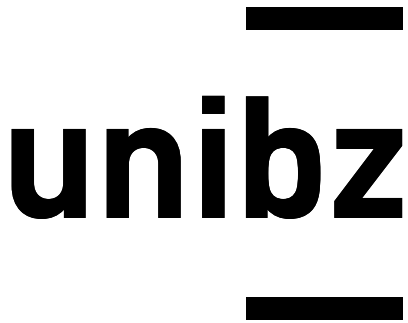
Data and Process Modelling

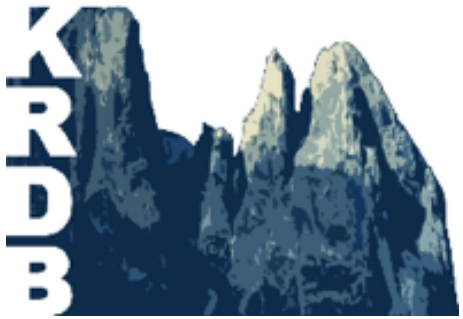
6. Schema Transformations

Marco Montali

KRDB Research Centre for Knowledge and Data
Faculty of Computer Science
Free University of Bozen-Bolzano

A.Y. 2014/2015





One Schema, Many Schemas

- The same domain may be mirrored in many (equivalent) conceptual schemas.
  - Equivalence means having the same intended models.
- The same conceptual schema may be mapped to different logical schemas.
- The same logical schema may be physically realized in different ways.

Questions

How to transform a conceptual schema into an equivalent conceptual schema? How to choose the "optimal" representation, considering in particular efficiency of database access?

Example: Persons and their Vices (on the whiteboard).

Predicate Specialization/Generalization

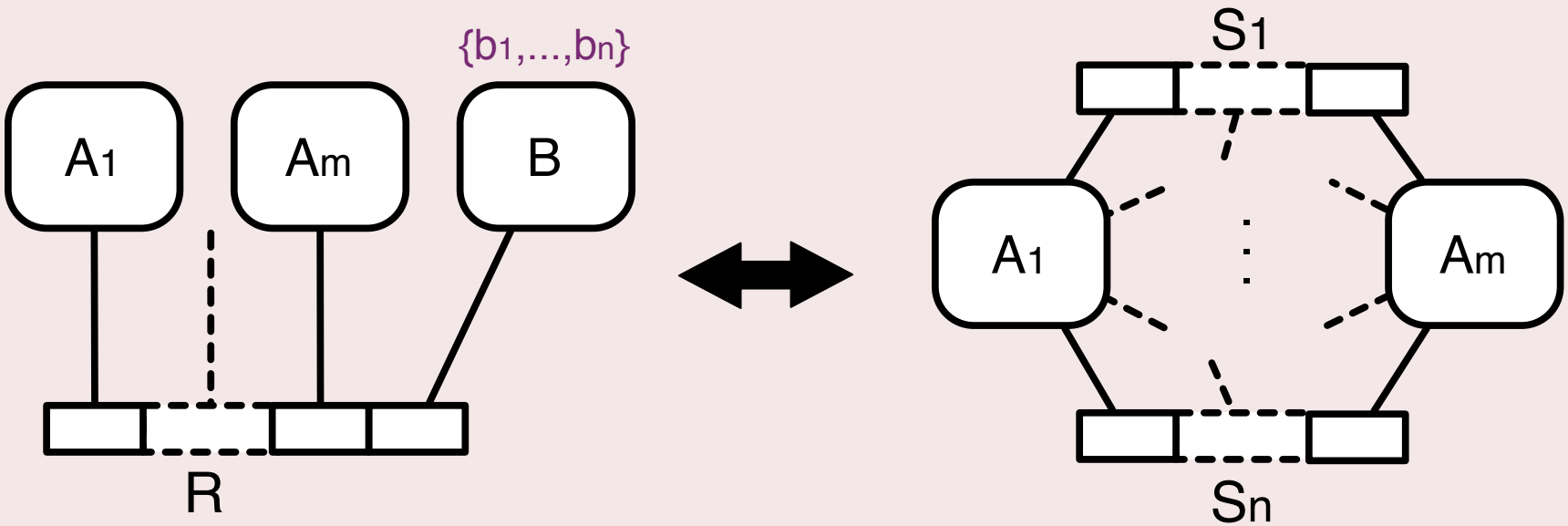
The transformation of a conceptual schema into an equivalent schema that substitutes a predicate with a more specific/general predicate, still preserving the key constraints.

Applied when the predicate has a finite number of cases, i.e.:

- is subject to a value constraint, or
- is subject to a frequency constraint.

Predicate Specialization/Generalization 1

PSG1



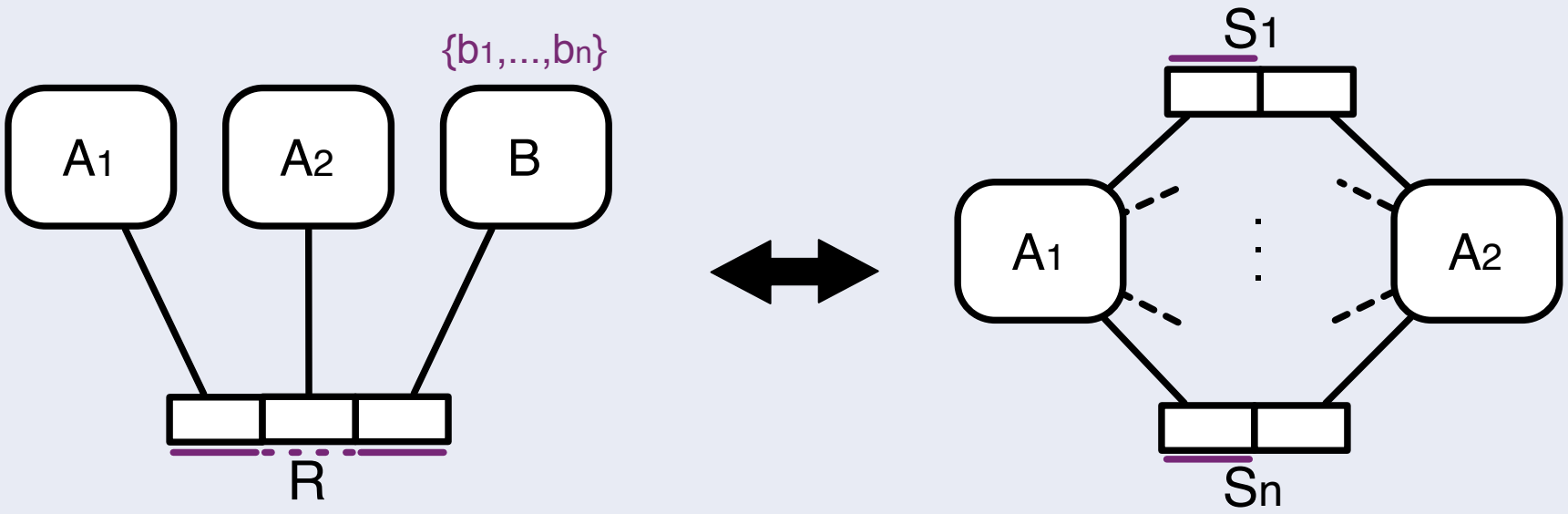
- $m \geq 1$
- S_i corresponds to R with B instantiated with b_i

Constraints have to be suitably translated as well.
See following corollaries (shown with ternary fact types, but they hold for any arity).

Predicate Specialization/Generalization 1

UC over B's role (and others).

