Algebraic Graph Transformation: NACs and Graph Constraints

Fernando Orejas

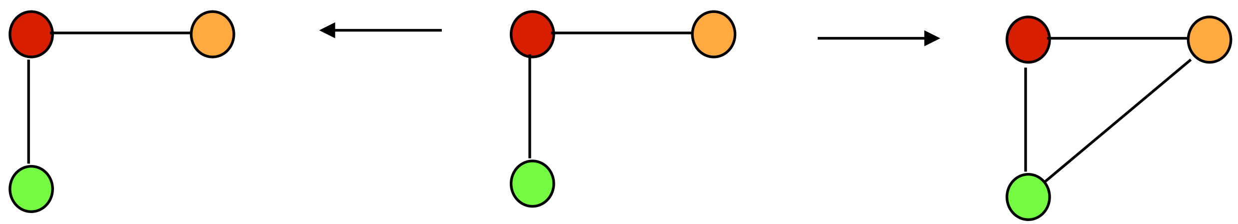
Royal Holloway University of London

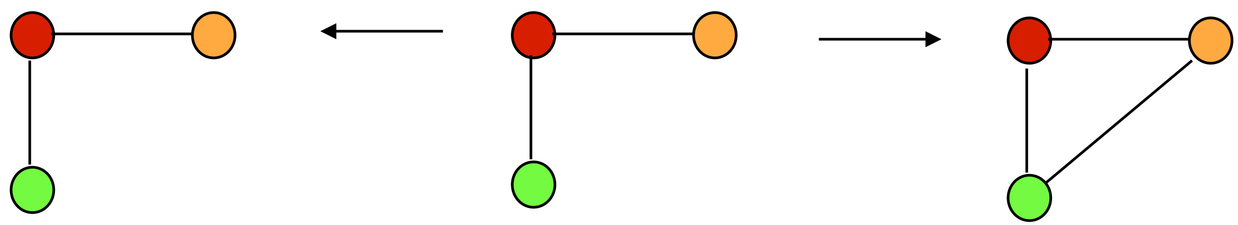
on leave from Universitat Politècnica de Catalunya, Barcelona

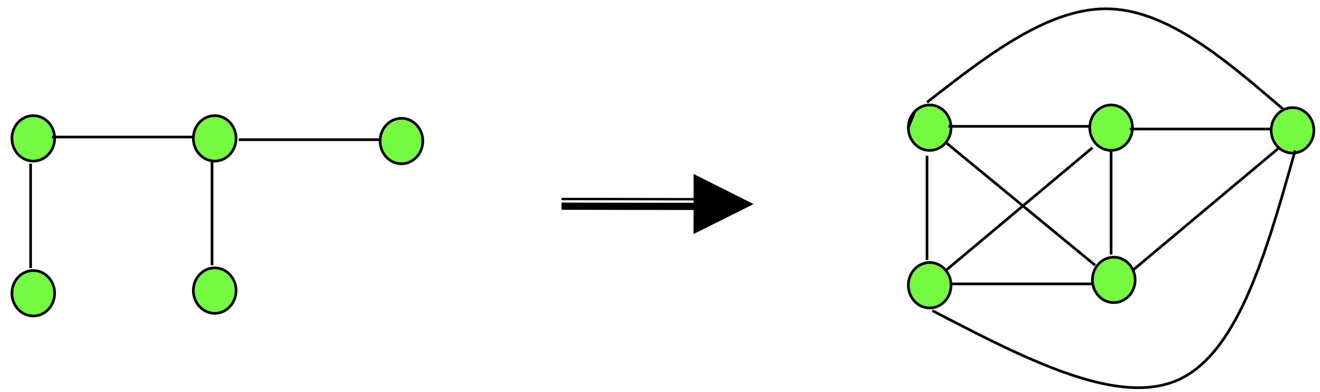
1. Introduction
2. NACs
3. Parallel Independence and Critical Pairs for Rules with NACs
4. Graph Constraints
5. Modelling and Reasoning with Constraints

