previous rule:
  - Create something needed for the next one
  - Delete something forbidden by a NAC of the next one

Concurrent-Rules induced by dependency



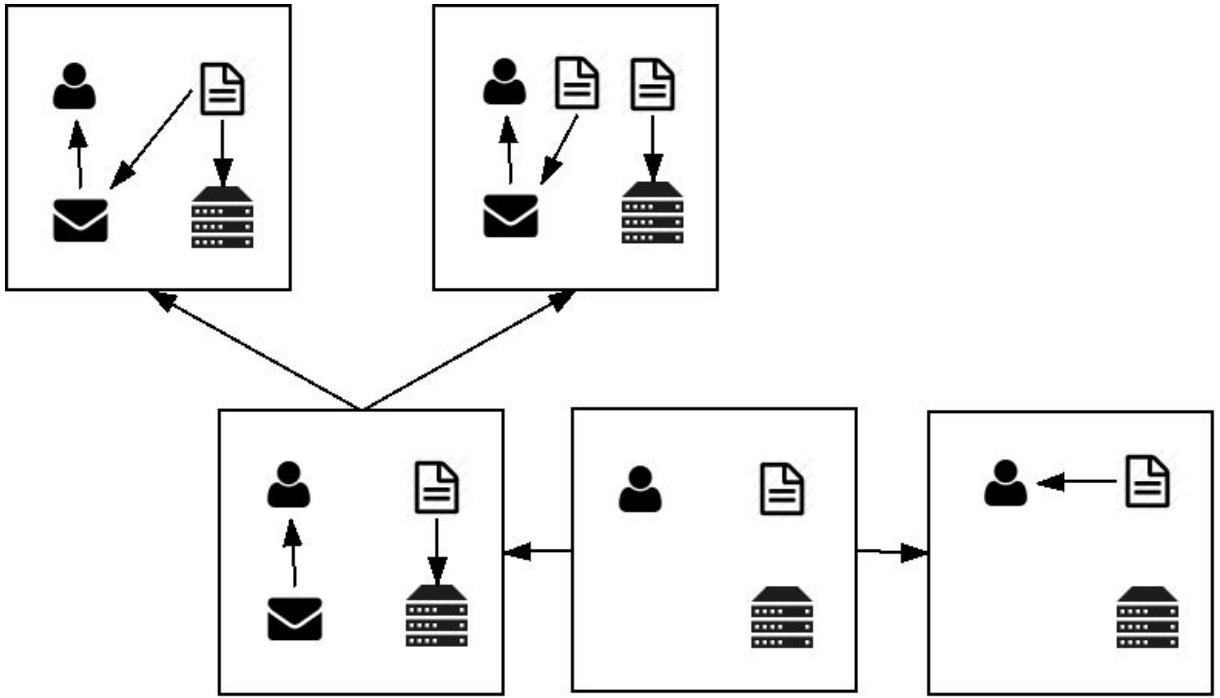
Concurrent-Rules induced by dependency



Maximal Concurrent Rules

- Calculate rules for pairs the represent the maximal possible interactions

Maximal Concurrent Rules



Verigraph

- All strategies above were implemented on verigraph
- These strategies can be combined
- The combination of strategies benefits from Haskell laziness

Results

- Concurrent rules for the 4 rules of the example grammar

Strategy	Time	Number of Rules	Number of NACs
None	1m30s	1021 rules	$\approx$ 4 nacs/rule
Constraints on Overlappings	8s	141 rules	$\approx$ 2 nacs/rule
Constraints on LHS and RHS	8s	141 rules	$\approx$ 2 nacs/rule
Dependencies	0.45s	2 rules	2 nacs/rule
Maximal Rule	0.32s	1 rule	2 nacs/rule

Table 1. Concurrent rules for the server grammar

Results

- Concurrent rules for an elevator system grammar with 8 rules

Strategy	Time	Number of Rules	Number of NACs
None	>2d	NA	NA
Constraints on Overlappings	30m	11313 rules	$\approx$ 0.7 nacs/rule
Constraints on LHS and RHS	1m55s	287 rules	$\approx$ 5 nacs/rule
Dependencies	2m15s	9 rules	$\approx$ 4 nacs/rule
Maximal Rule	0.93s	0 rules	0 nacs/rule

Table 2. Concurrent rules for elevator grammar

Conclusion

- Performance improvement
- Focus on meaningful rules
- Available on verigraph: https://github.com/Verites/verigraph

Ongoing/Future Work

- Use of concurrent rules for Use Cases analysis
- Use of concurrent rules for generating test cases
- Use of other static analysis for generating test cases

Acknowledgement

Contact

- Jonas S. Bezerra: jsbezerra@inf.ufrgs.br
- Profª. Drª. Leila Ribeiro: leila@inf.ufrgs.br

Calculation and Applications of Concurrent Rules