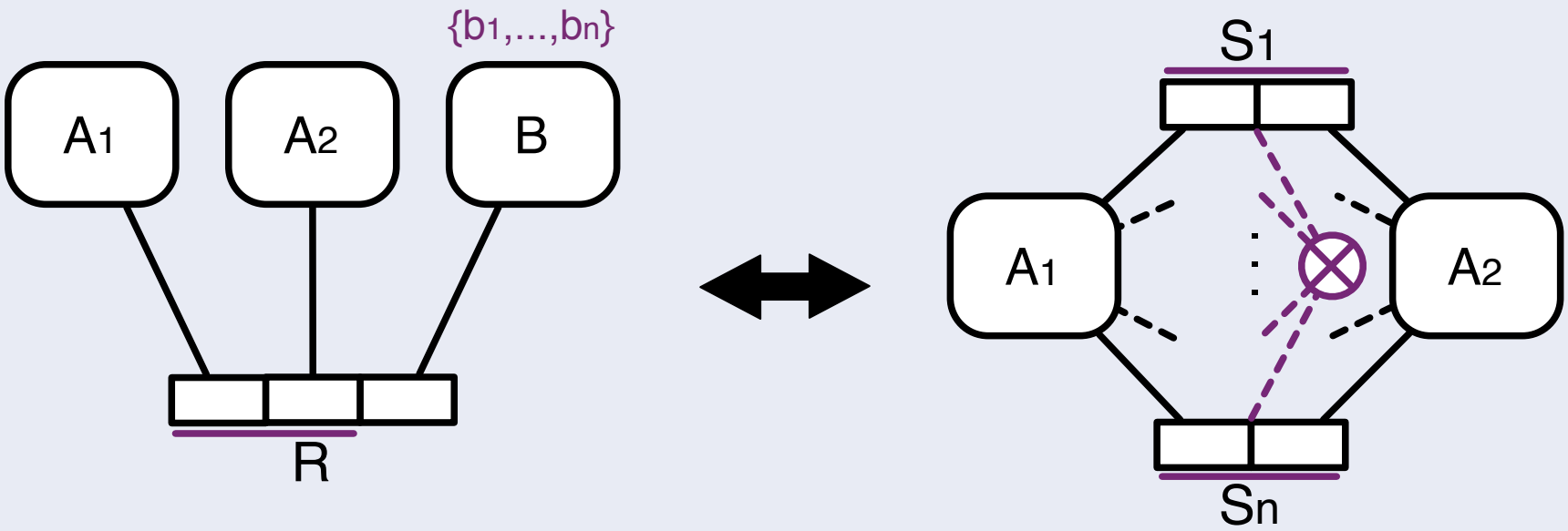
PSG1 - Corollary 1



Predicate Specialization/Generalization 1

UC over all roles but B's.

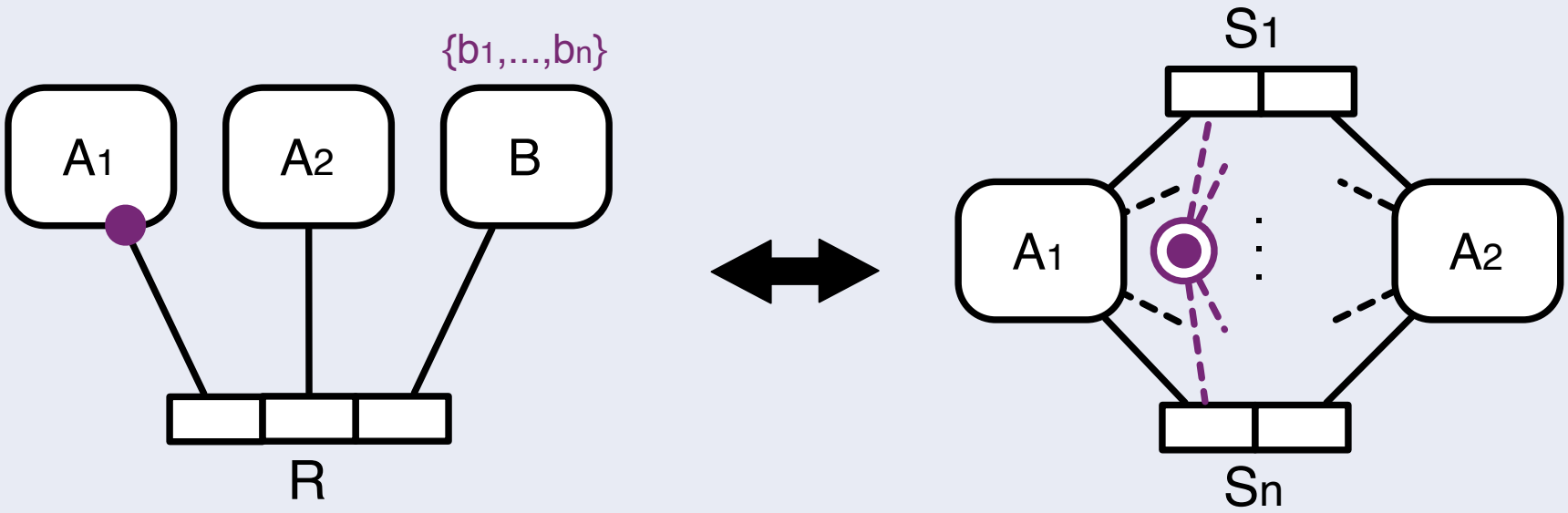
PSG1 - Corollary 2



Predicate Specialization/Generalization 1

Mandatory participation of an entity type.

PSG1 - Corollary 3



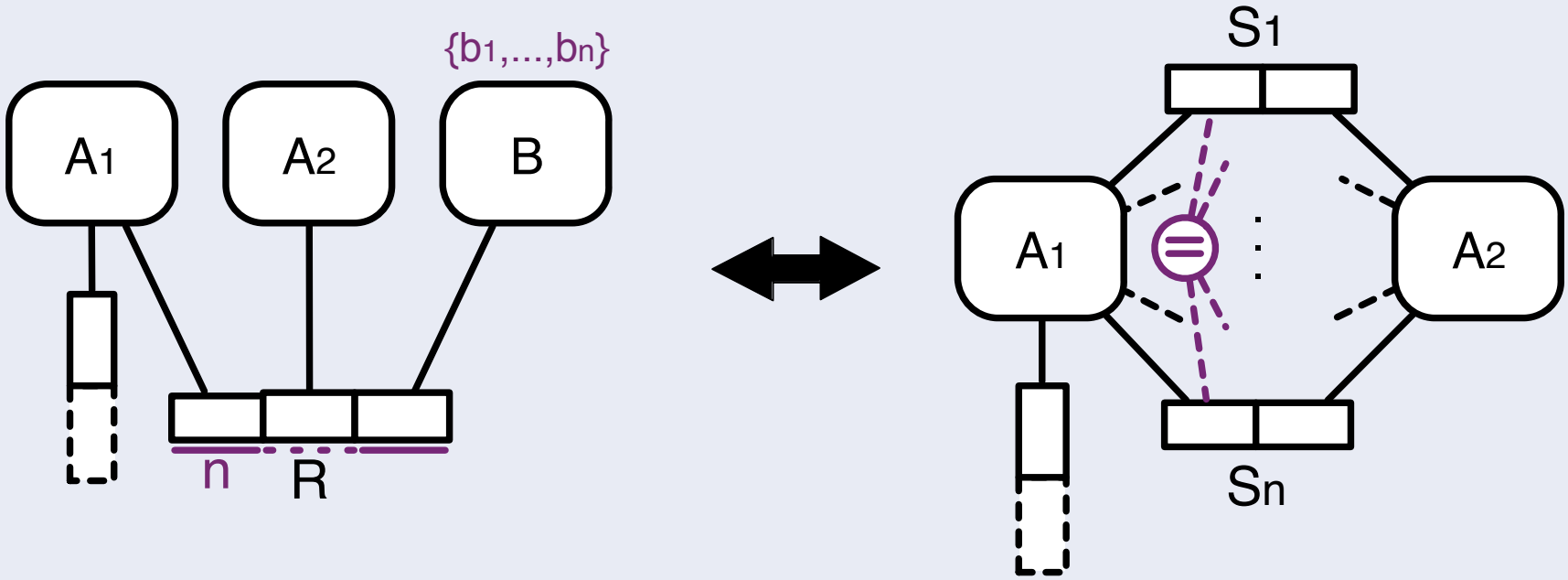
Works also for a mandatory role disjunction of an entity type.

- Remember that the same entity type can play multiple roles in the same fact type.

Predicate Specialization/Generalization 1

Optional participation with frequency constraint.