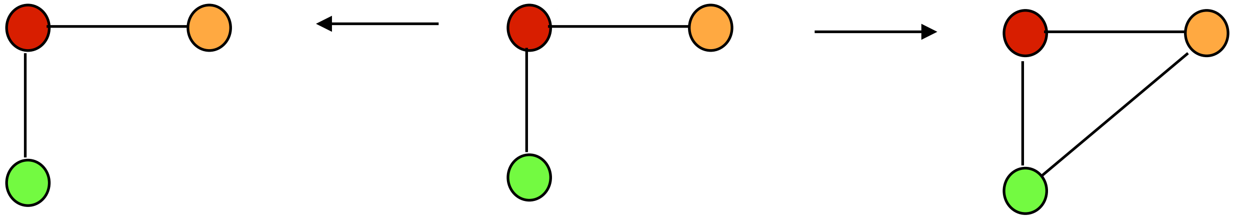
Introduction

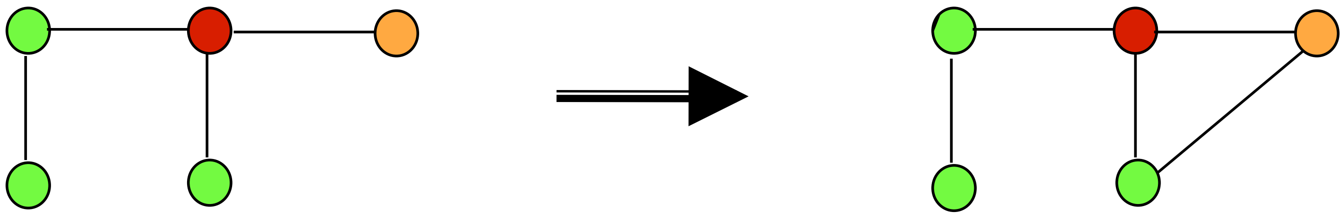
NACs and Constraints

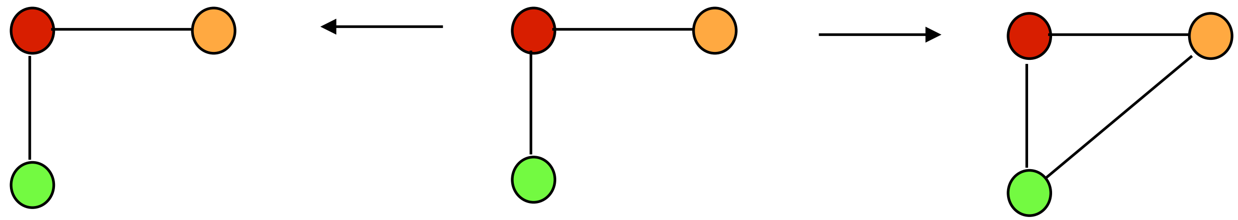


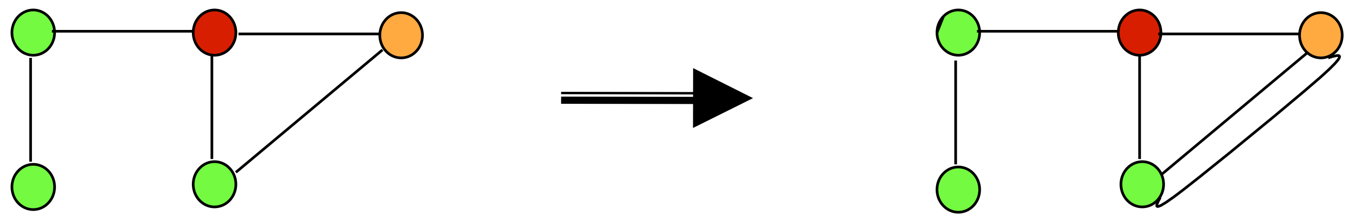


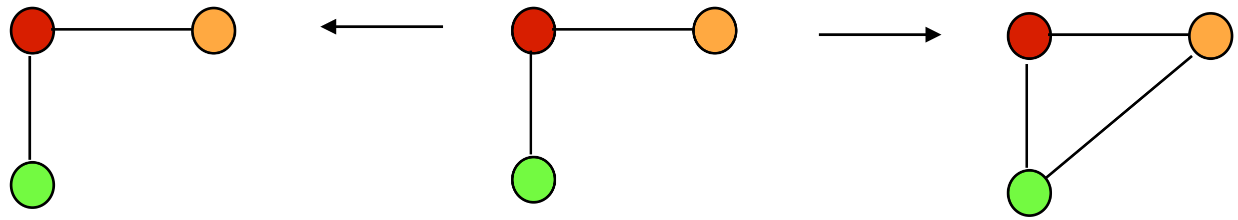


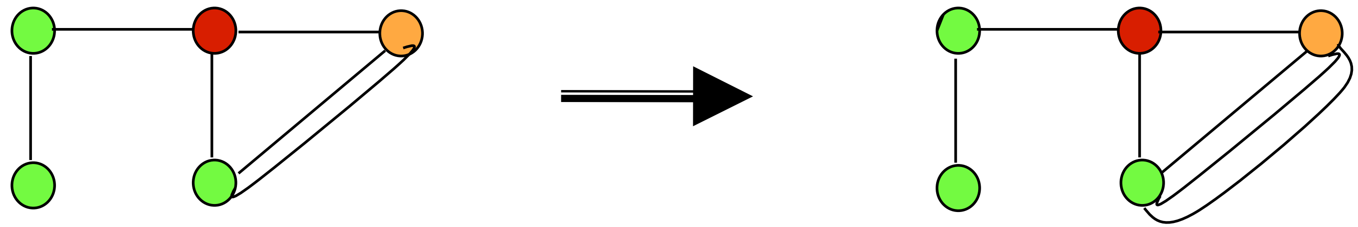


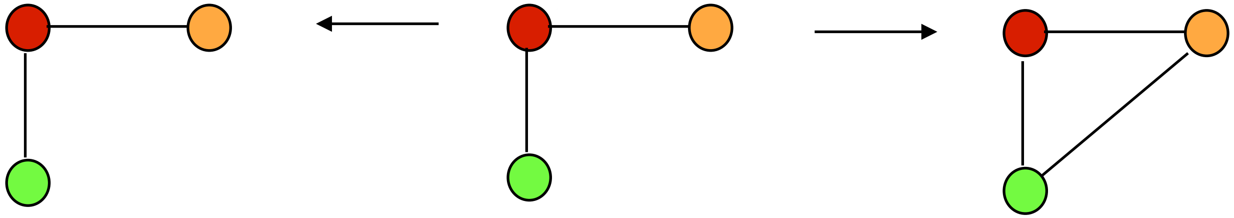


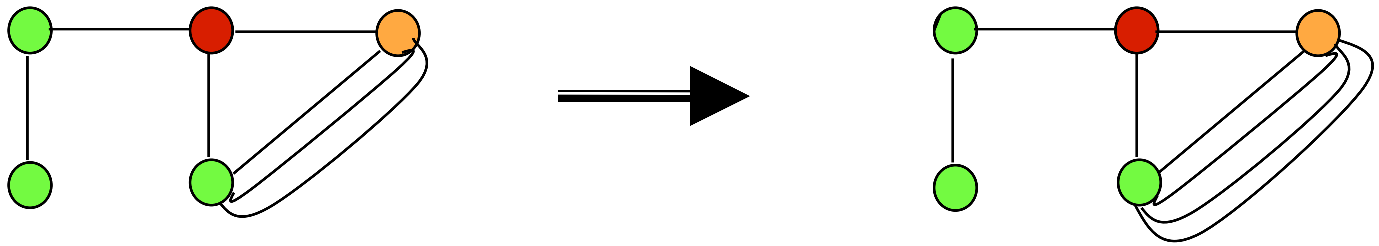












Negative Application Conditions

(Negative) Application Conditions (NACs)

Given a rule

$$p = L \xleftarrow{l} K \xrightarrow{r} R$$

a negative application condition is an inclusion

$$L \rightarrow C \quad \text{or} \quad L \rightarrow X \rightarrow C$$

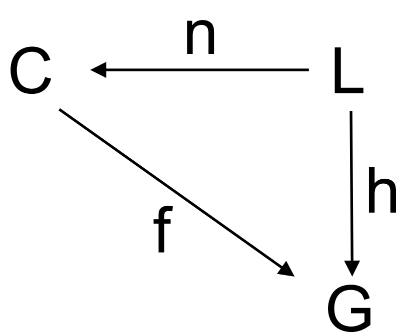
or

$$R \rightarrow C \quad \text{or} \quad R \rightarrow X \rightarrow C$$

NAC satisfaction:

$(h: L \rightarrow G) \models L \rightarrow C$

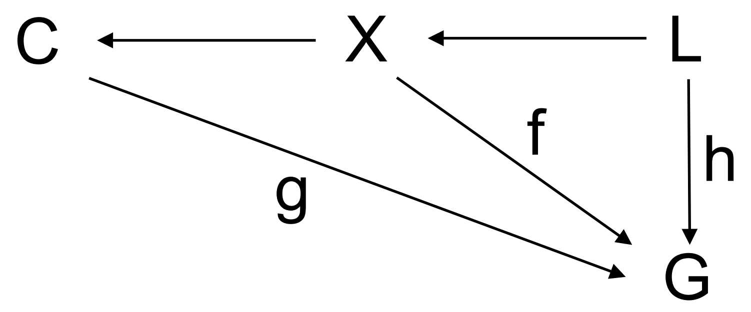
there is no $f: C \rightarrow G$ such that the diagram commutes



NAC satisfaction:

(h: L → G) |= L → X → C

if for every f: X → G such that the left triangle commutes
there is a g: C → G such that the right triangle commutes



Graph Transformation with NACs

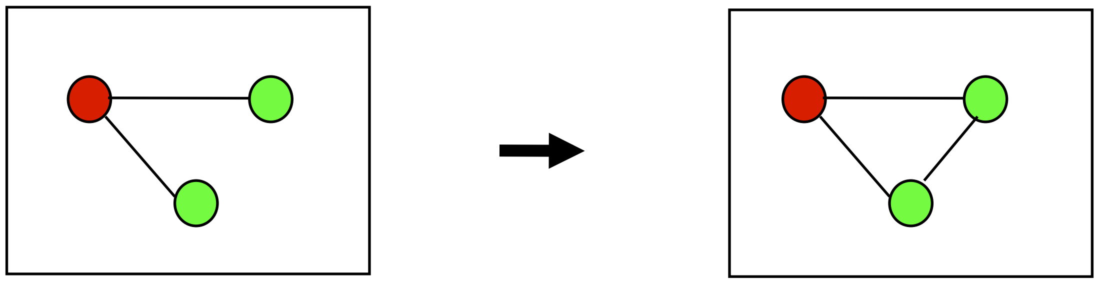
Given the rule

$$p = L \xleftarrow{l} K \xrightarrow{r} R$$

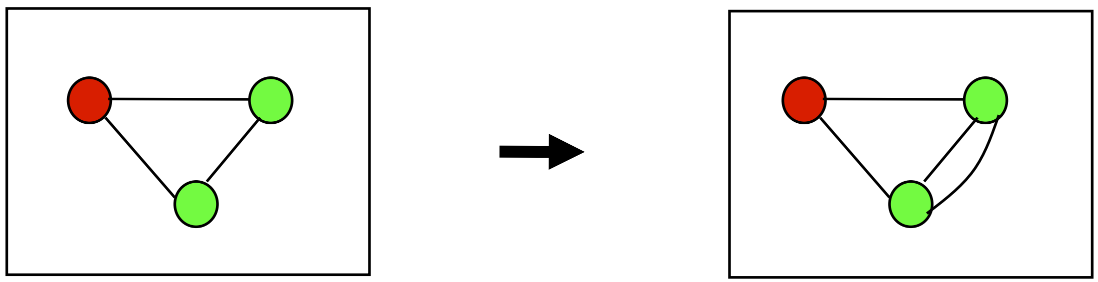
together with NACs (N_L, N_R), we can apply p to G vía m:

$$\begin{array}{ccccc} L & \xleftarrow{l} & K & \xrightarrow{r} & R \\ {\scriptstyle m}\downarrow & & \downarrow & & \downarrow{\scriptstyle m'} \\ G & \longleftarrow & D & \longrightarrow & H \end{array}$$

if m satisfies all the NACs in N_L and m' satisfies all the NACs in N_R



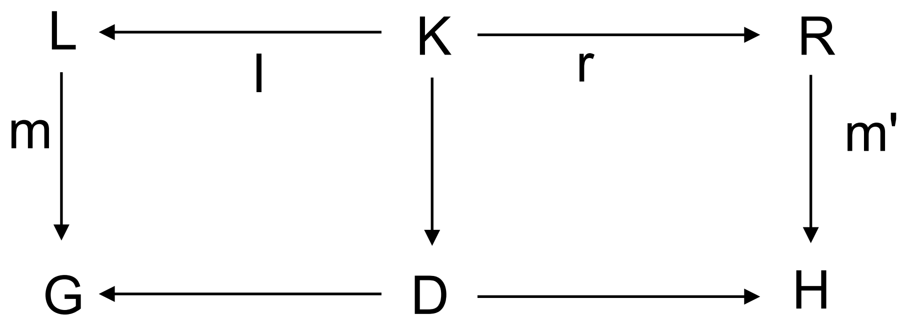
L C



R C

Right NACs elimination

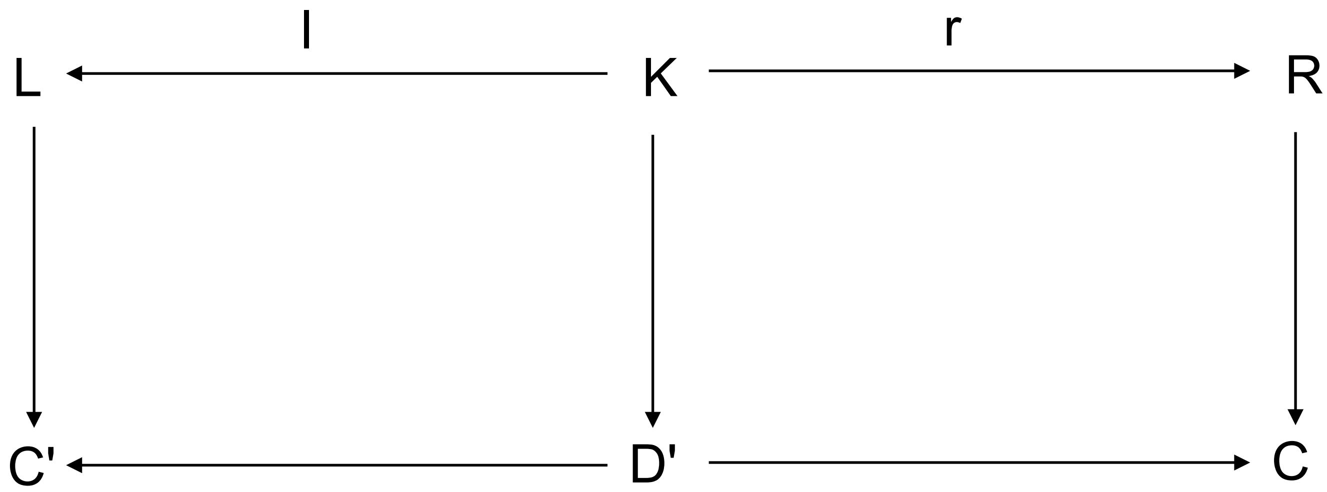
If R → C is a right NAC, there is a left NAC L → C' such that:



m satisfies L → C' if and only if m' satisfies R→ C .

