From Constraints to Application Conditions

Presentation

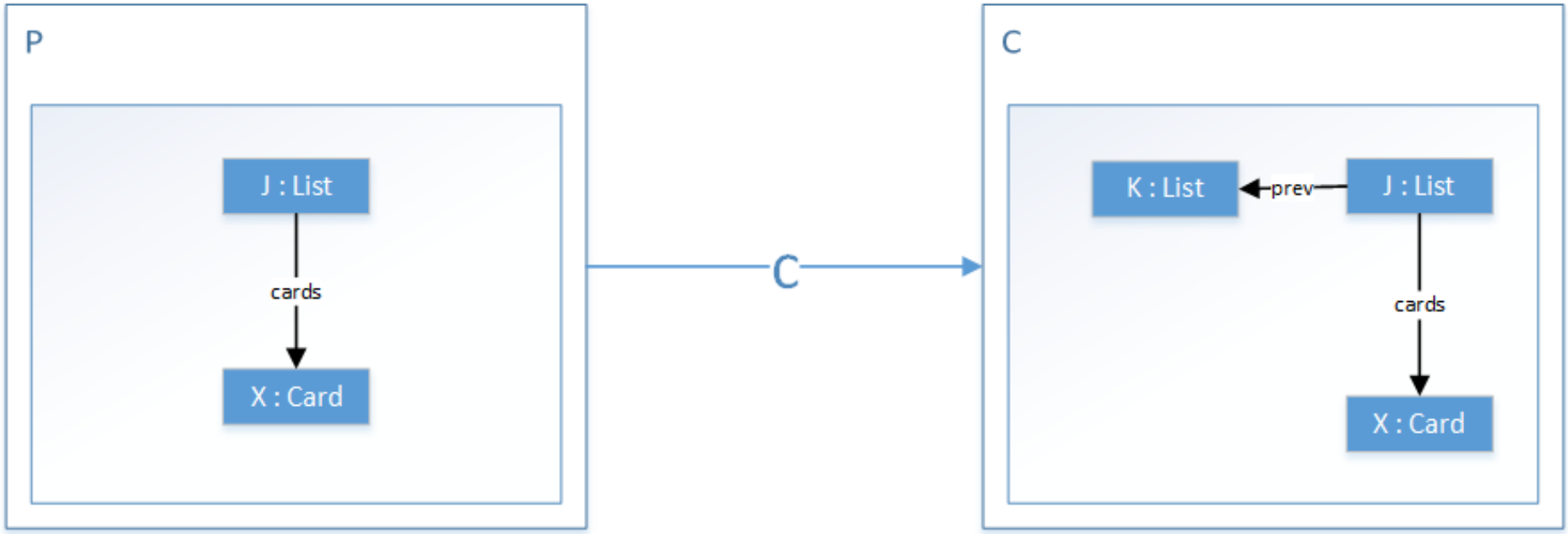
Robin Oppermann and Patrick Robrecht

06/02/2017

Fundamentals of Model-Driven Engineering
Jun.-Prof. Anthony Anjorin
Research Group Database and Information Systems
Department of Computer Science
Paderborn University

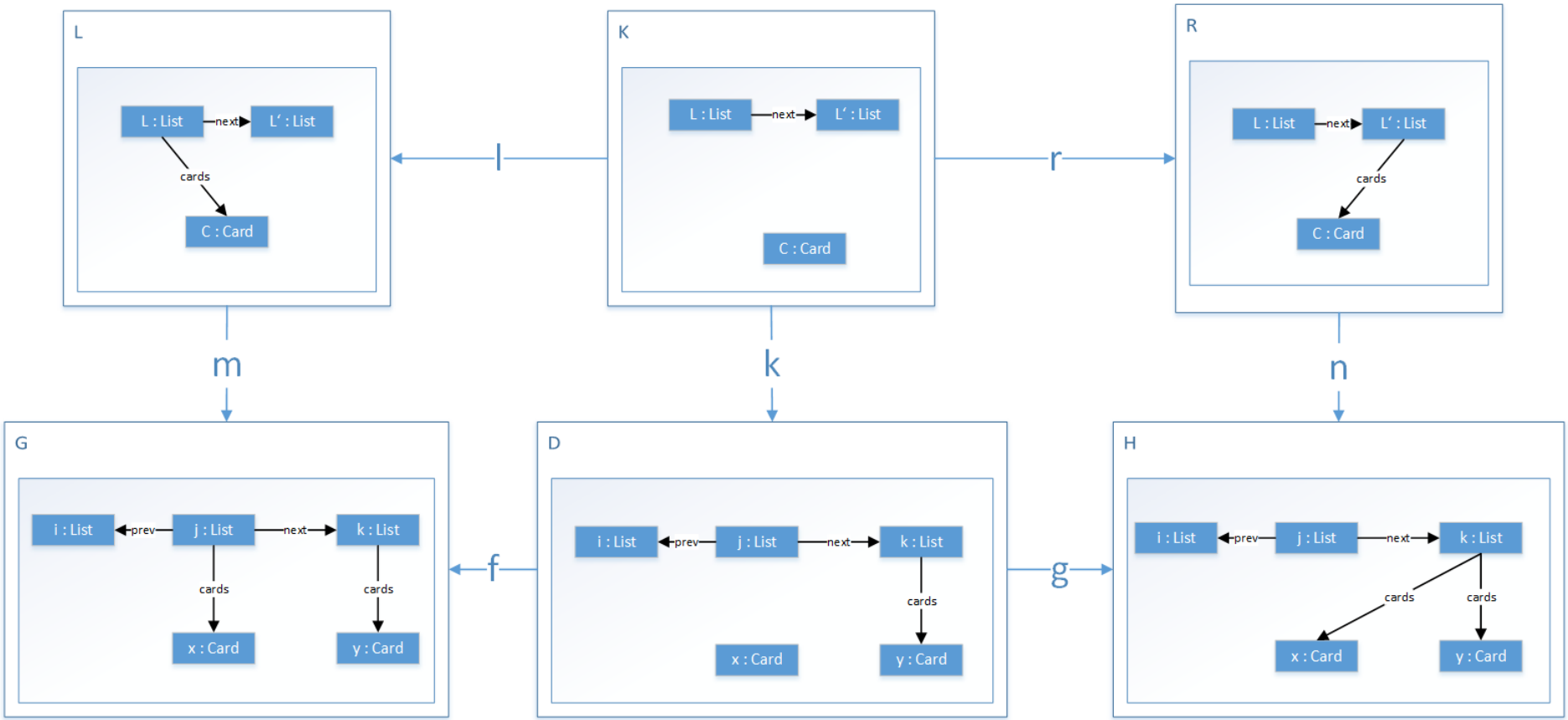
Introduction

Constraint Example: Each List J with a card X has a previous list K



Introduction

Rule Example: Moving Card C from List L to L'



Introduction

Why do we want to construct application conditions from constraints?