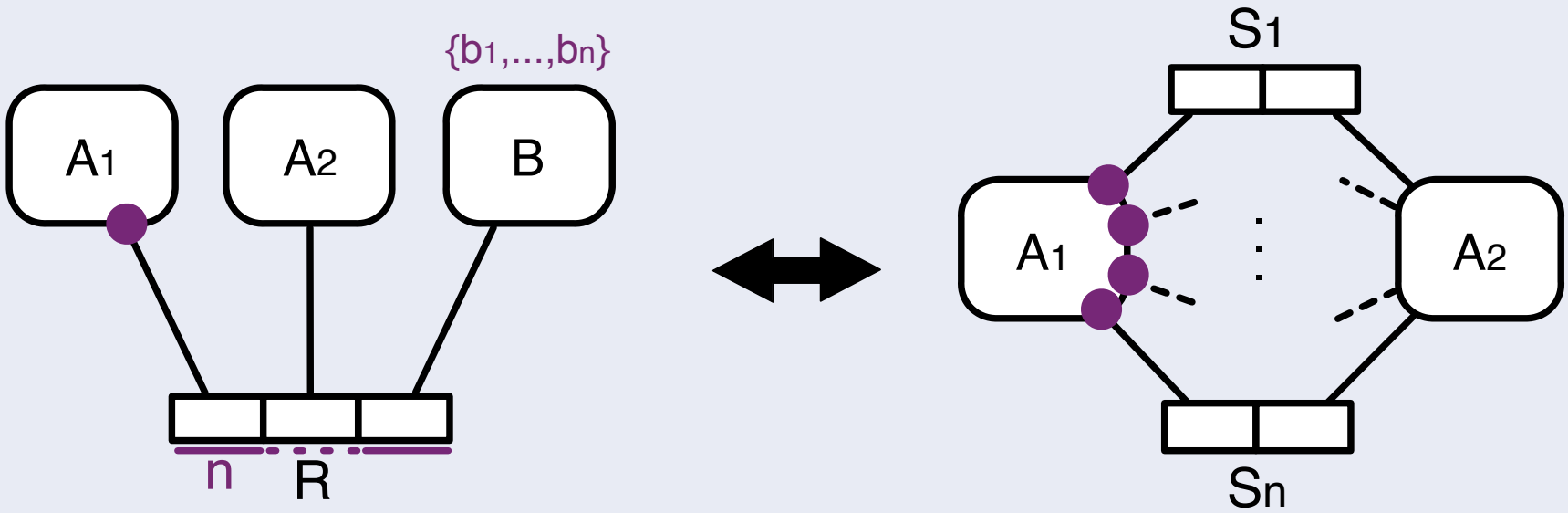
PSG1 - Corollary 4



Predicate Specialization/Generalization 1

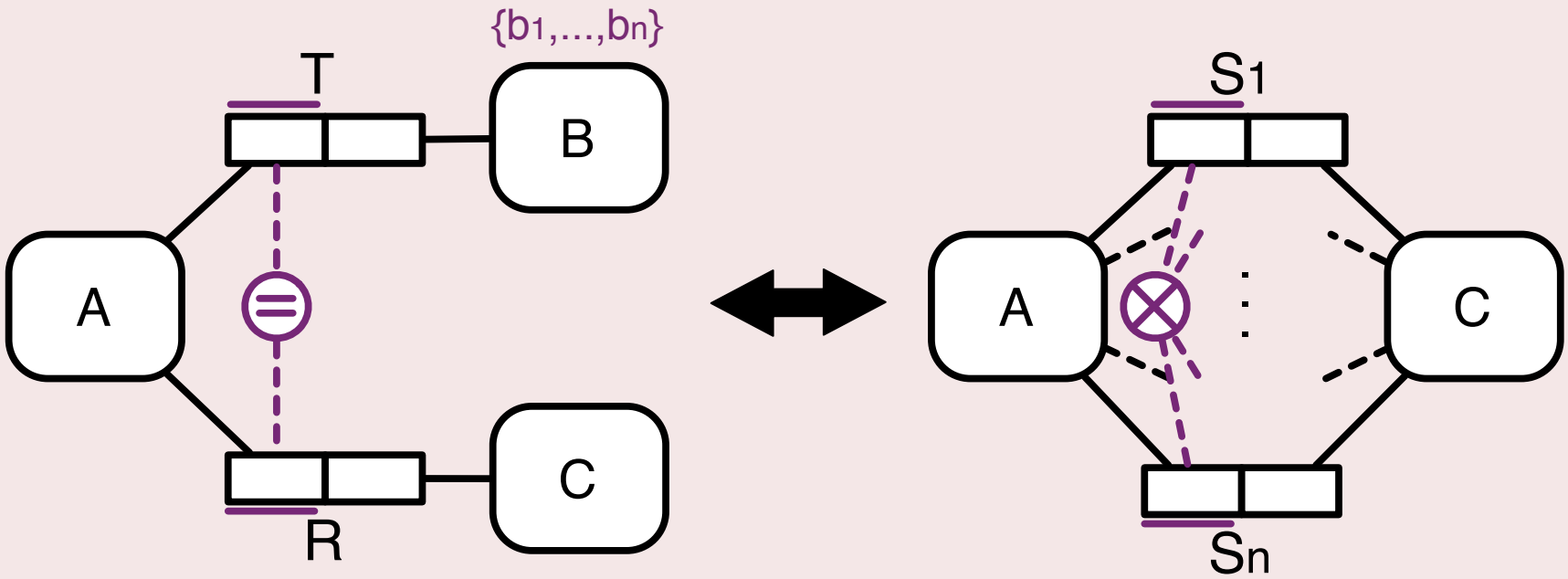
Mandatory participation with frequency constraint.

PSG1 - Corollary 5



Predicate Specialization/Generalization 2

PSG2

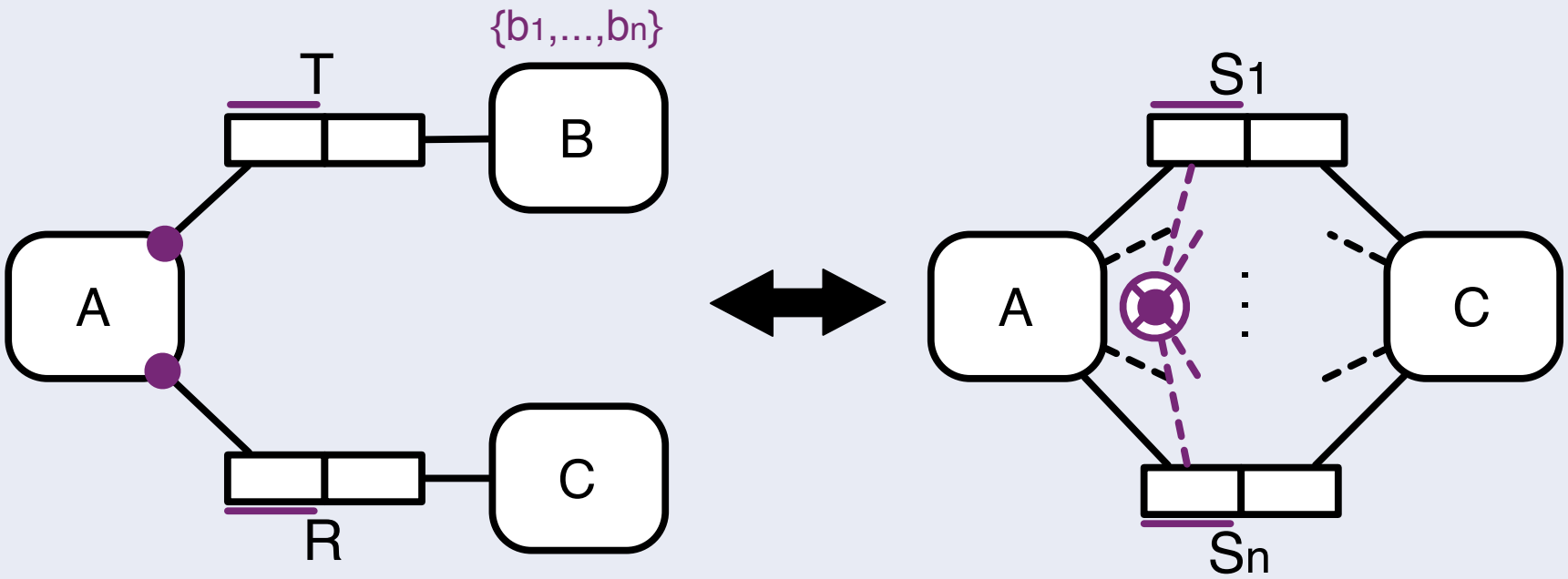


- S_i corresponds to R where T is restricted to instance b_i of B.

Predicate Specialization/Generalization 2

Mandatory roles of A.