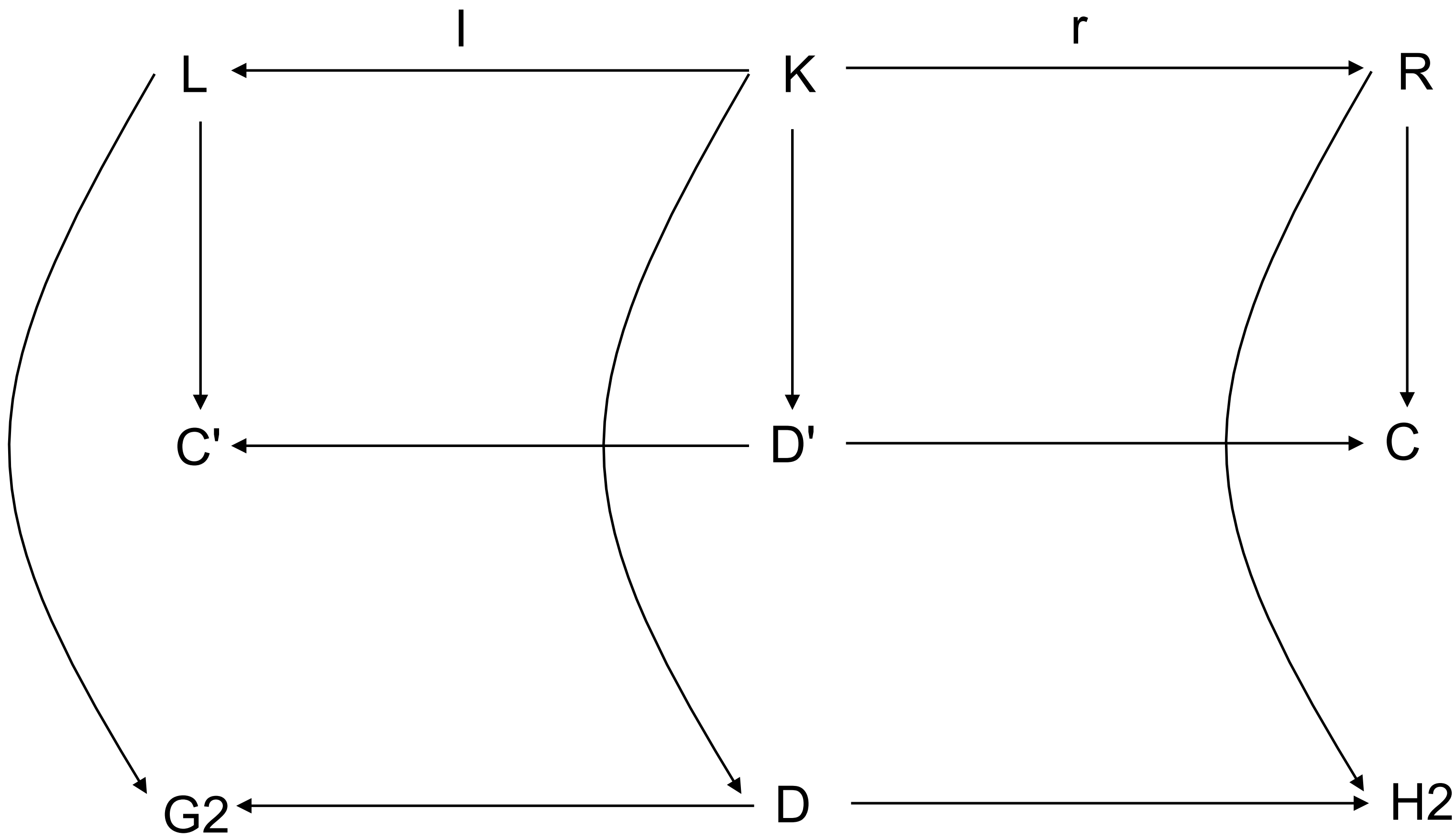
Right NACs elimination (Proof)

Construction:

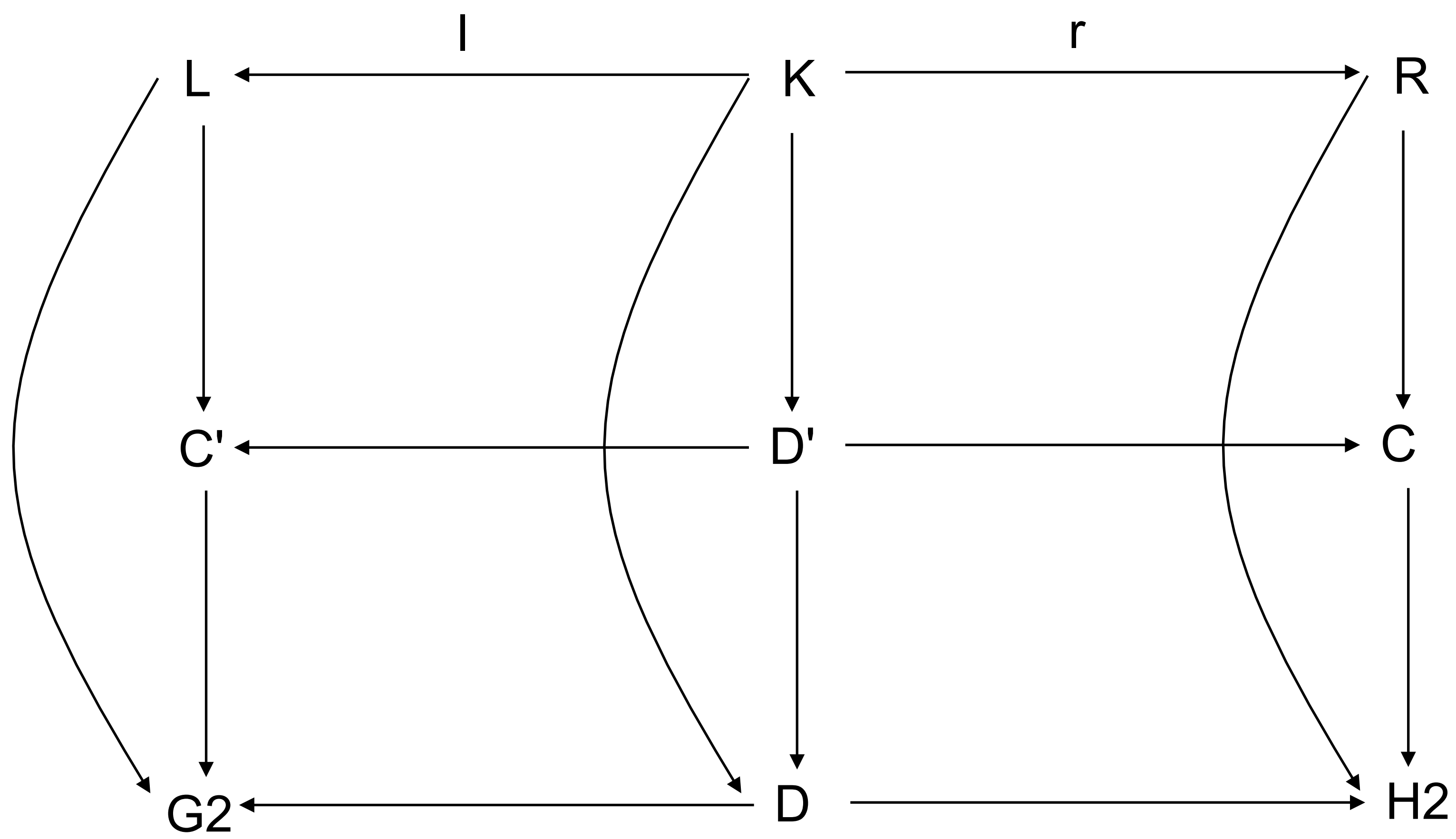


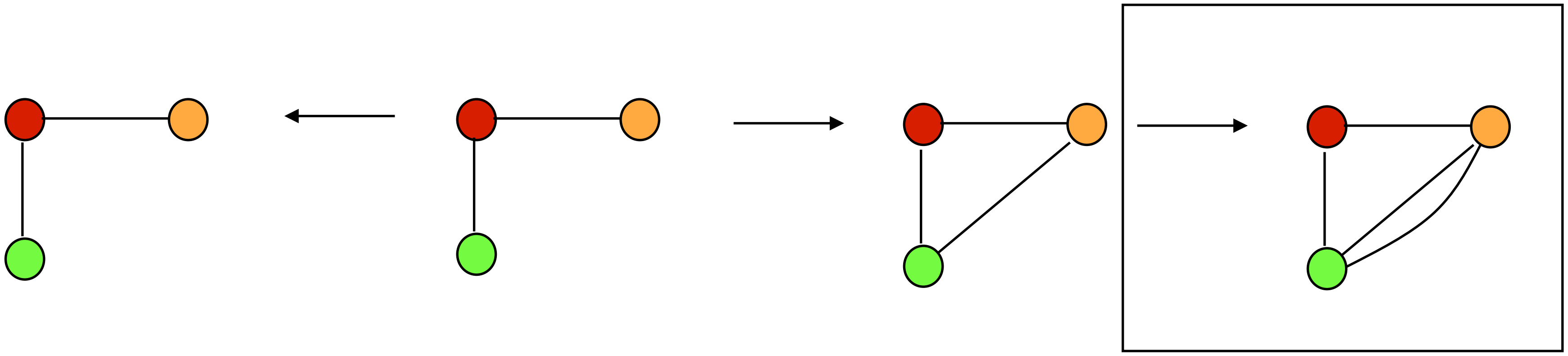
If there is a pushout complement D'