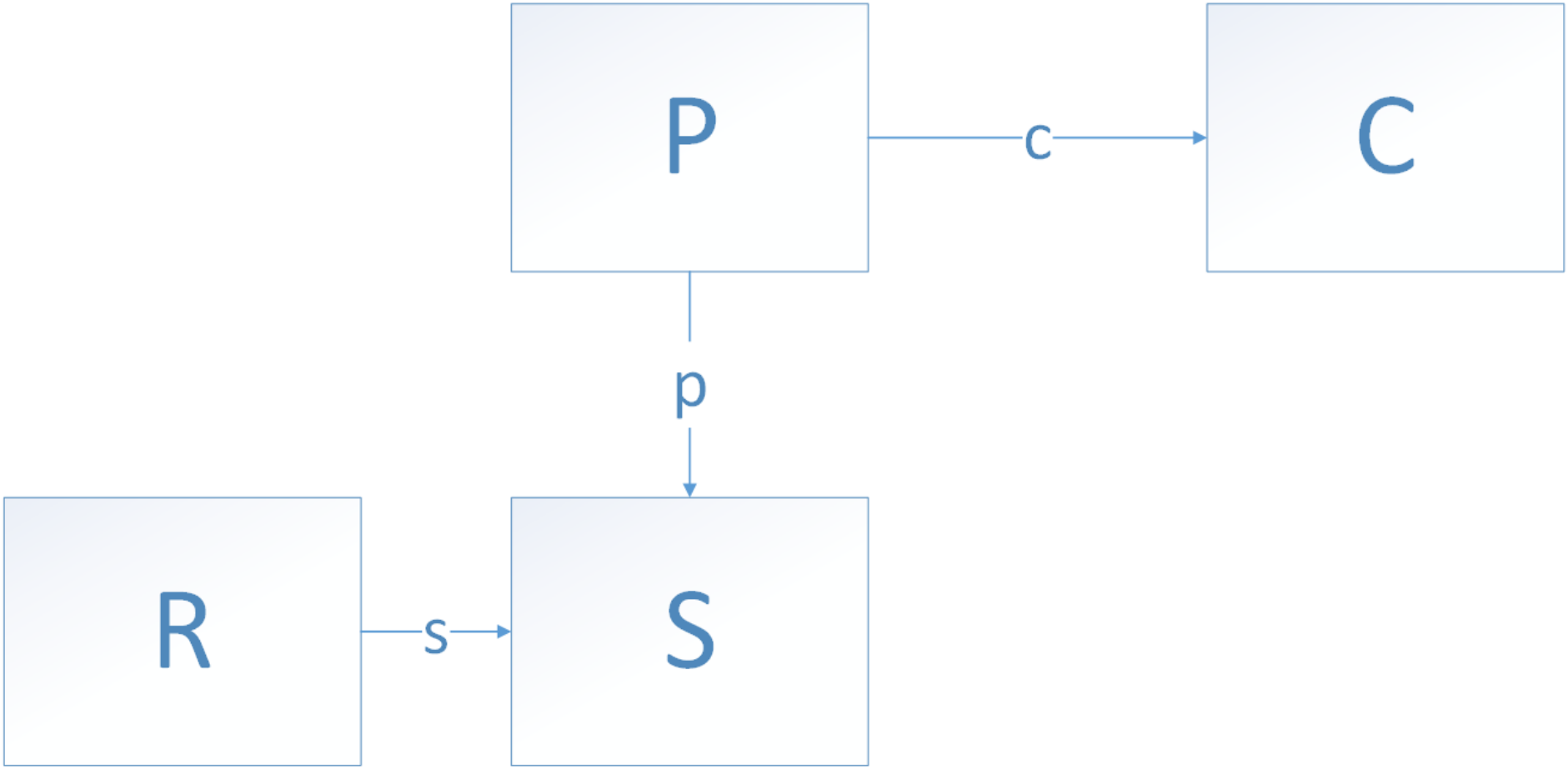
- model transformation system containing sets of rules and constraints
- need to ensure: graph after rule application does not violate a constraint
- idea: construct application conditions to check this before rule application
- regeneration after changes in rules / constraints necessary
  $\Rightarrow$ construction needs to be automatized

Contents

Contents

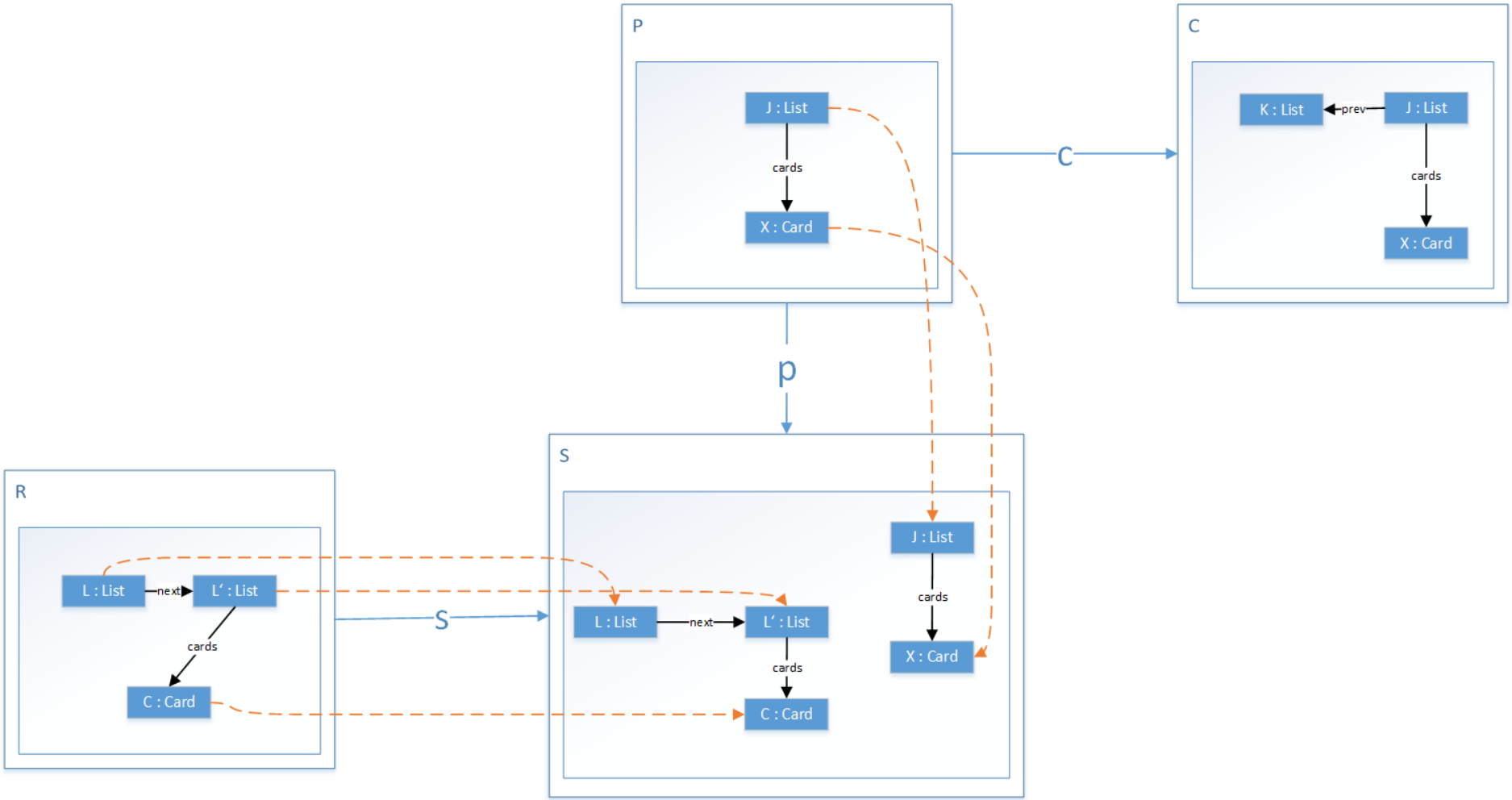
Construction of Application Conditions from Constraints

Construct possible epimorphic gluings S – Schema



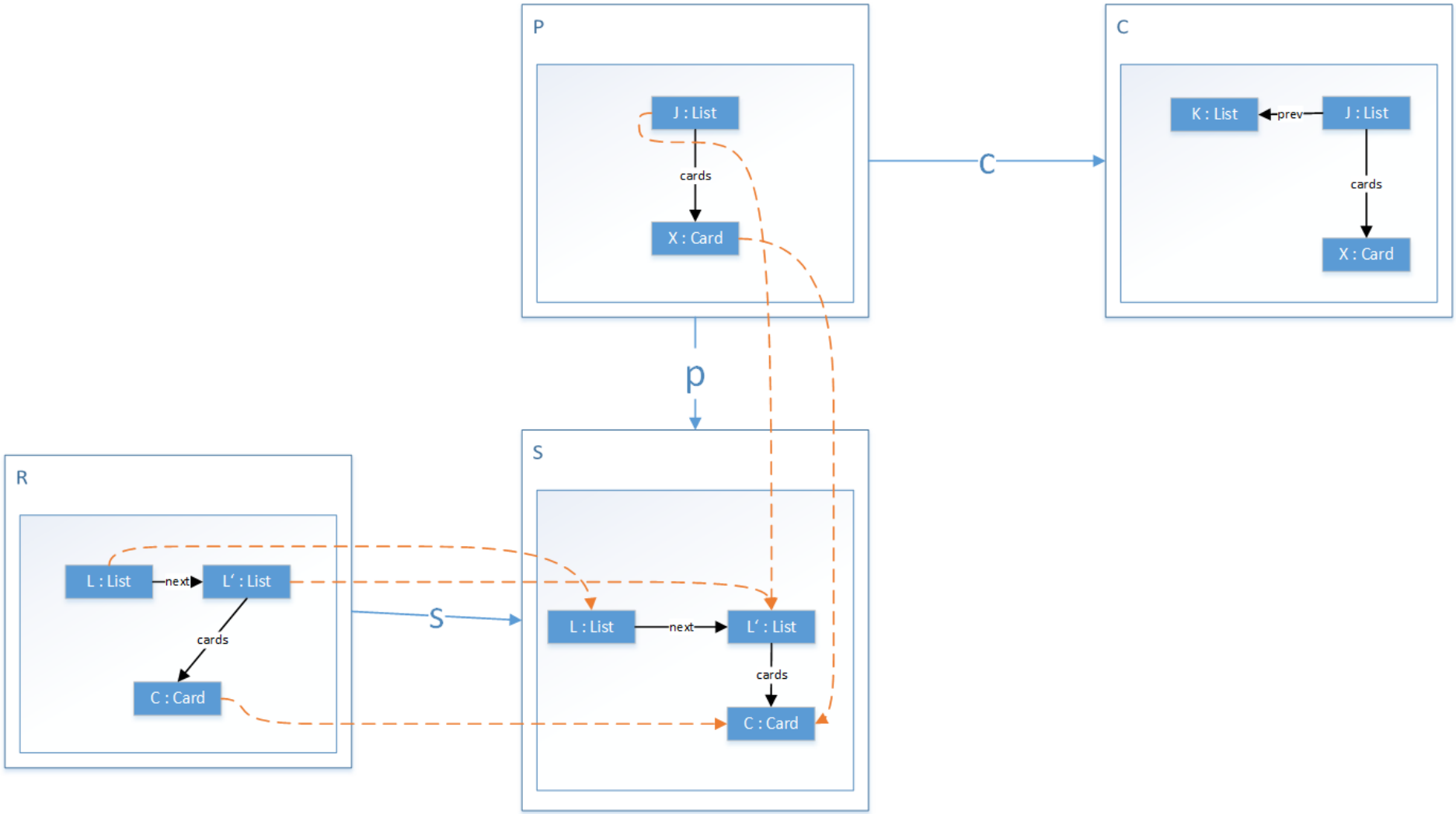
Construction of Application Conditions from Constraints