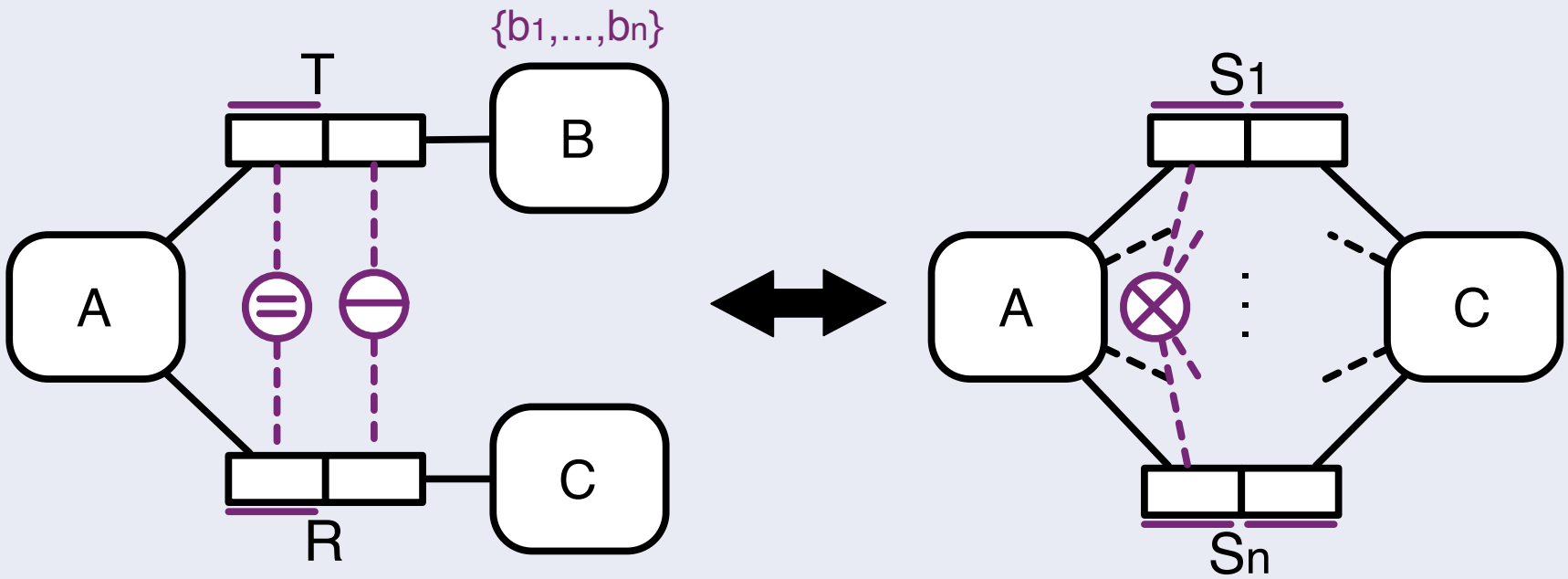
PSG2 - Corollary 1



Predicate Specialization/Generalization 2

External UC over B and C.

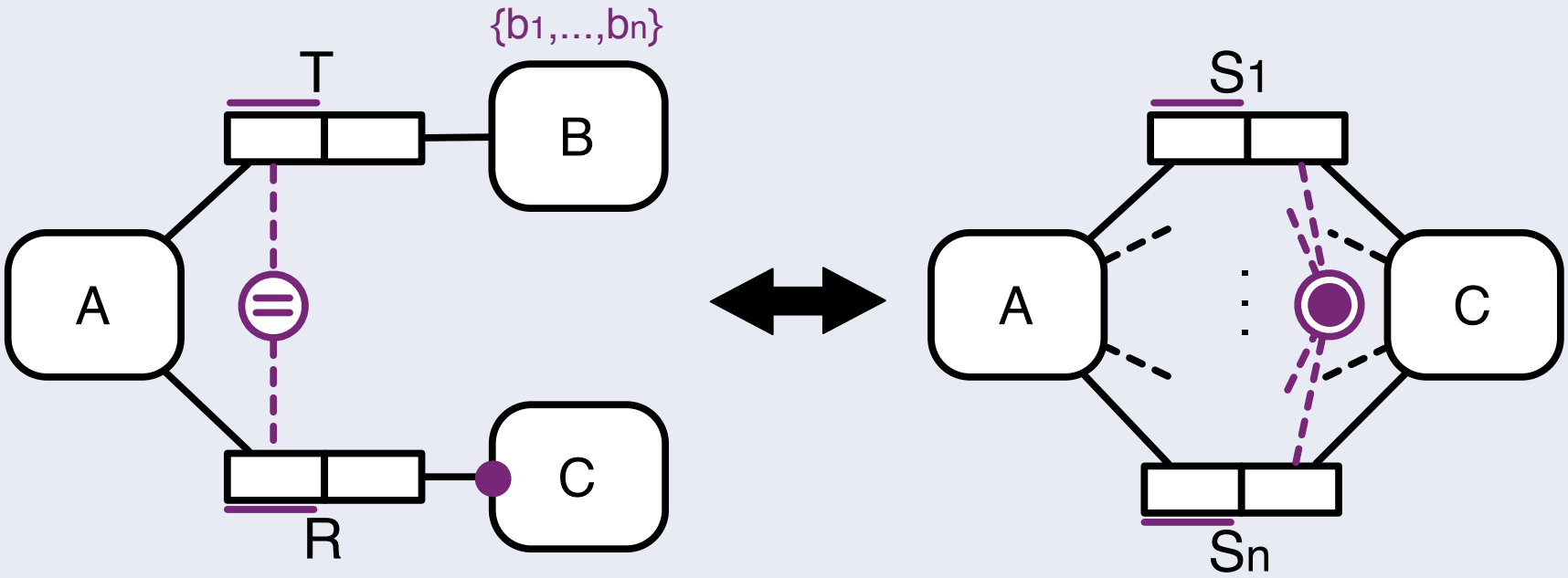
PSG1 - Corollary 2



Predicate Specialization/Generalization 2

Mandatory participation of C.

PSG1 - Corollary 3

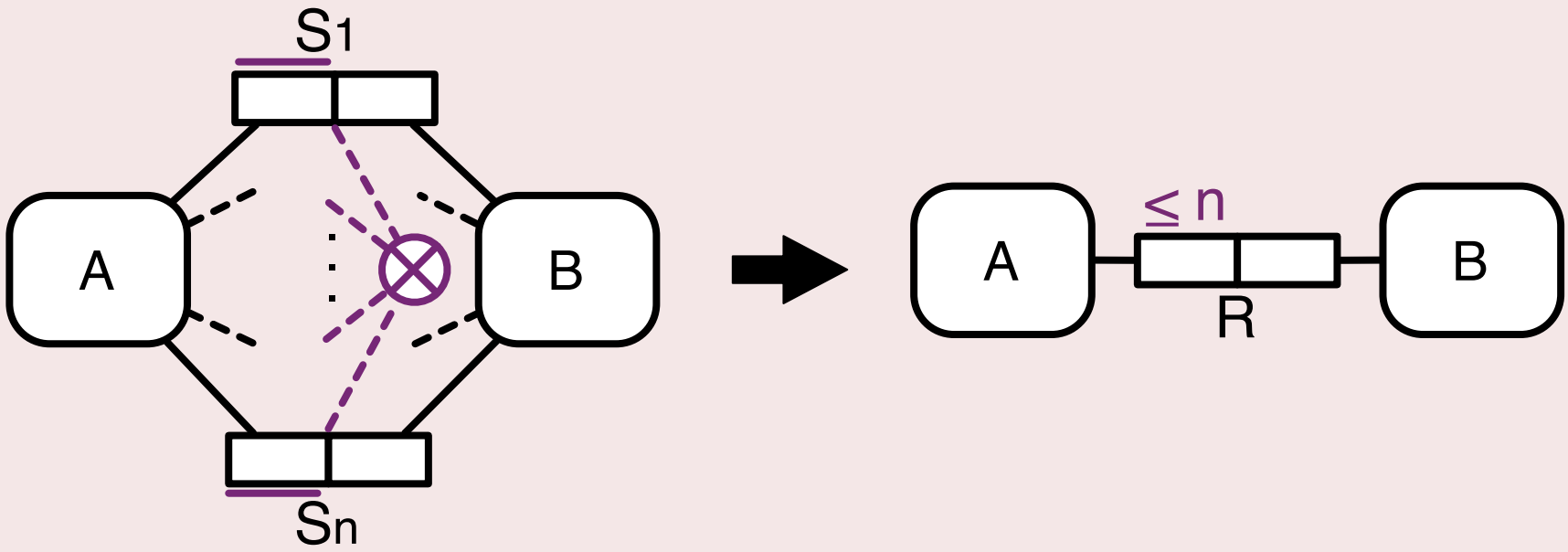


Predicate Specialization/Generalization 3

A non-equivalent transformation.

- From left to right: weakening.
- From right to left: strengthening.