Right NACs elimination

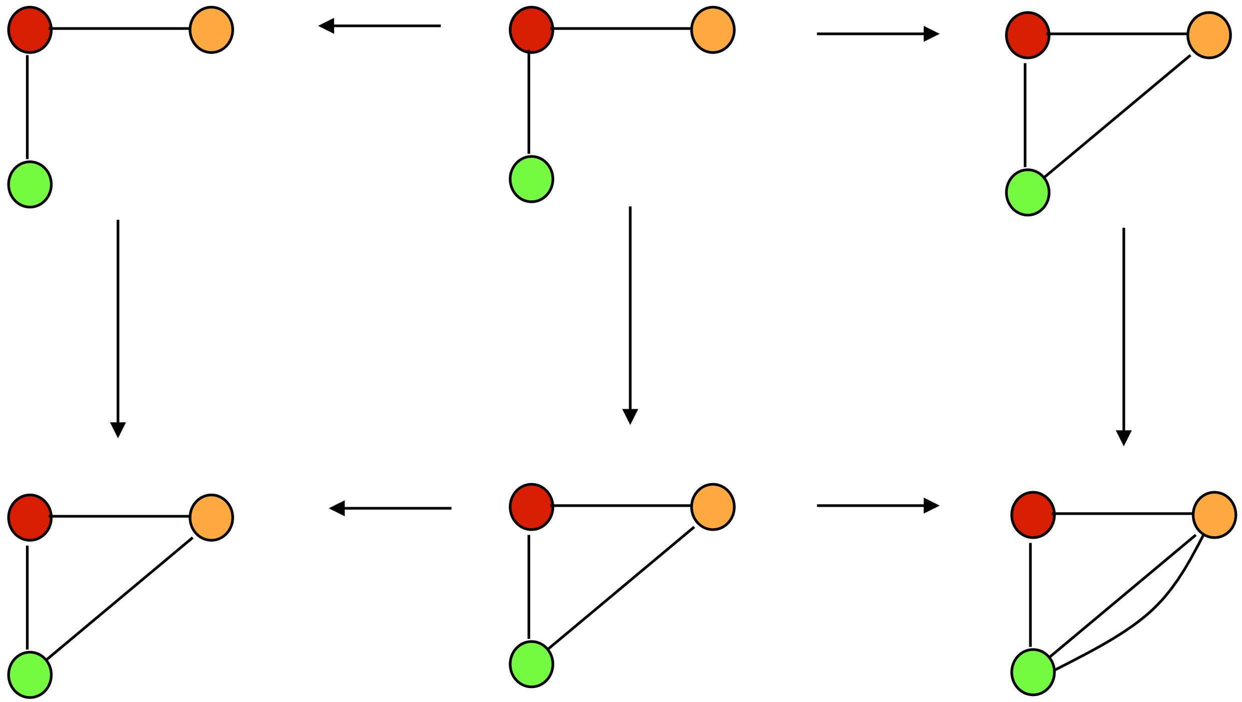


Right NACs elimination



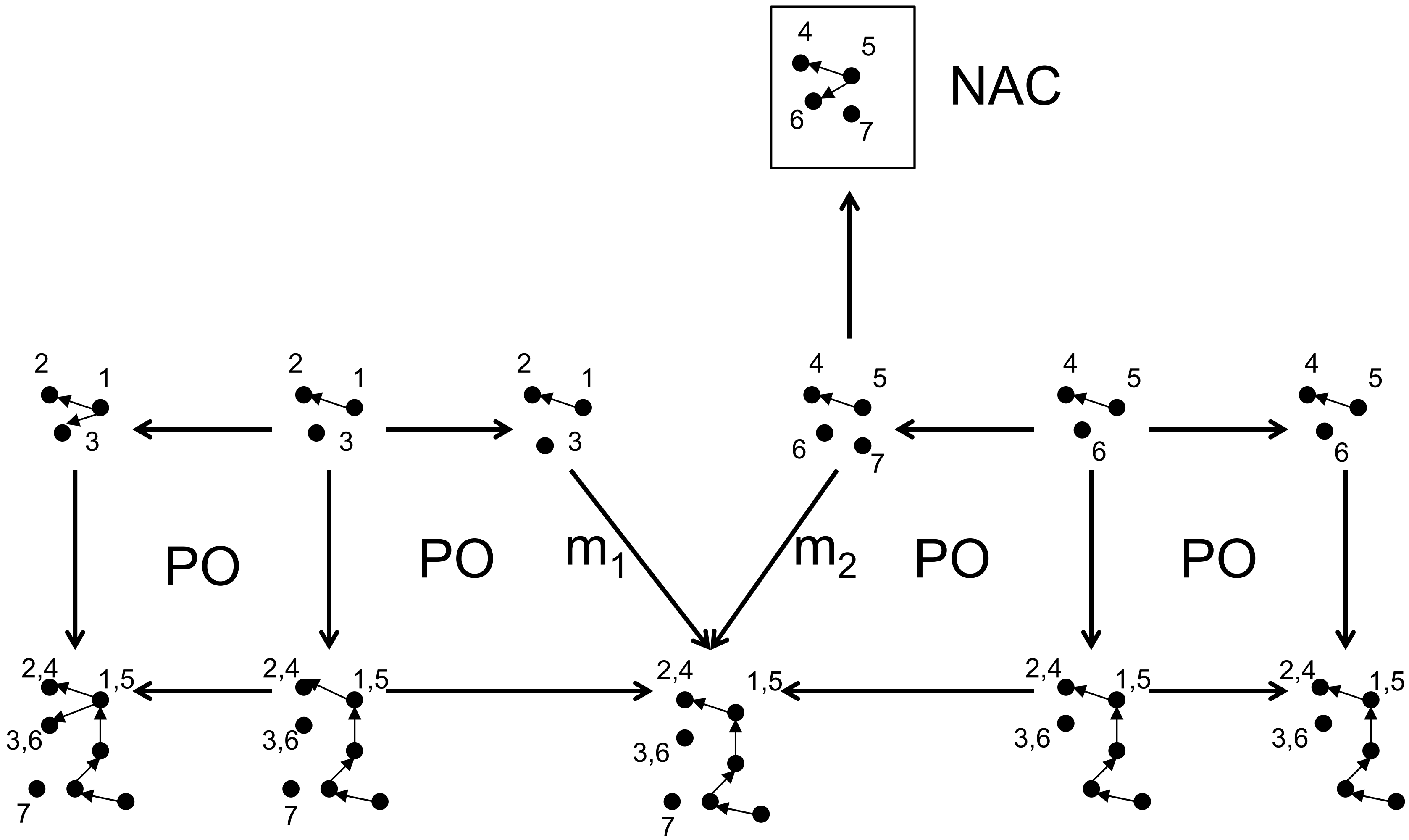


NAC



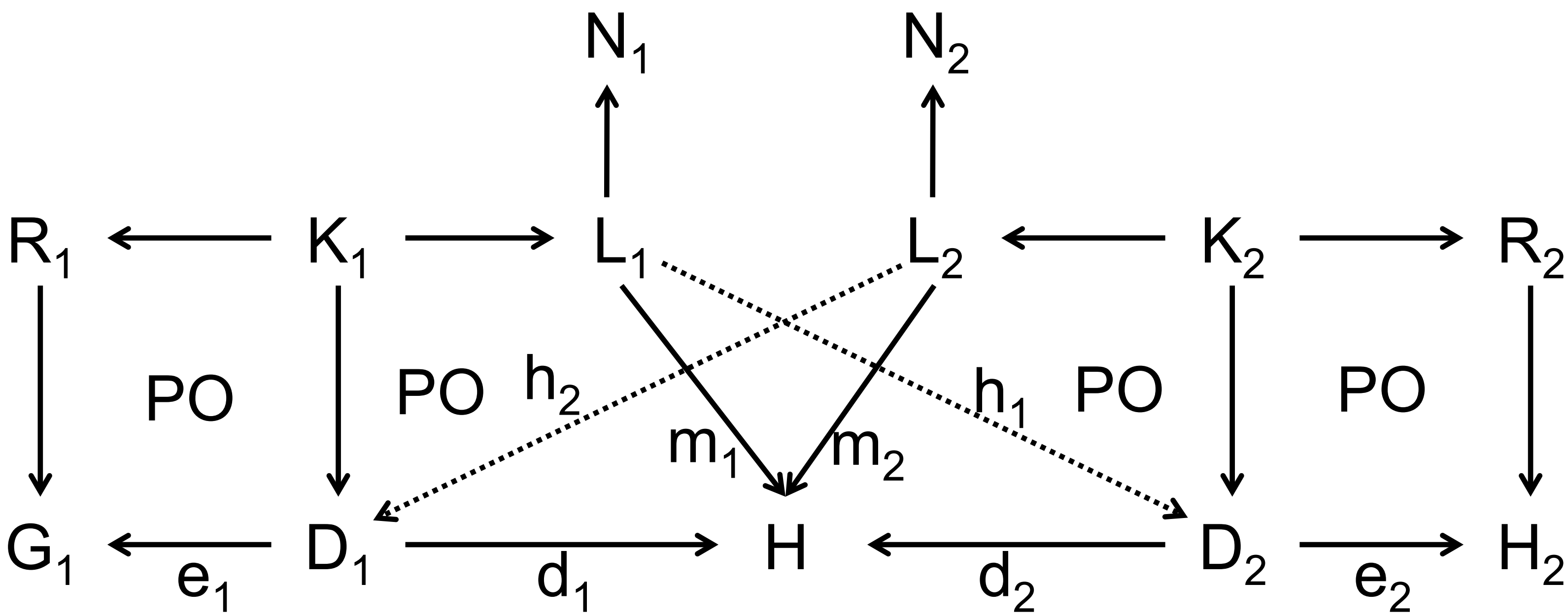