Construct possible epimorphic gluings S – Example Step 1



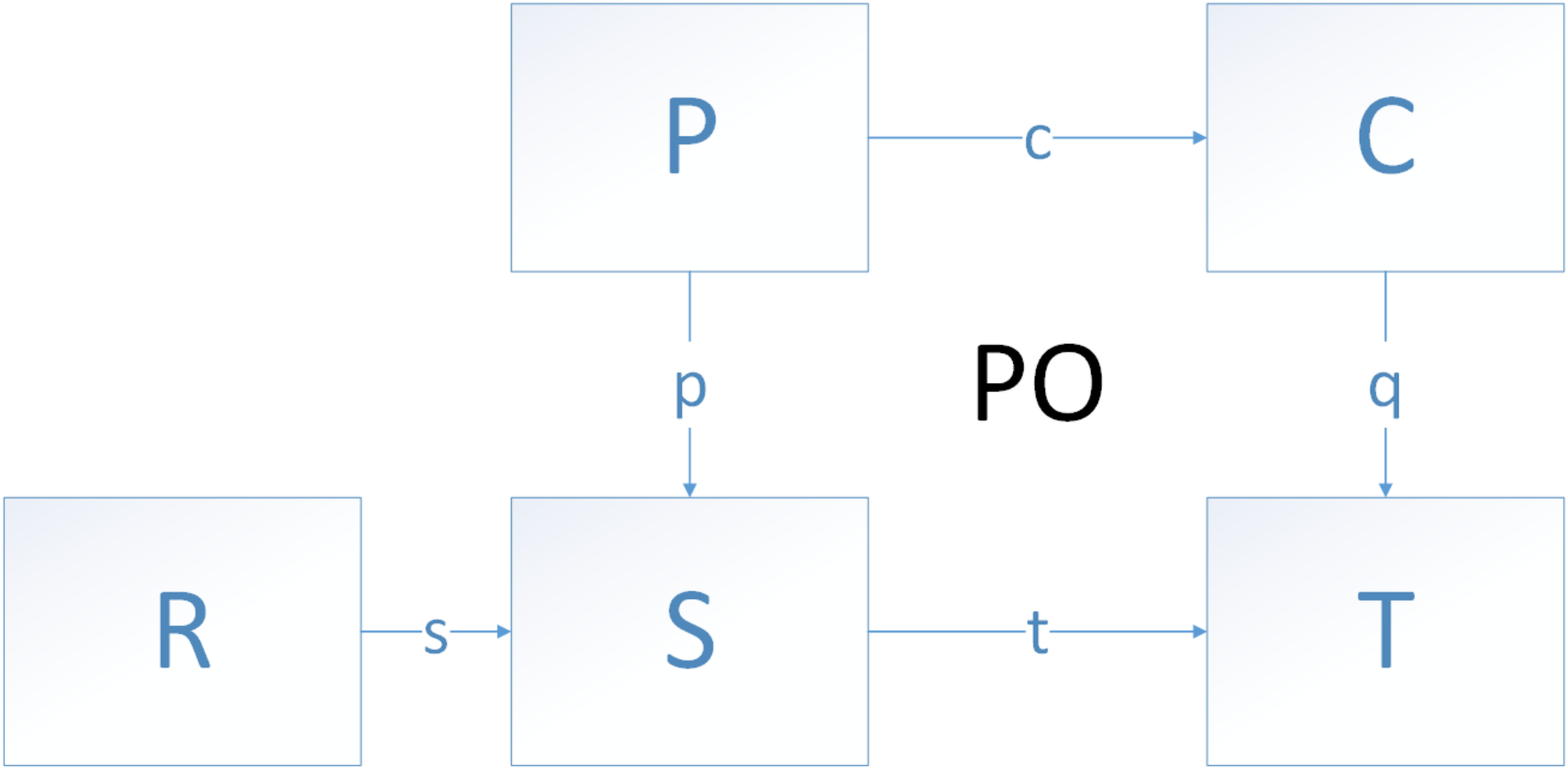
Construction of Application Conditions from Constraints

Construct possible epimorphic gluings S – Example Step 2



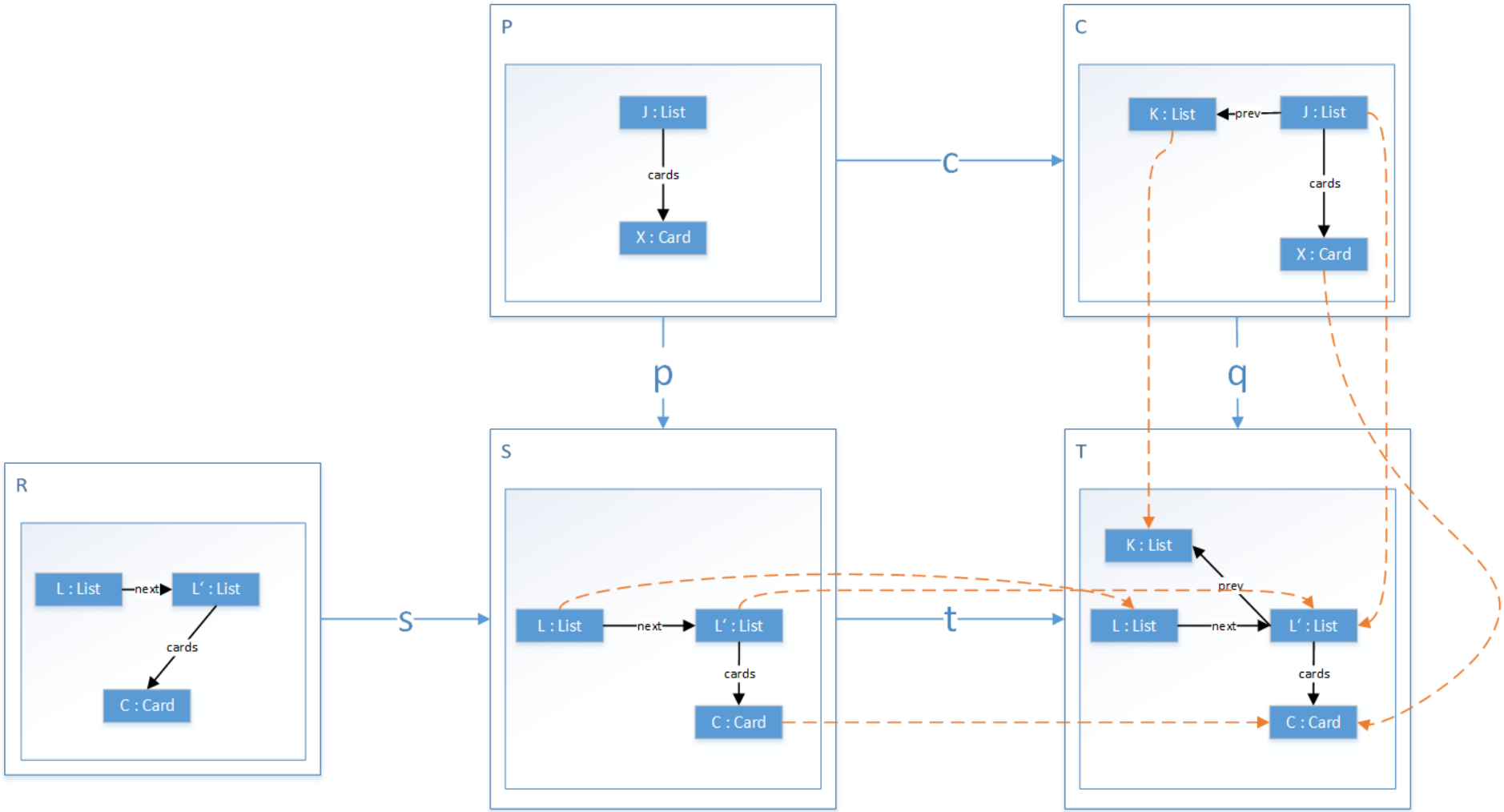
Construction of Application Conditions from Constraints