PSG3

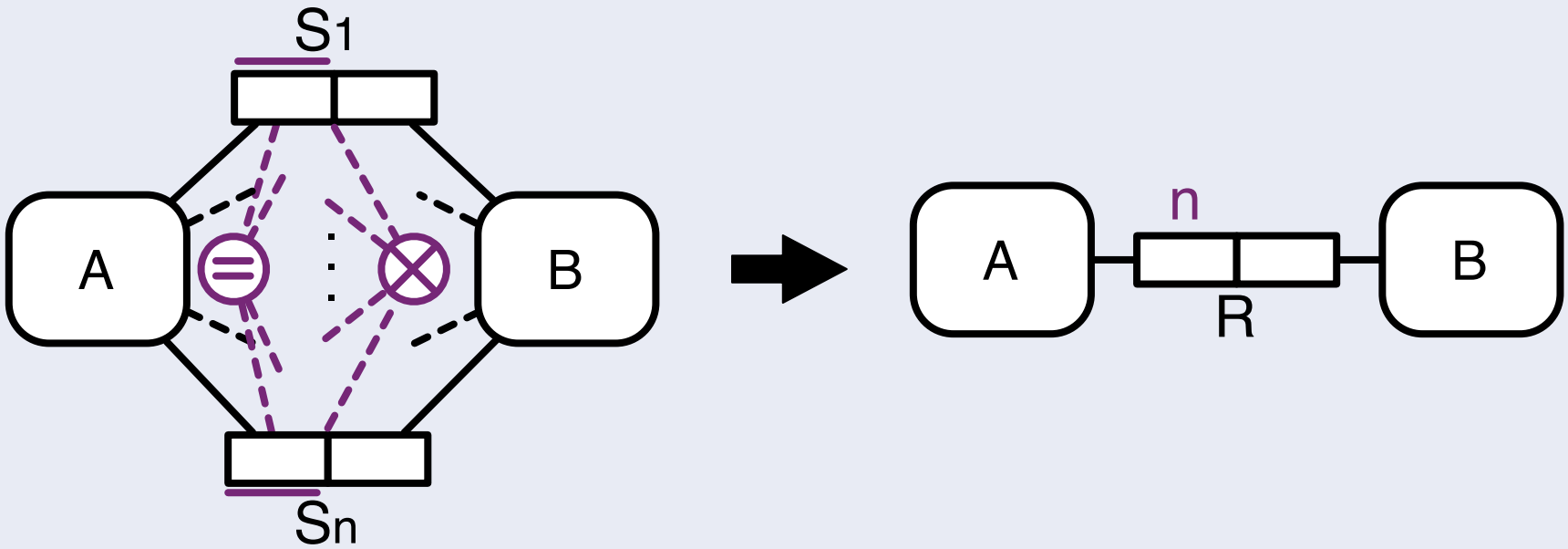


- Each S_i corresponds to one instance of R.

Predicate Specialization/Generalization 3

Equality constraint on A's roles.

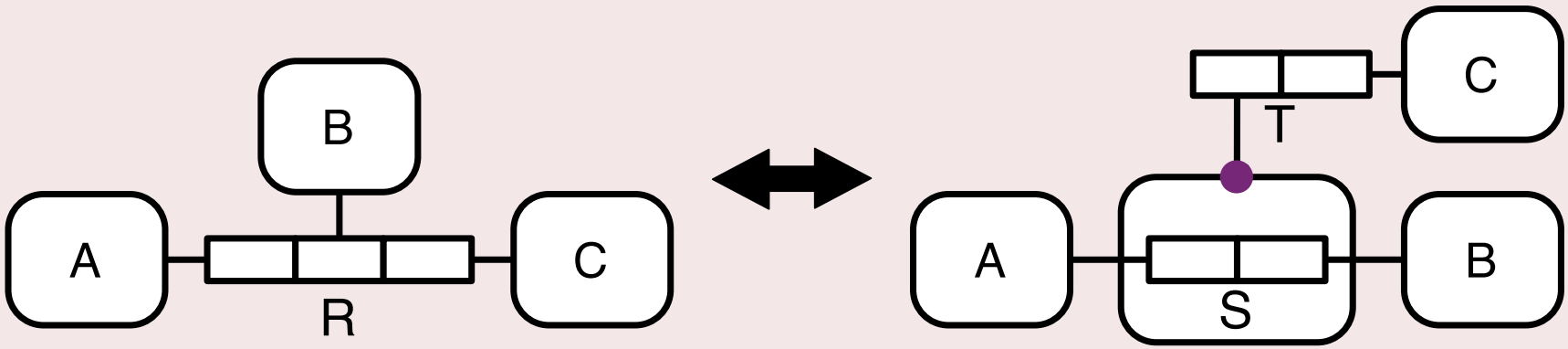
PSG3 - Corollary 1



Nested/Flat Predicates

N.B.: we already saw the correspondence between nesting and co-reference.

N/F

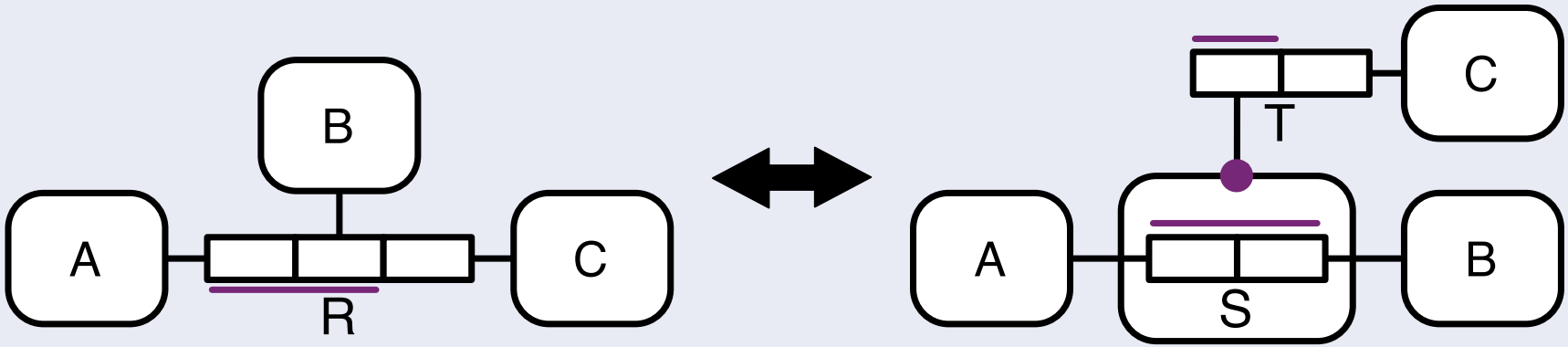


where Rabc if and only if (a,b) Tc.

Nested/Flat Predicates

UC over nested roles.

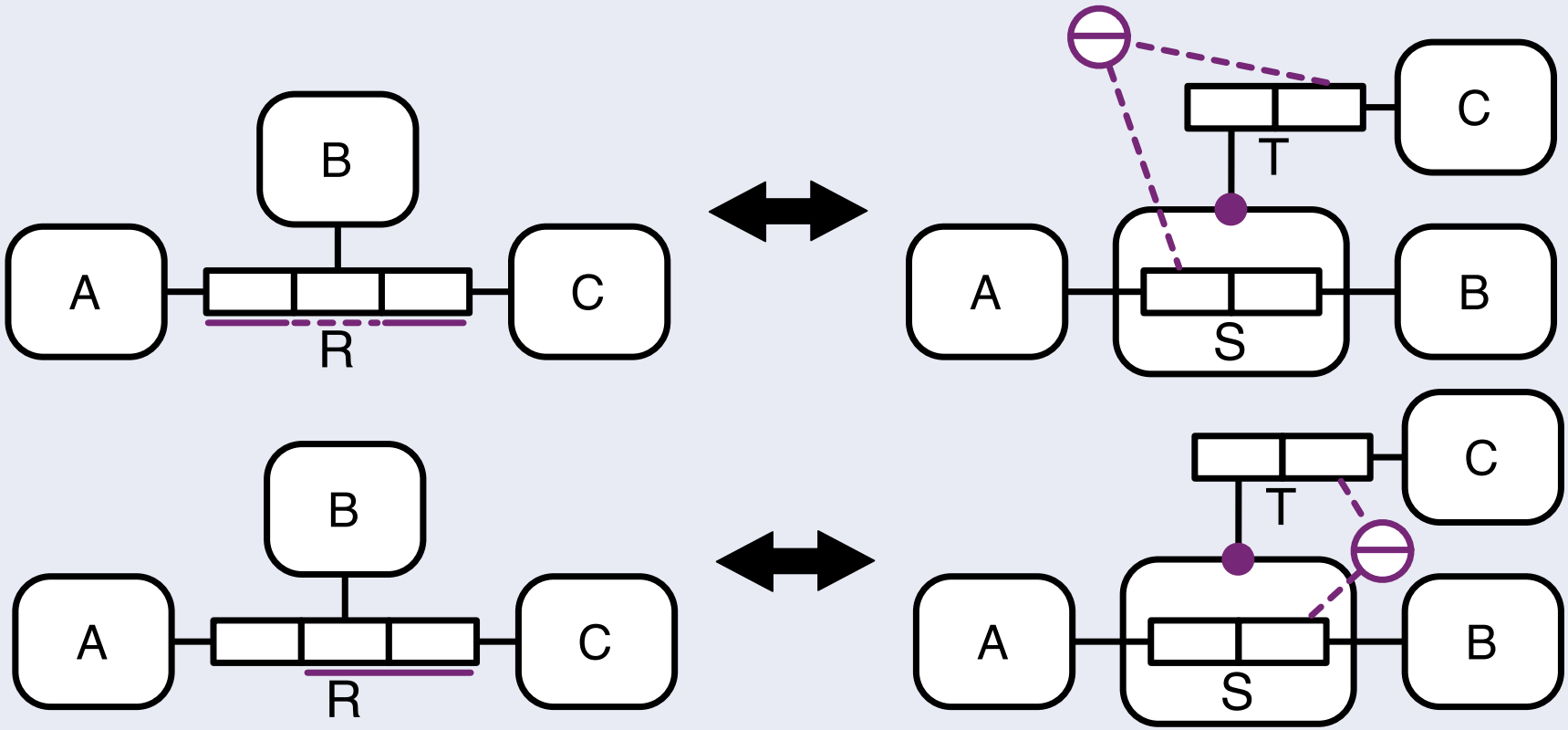
N/F - Corollary 1



Nested/Flat Predicates

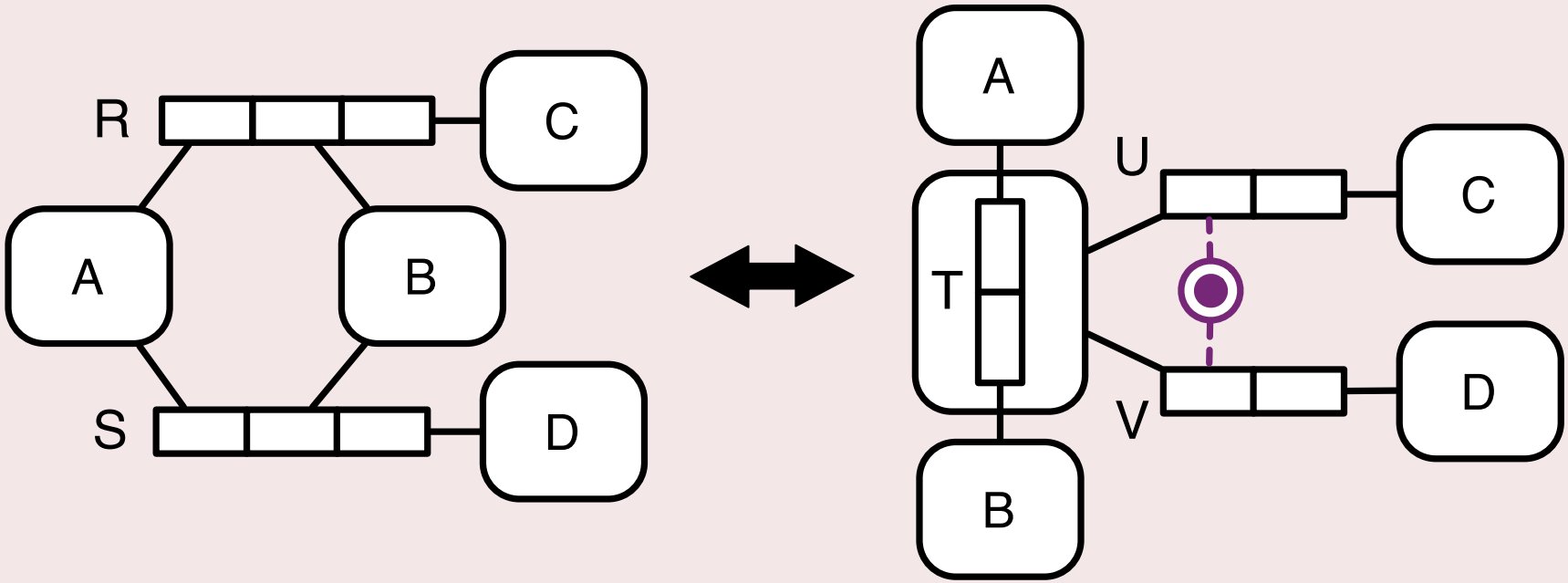
UC over a nested and non-nested role.

N/F - Corollary 2



Nested/Flat Predicates - Extended

Ext. N/F



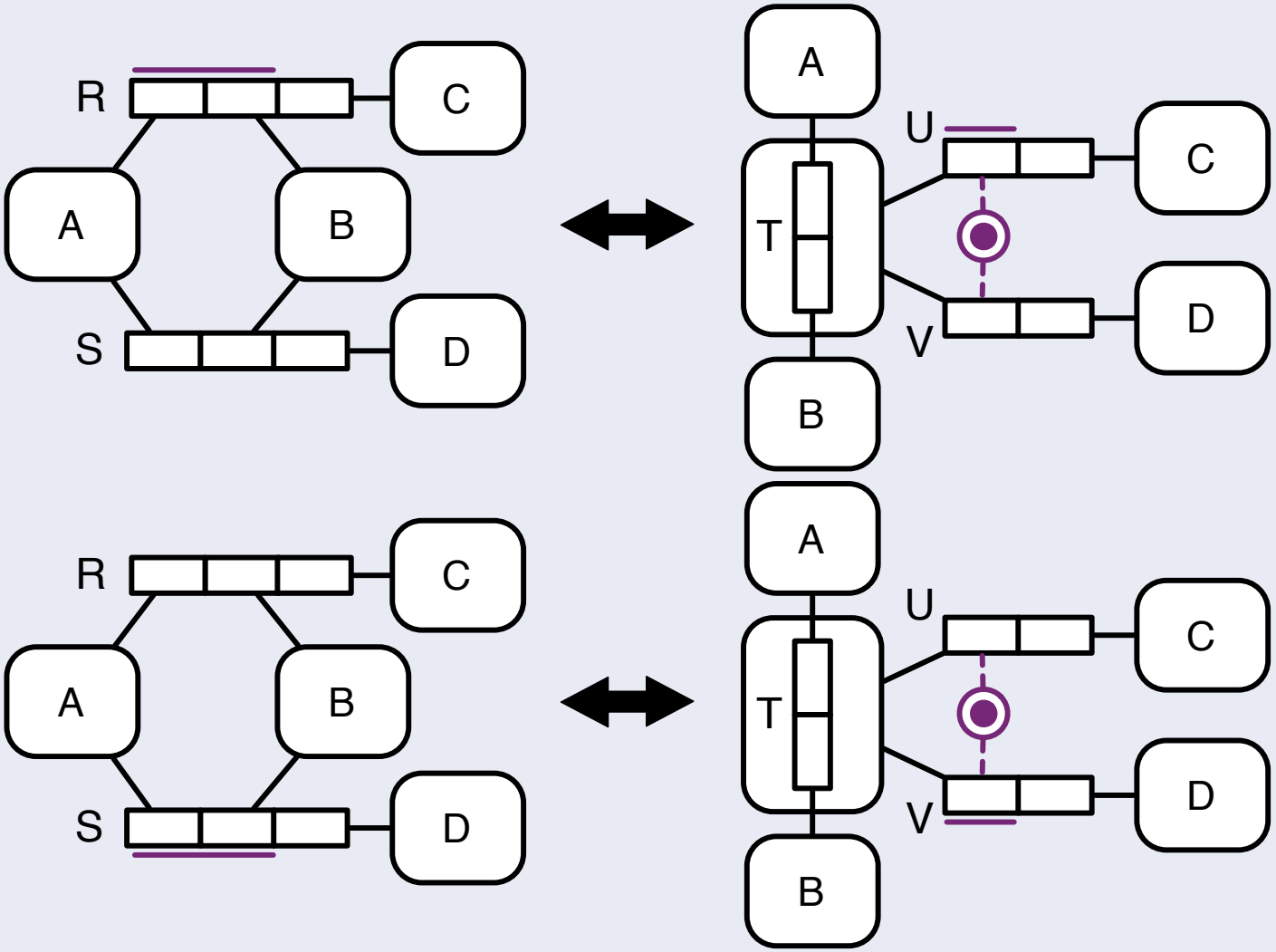
where:

- Rabc if and only if (a,b) Uc.
- Sabc if and only if (a,b) Vc.
- $T = R$[a,b] $\cup$ S[a,b].