Construct pushout T – Schema



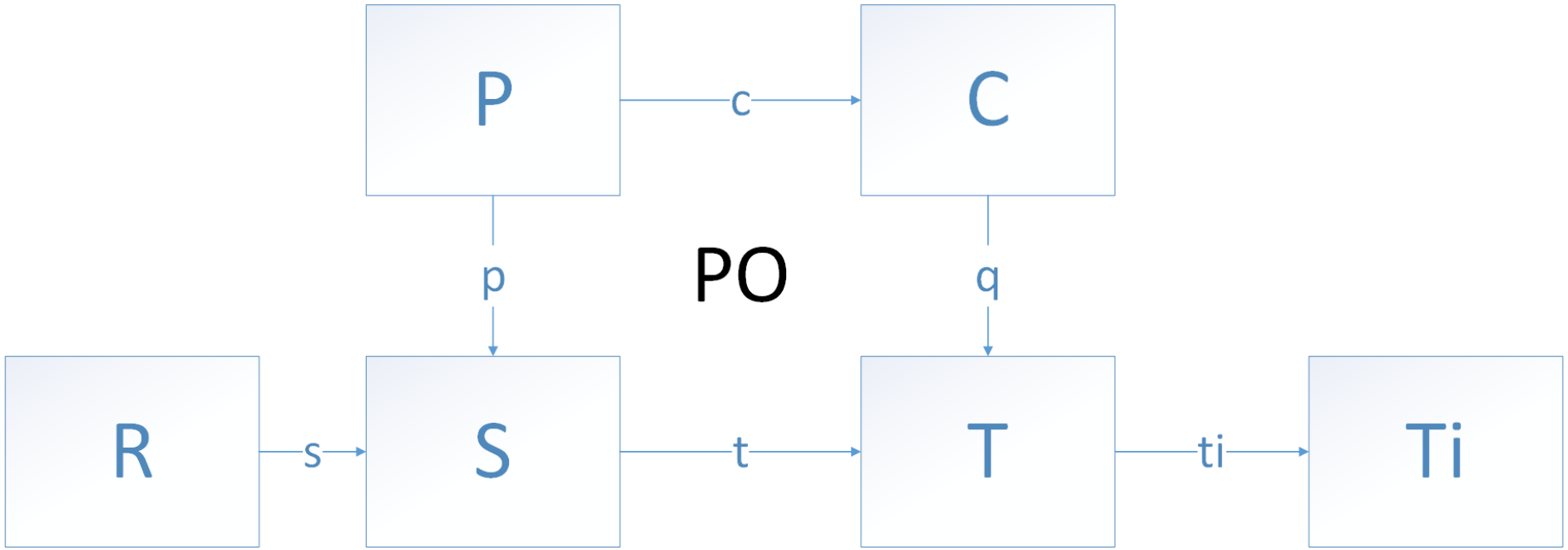
Construction of Application Conditions from Constraints

Construct pushout T – Example



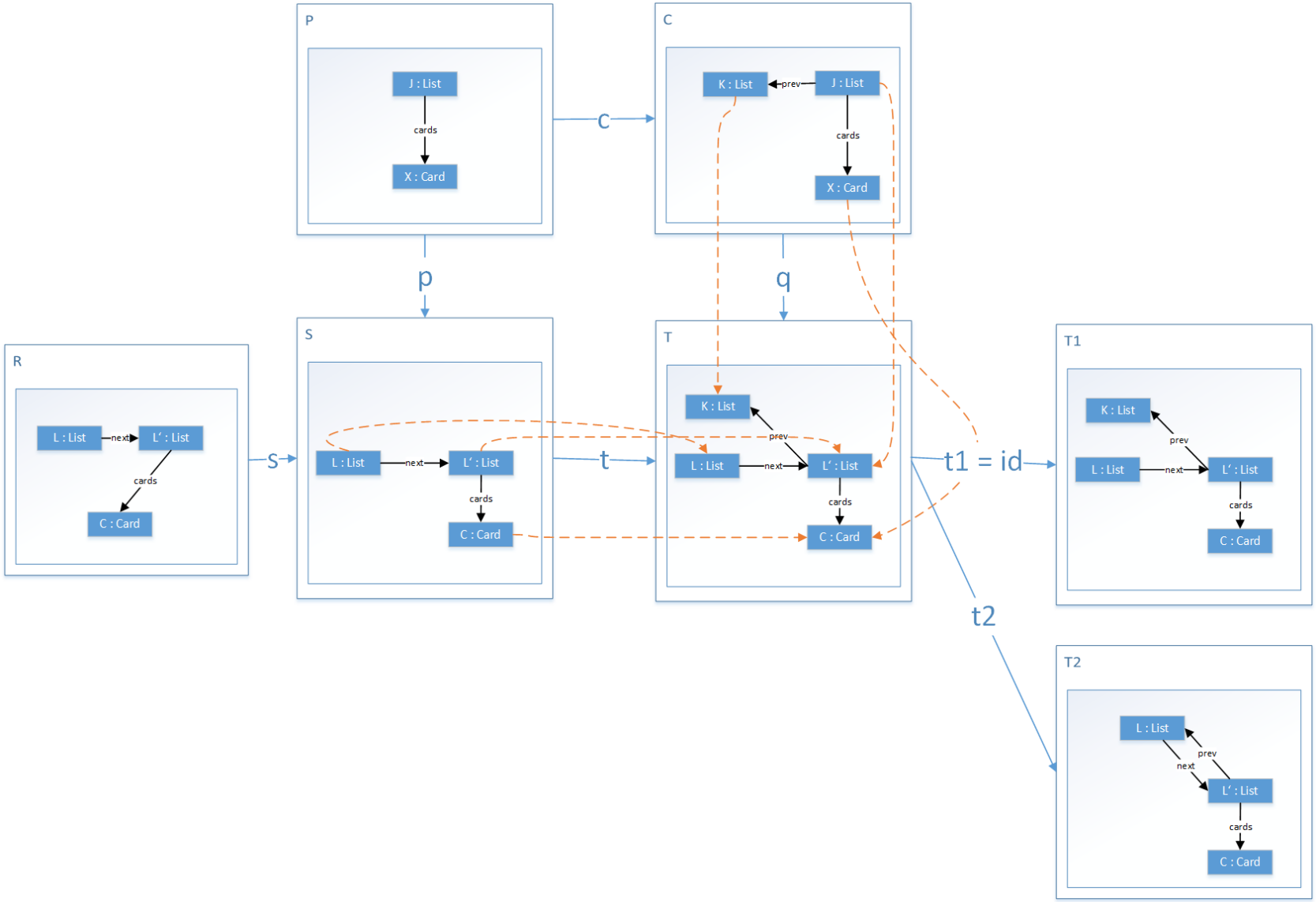
Construction of Application Conditions from Constraints