Nested/Flat Predicates - Extended

UC just over nested roles.

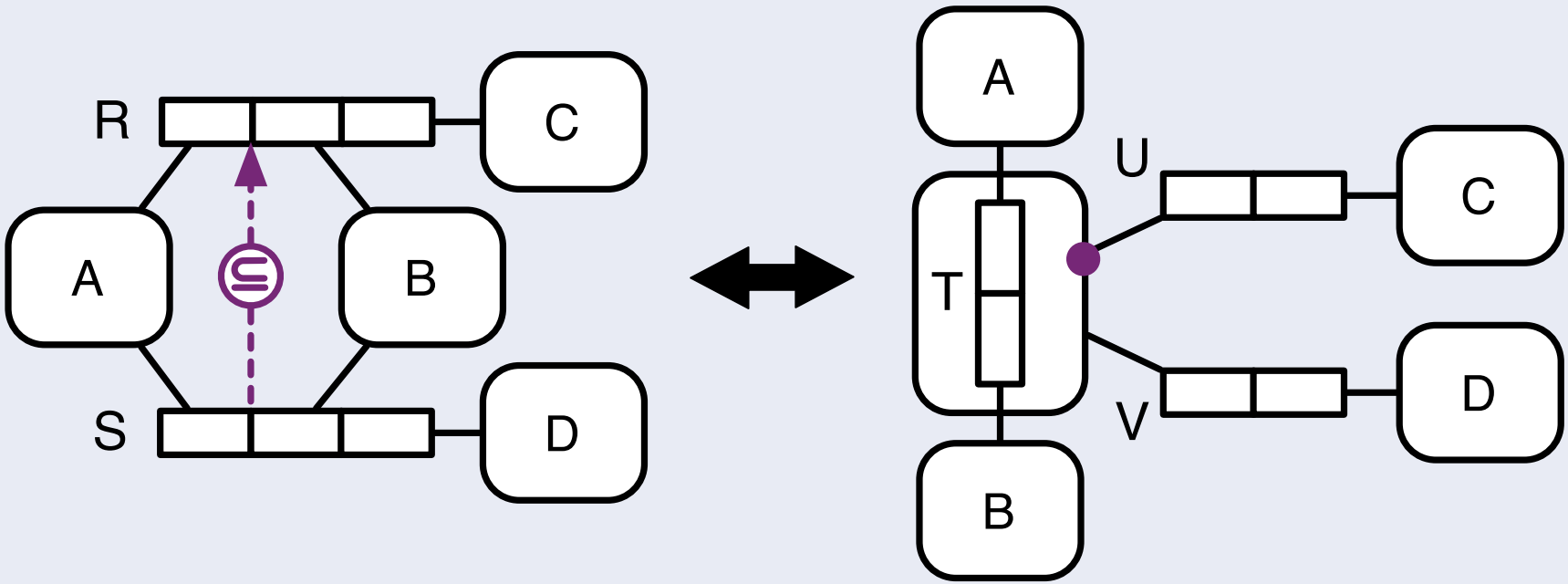
Ext. N/F - Corollary 1



Nested/Flat Predicates - Extended

Pair-subset constraint.

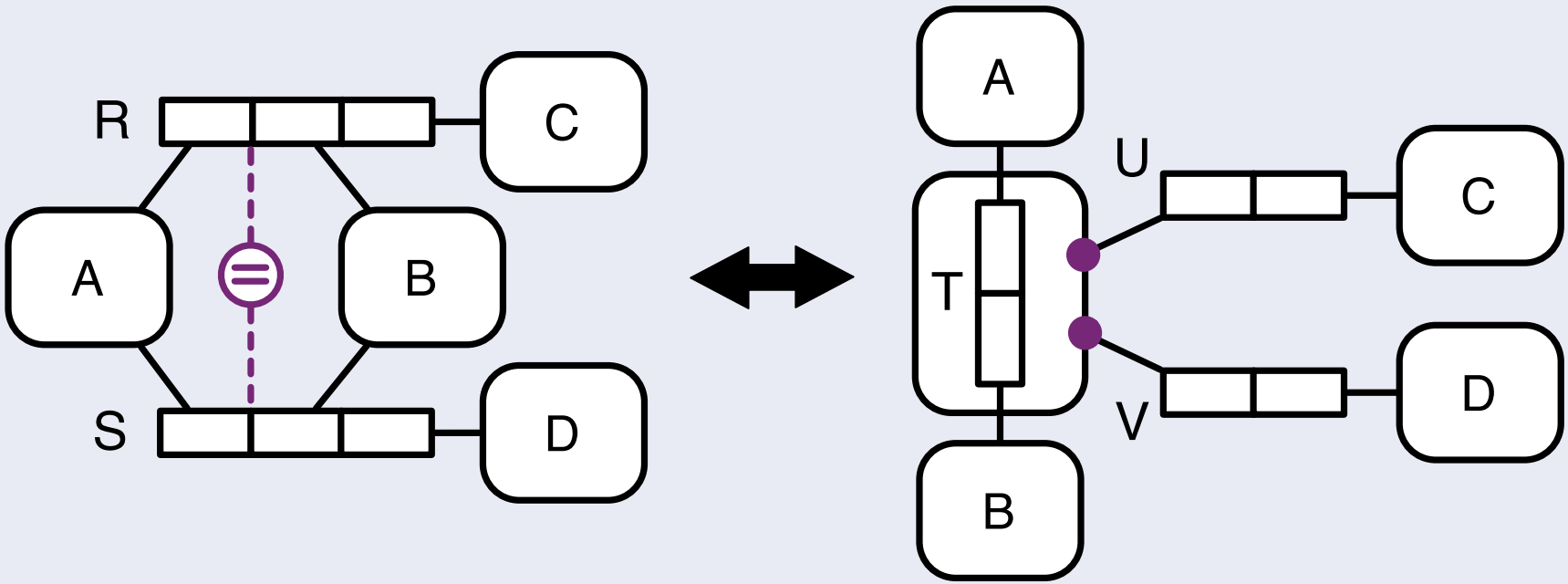
Ext. N/F - Corollary 2



Nested/Flat Predicates - Extended

Pair-equality constraint.

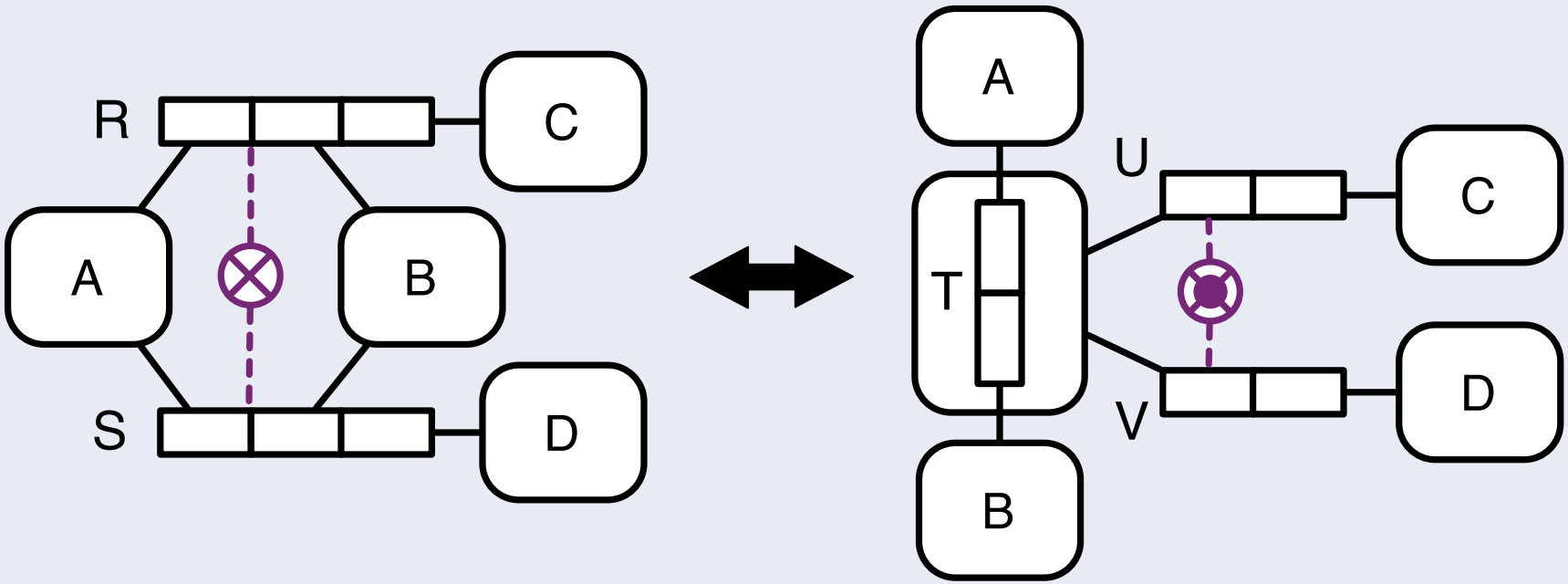
Ext. N/F - Corollary 3



Nested/Flat Predicates - Extended

Pair-exclusion constraint.

Ext. N/F - Corollary 4



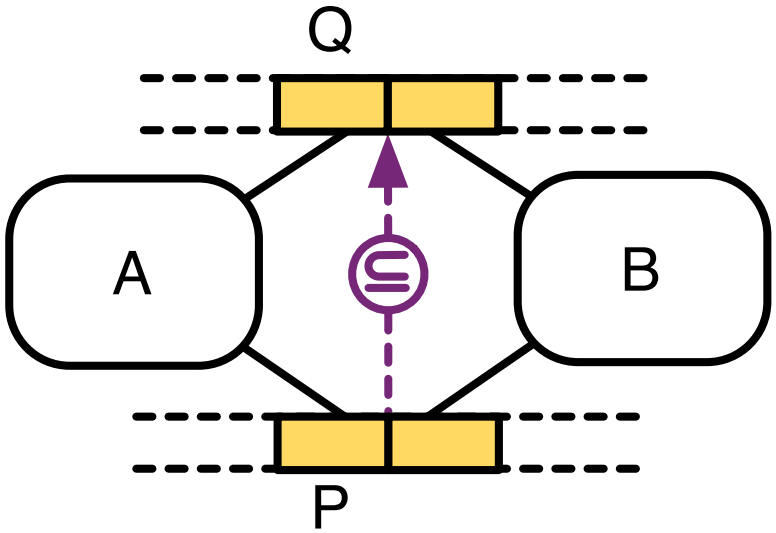
Overlap Algorithm

Guides nesting when two or more fact types contain compatible role sequences (of at least two roles), with (possible) overlapping tuples.

The algorithm has two parameters:

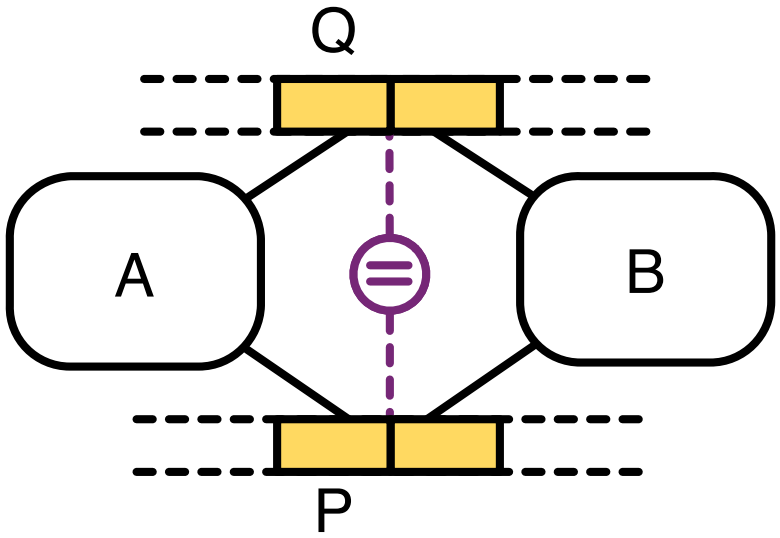
1. Amount of overlapping: one subset of the others, total overlapping, no overlapping, possible overlapping.
   - Determined by the constraint attached to the role sequence
2. Role sequences cover the entire relation (whole relation) vs they are just a projection of larger relations (partial relation).

Overlap Algorithm - Case 1



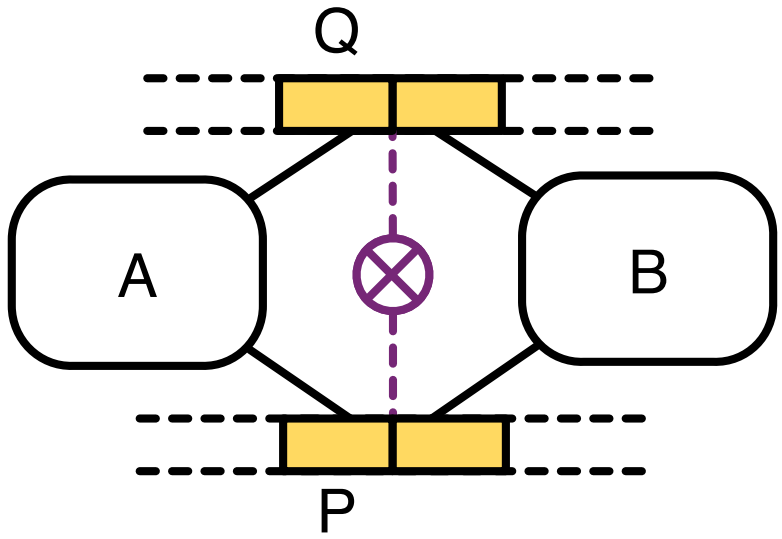
1. Objectify Q.
2. **If** Q is *partial*
   Then
   2.1 Attach rest of its predicate via mandatory extra role.
3. **If** P is *partial*
   Then
   3.1 Attach rest of its predicate via optional extra role.
   Else
   3.2 Attach optional unary role “is-P”.

Overlap Algorithm - Case 2



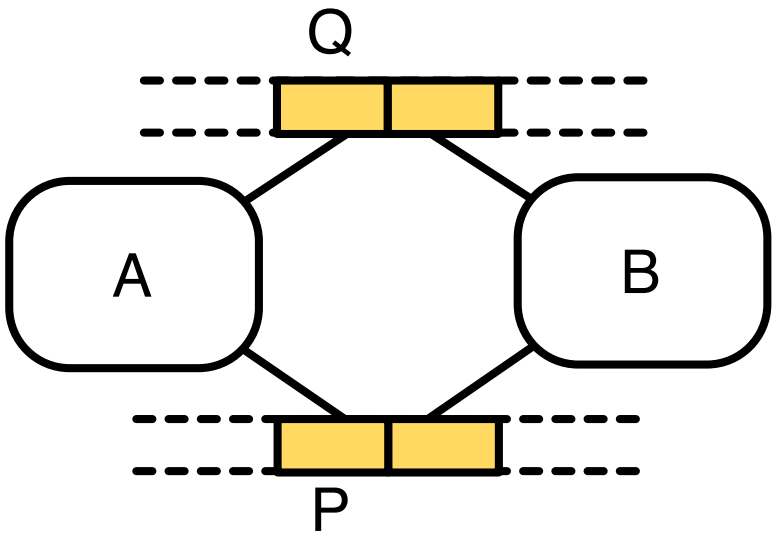
1. **If** P or Q is *partial*
   Then
   1.1 Objectify the single predicate "P and Q".
   1.2 Attach rest of their predicates via mandatory extra role(s).
   Else
   1.3 Replace both P and Q with a single binary predicate "P and Q".

Overlap Algorithm - Case 3



1. Objectify "P or Q".
2. **If** P is *partial*
   Then
   2.1 Attach rest of its predicate via optional extra role.
   Else
   2.2 Attach unary role "is-P".
3. Do the same with Q.
4. Mark the attached roles with an exclusive and disjunctively mandatory constraint.

Overlap Algorithm - Case 4



1. Objectify "P or Q".
2. **If** P is *partial*
   Then
   2.1 Attach rest of its predicate via optional extra role.
   Else
   2.2 Attach unary role "is-P".
3. Do the same with Q.
4. Mark the attached roles with a disjunctively mandatory constraint.

Schema Optimization

The transformation of a conceptual schema into an alternative conceptual schema that maps more efficiently to a relational schema.

Four factors are involved:

1. Target system (which technology used to deploy the system).
2. Query pattern (which questions the system must be able to answer).
3. Update pattern (which expected insertions, deletions, modifications).
4. Clarity (to which extent the obtained relational schema is understandable).

Two main steps:

1. Reduce the number of mapped tables.
2. Simplify individual tables.

Example

On the whiteboard!