Construct epimorphic gluings T_i – Schema



Construction of Application Conditions from Constraints

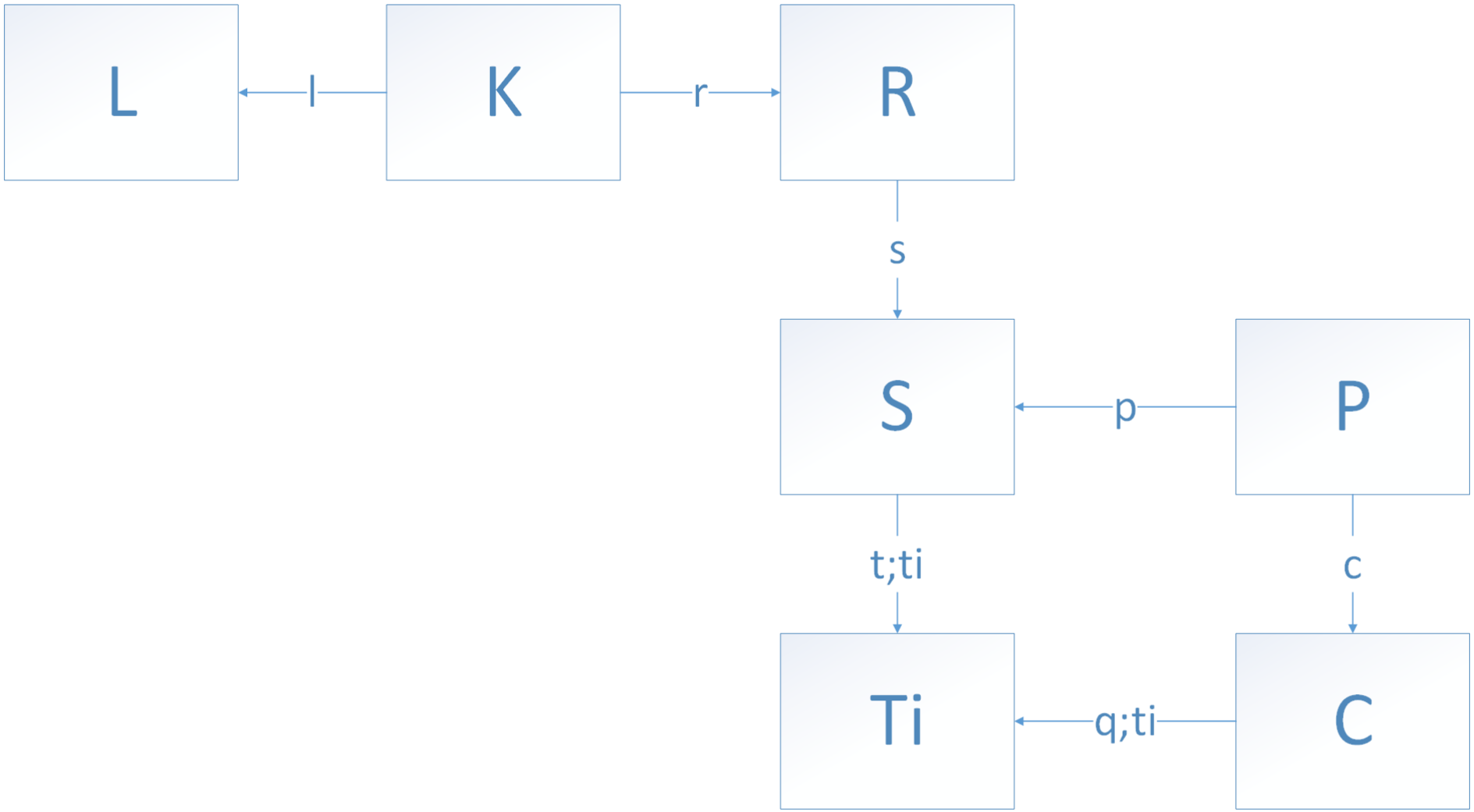
Construct epimorphic gluings T_i – Example



Contents

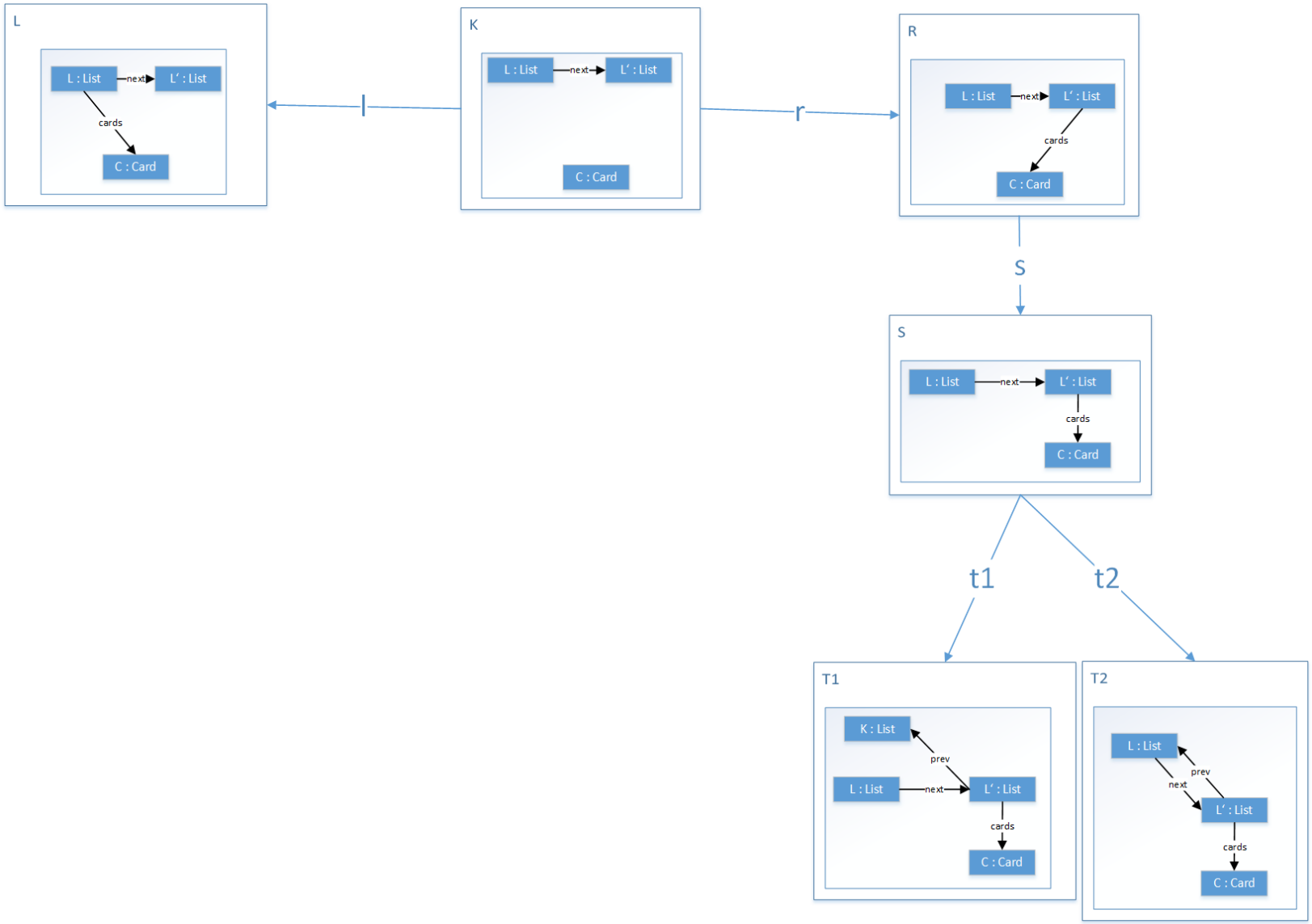
Construction of Left from Right Application Conditions

What we have done so far: Right Application Conditions



Construction of Left from Right Application Conditions

Right Application Condition – Example



Construction of Left from Right Application Conditions

Left and Right Application Conditions

Right application condition

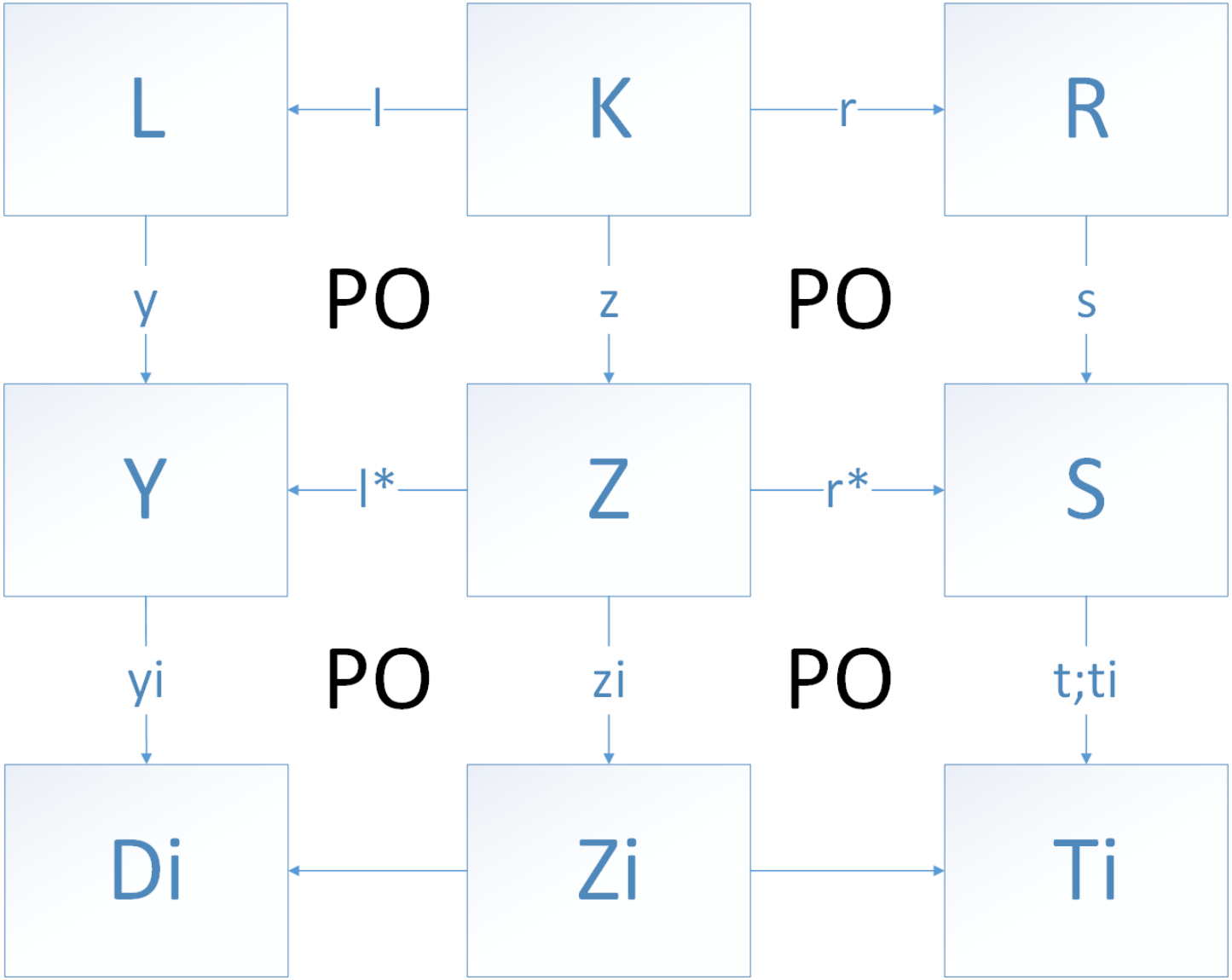
Rule is applicable if the right application condition holds in H (i. e. after rule application).

Left application condition

Rule is applicable if the left application condition holds in G (i. e. before rule application), so the application of the rule doesn't result in a graph violating one of the constraints.

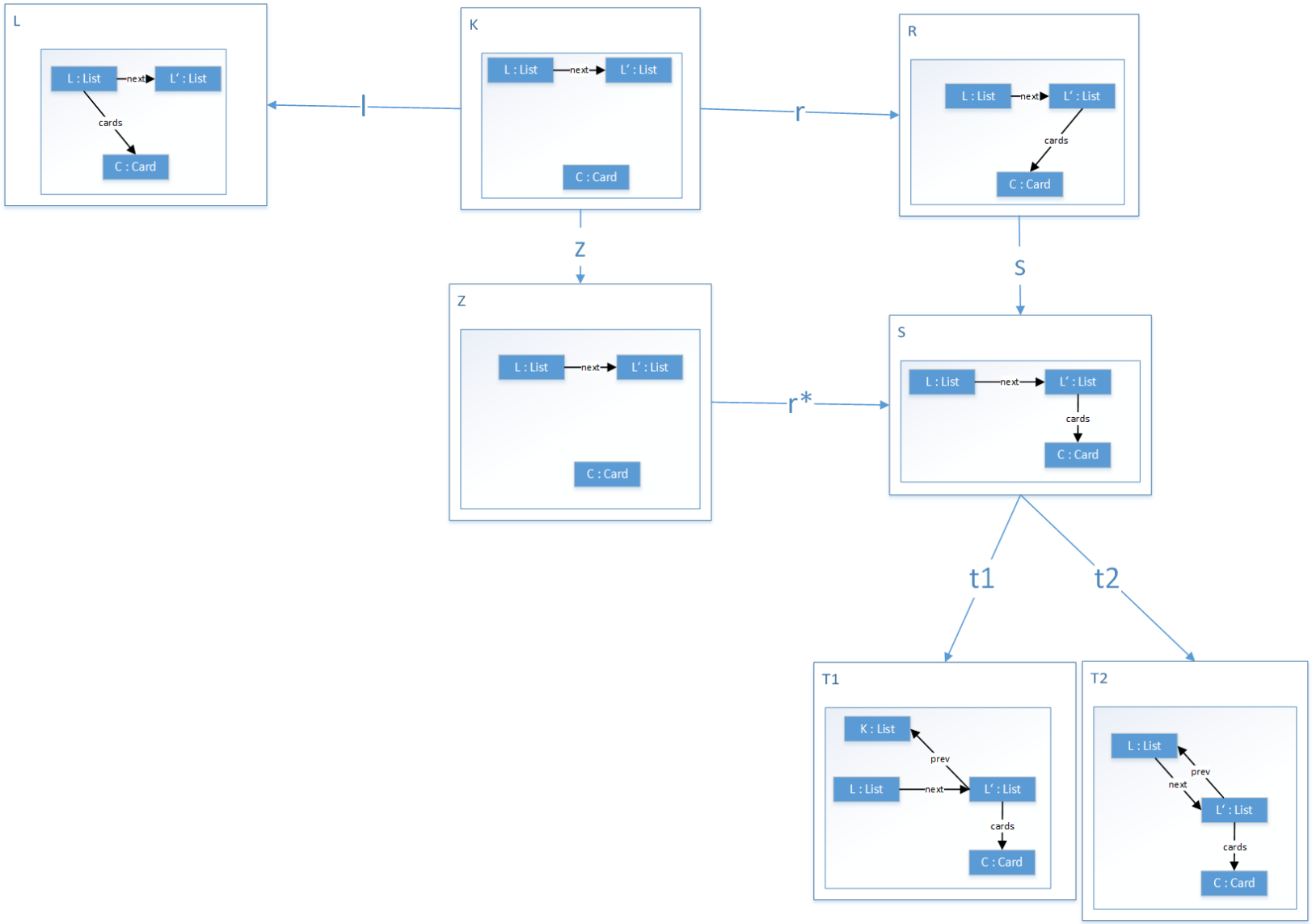
Construction of Left from Right Application Conditions

From Right to Left Application Conditions – Schema



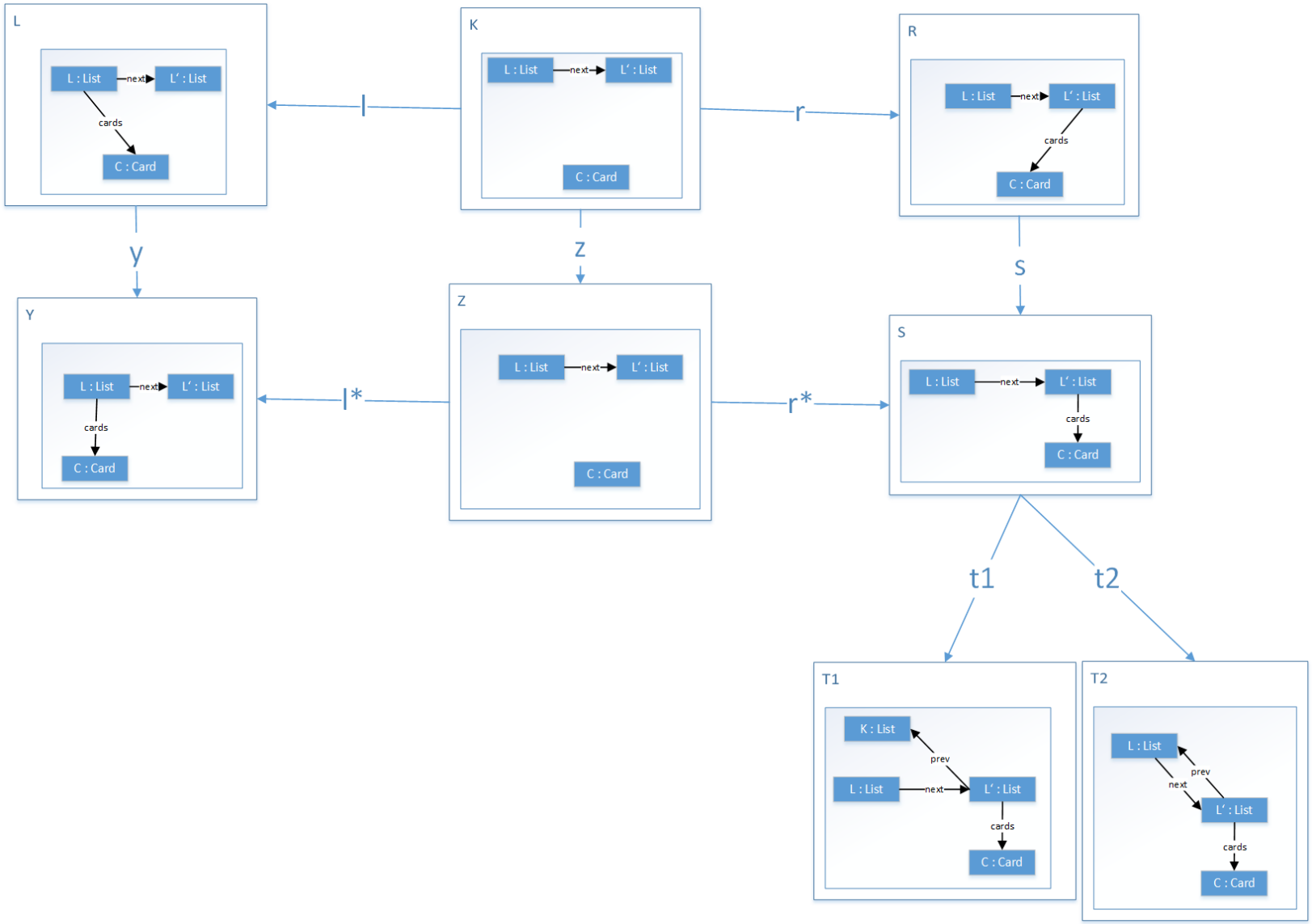
Construction of Left from Right Application Conditions

Construct pushout complement Z – Example



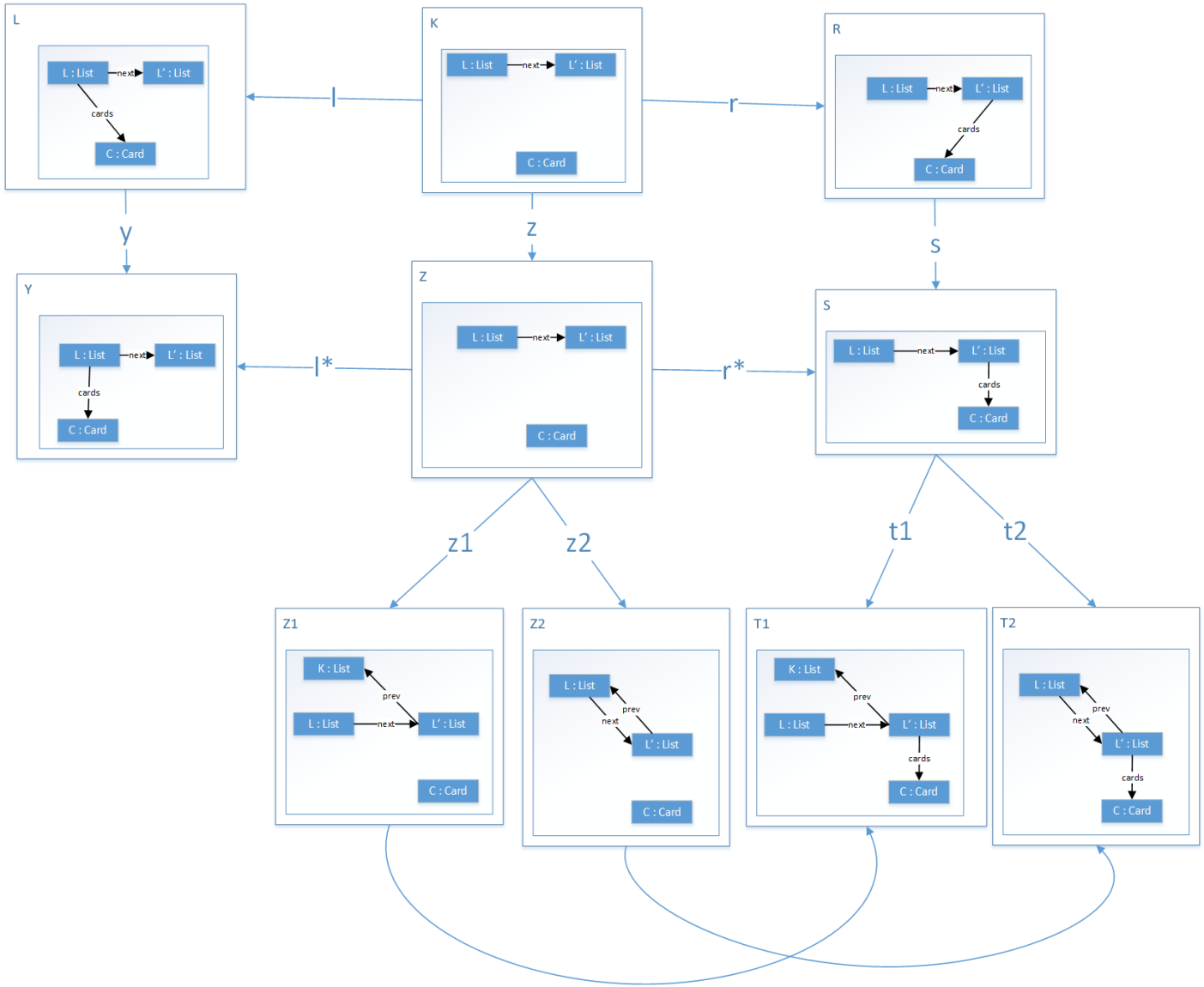
Construction of Left from Right Application Conditions

Construct pushout Y – Example



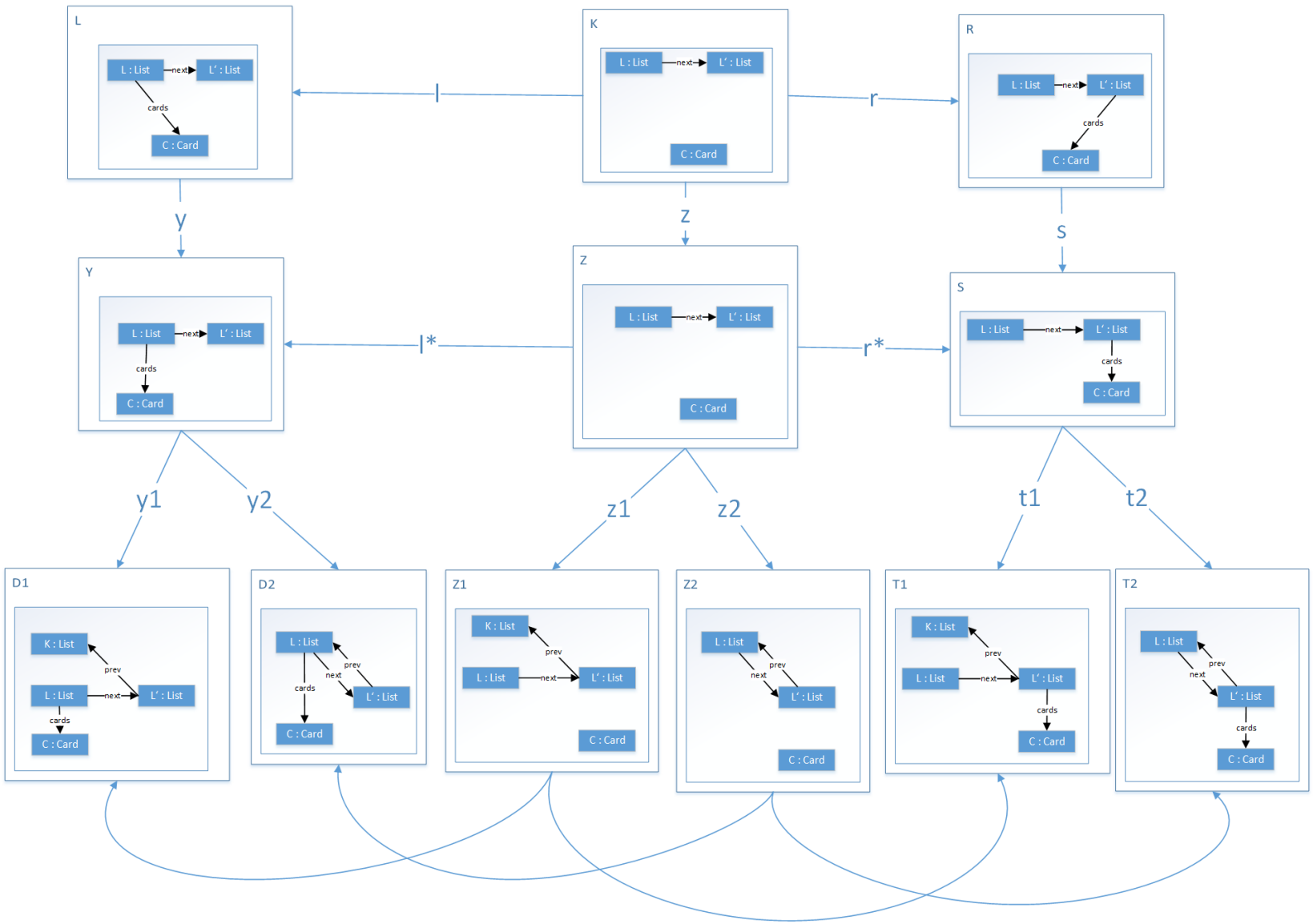
Construction of Left from Right Application Conditions

Construct pushout complements Z_i – Example



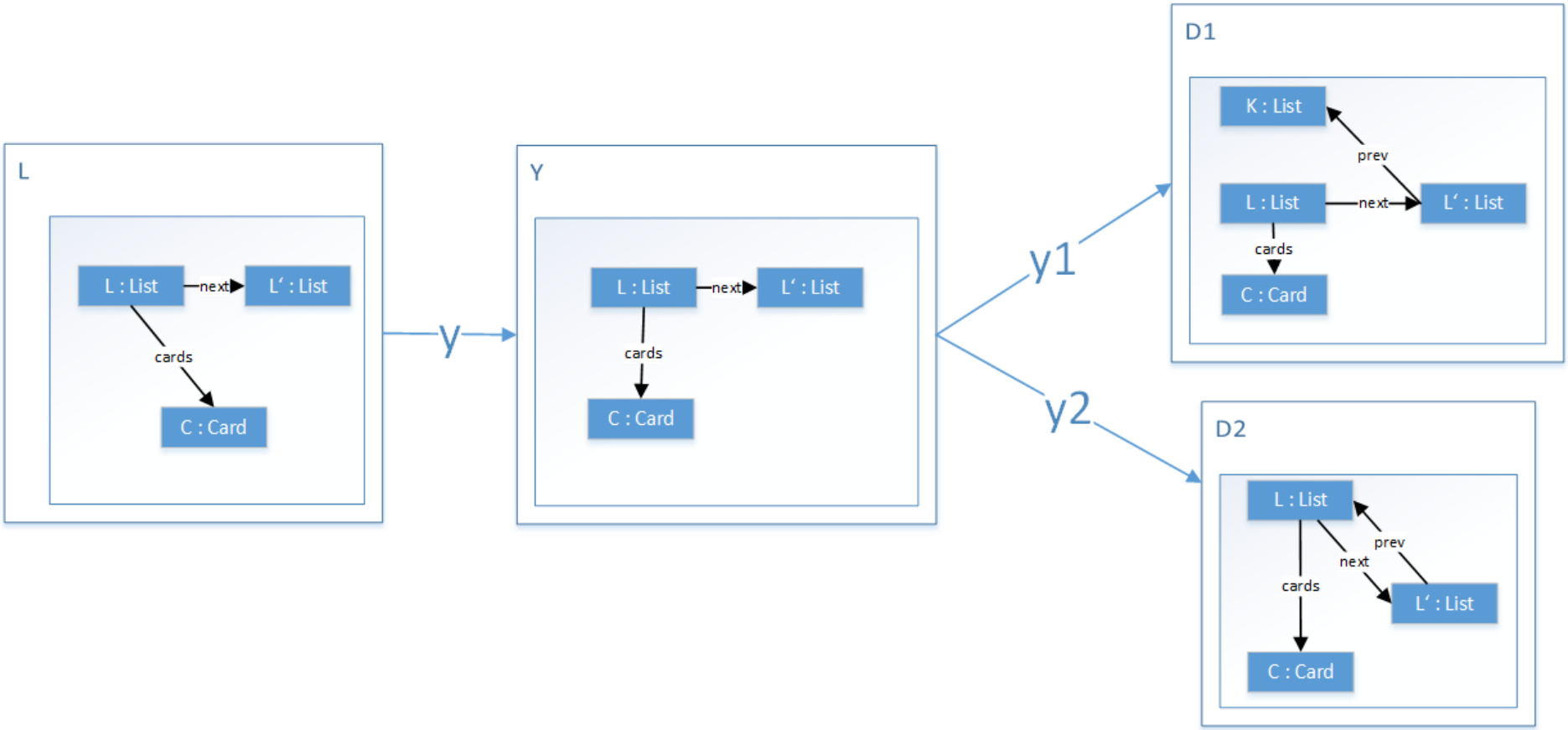
Construction of Left from Right Application Conditions

Construct pushout D_i – Example



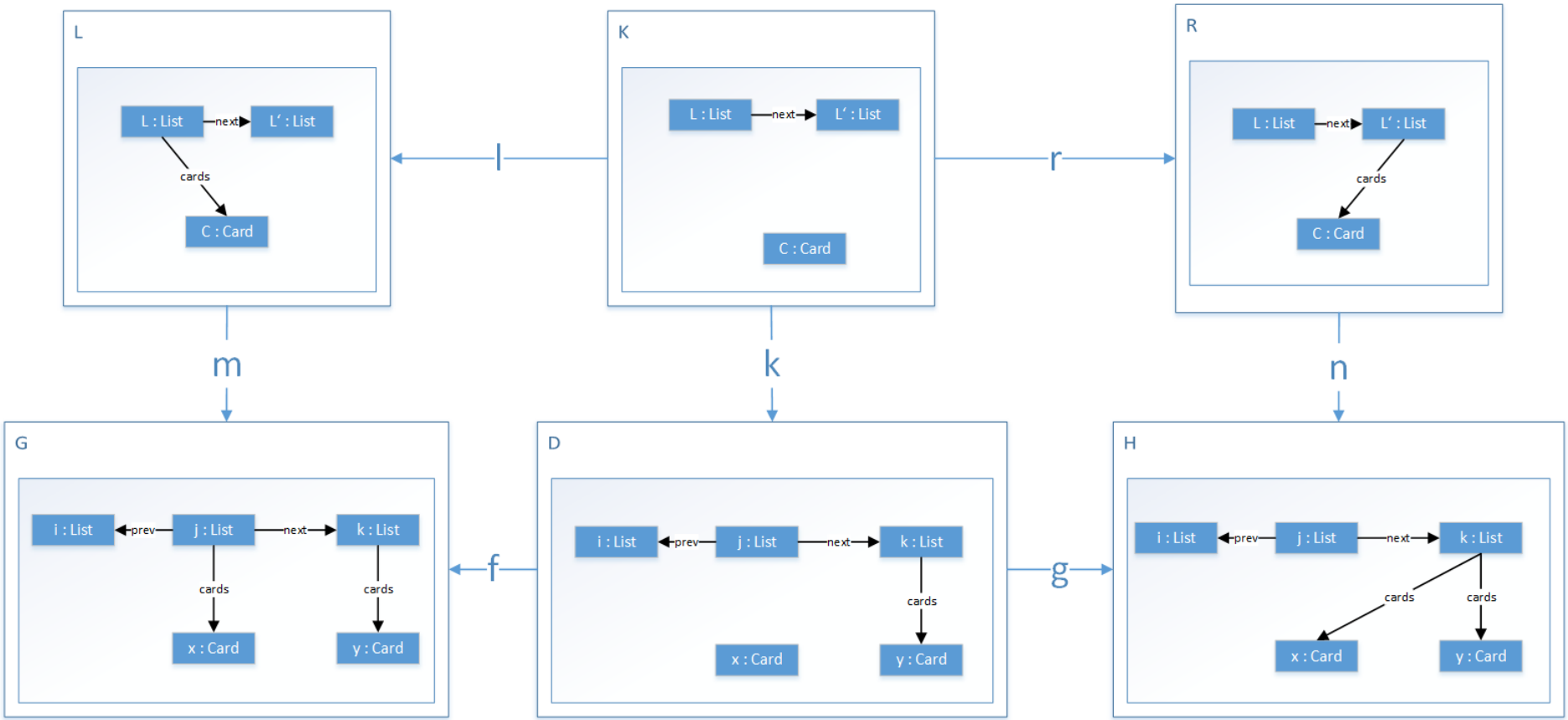
Construction of Left from Right Application Conditions

Left Application Condition – Example



Rule Application allowed?

Rule Example: Moving Card C from List L to L'



Contents

Conclusion

Our implementation

- interesting topic, worth repeating with focus on current performance limitations on larger examples
- The construction of application conditions can be implemented with the code from the exercises
  - currently only implemented for constraint $c : P \to C$ (not multiple conclusions)

Problems during implementation

- difficult to output diagrams in PlantUML as labels are used to identify objects in the diagrams (but there exist multiple objects with the same label) - only limited help for debugging
- choose left or right and first or second in corners/spans? (missing documentation!)

$\Rightarrow$ code generation from category diagrams would be great!

Literature

Hartmut Ehrig, Karsten Ehrig, Ulrike Prange, and Gabriele Taentzer:
Fundamentals of Algebraic Graph Transformation.
Monographs in Theoretical Computer Science. An EATCS Series.
Springer, 2006.
Sections 7.2 and 7.3 (pp. 156-164)