Parallel Independence and Critical Pairs for Rules with NACs

Parallel Independence



Parallel Independence with NACs

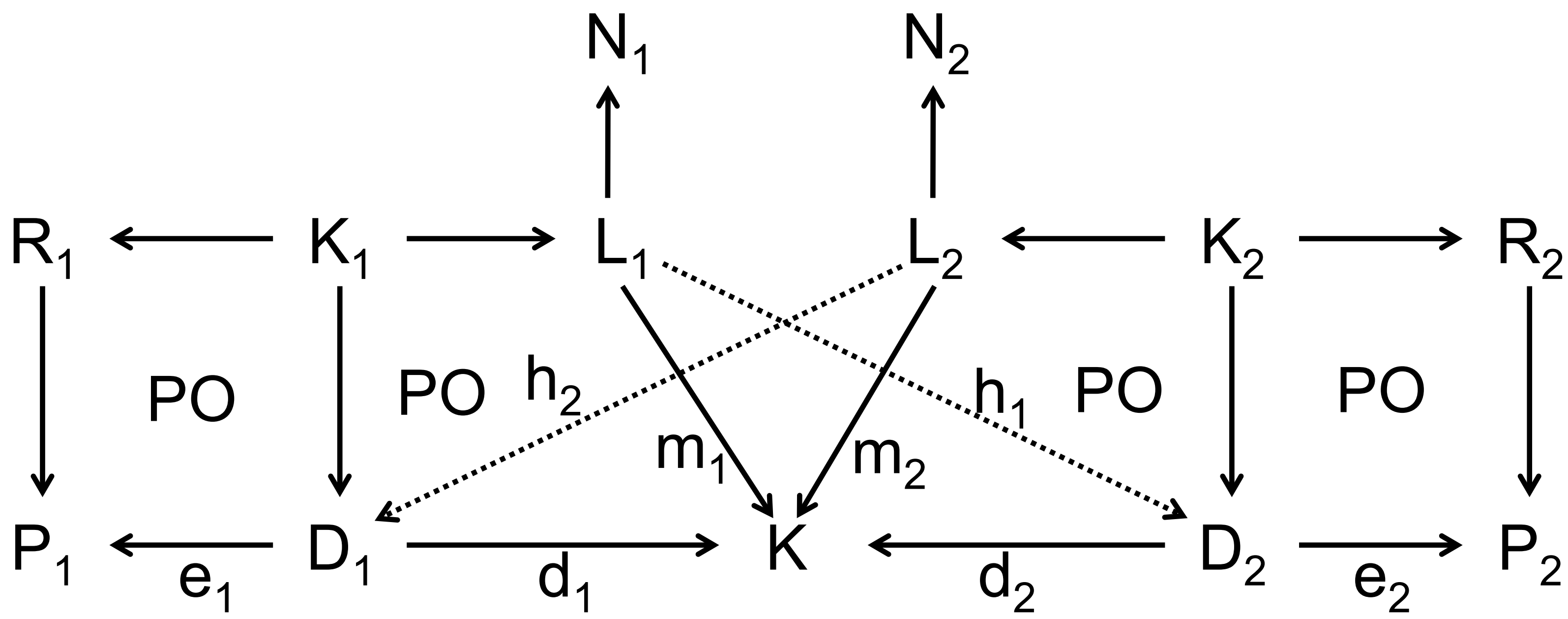
Two transformations with NACs on the same graph G are parallel independent:



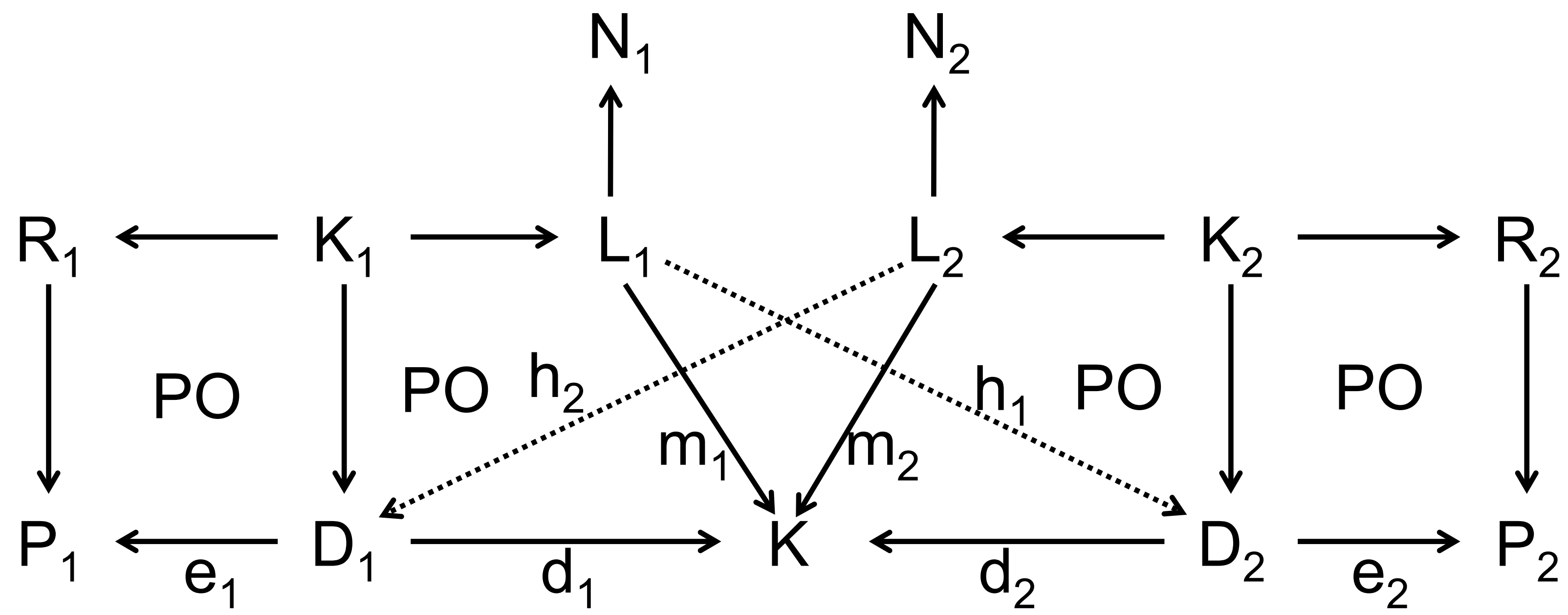
if there exist morphisms h_1 and h_2 such that $d_2 \cdot h_1 = m_1$, $d_1 \cdot h_2 = m_2$, $e_1 \cdot h_2 \models N_2$, and $e_2 \cdot h_1 \models N_1$

Critical Pairs for Rules with NACs

A critical pair consists of two transformations with NACs on the same graph K such that m_1 and m_2 are jointly surjective and one of the following conditions hold:



1. There is no morphism h_1 such that $d_2 \cdot h_1 = m_1$



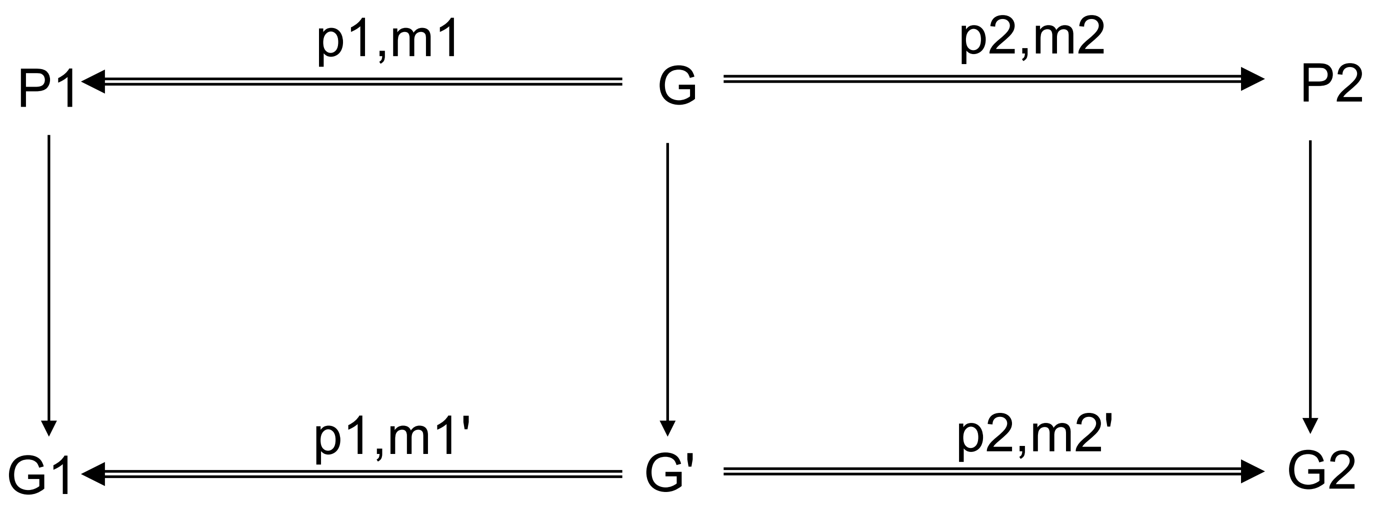
2. There is no morphism h_2 such that $d_1 \cdot h_2 = m_2$
3. There is a morphism h_1 but $e_2 \cdot h_1$ |= N_1
4. There is a morphism h_2 but $e_1 \cdot h_2$ |= N_2

Completeness Theorem

For any two parallel dependent direct transformations with NACs:

$$G1 \; {}_{p1,m1'}\Leftarrow G' \Rightarrow_{p2,m2'} G2$$

there is a critical pair $P1 \; {}_{p1,m1}\Leftarrow G \Rightarrow_{p2,m2} P2$ such that:



Local Confluence Theorem

A graph transformation system with NACs is locally confluent if all its critical pairs are strictly NAC-confluent.

Elimination of conflicts

Given the critical pair $P1 \,_{p1,m1}\!\Leftarrow K \Rightarrow_{p2,m2} P2$, if we know that, in the situation where the conflict happens, the right rule to be applied is p1, then the conflict can be eliminated by adding to p2 the *filter NAC* $L2 \rightarrow K$.

Generalization

The previous results have been generalized to arbitrary M-adhesive categories where, in addition, NACs are arbitrary conditions written in the Logic of Nested Conditions (A. Habel and K.H Penneman (2009)), as expressive as First-Order Logic.

Graph Constraints

Two kinds of constraints:

Basic Constraints

- C

Conditional Constraints

- $(c: X \rightarrow C)$

Two kinds of constraints:

Basic Constraints

– C

Conditional Constraints

– (c: $X \rightarrow C$)

Basic constraints are a special case of conditional constraints (when $X = \emptyset$)

Two kinds of constraints:

plus ¬, ∧, ∨

Basic Constraints

- C

Conditional Constraints

- (c: X → C)

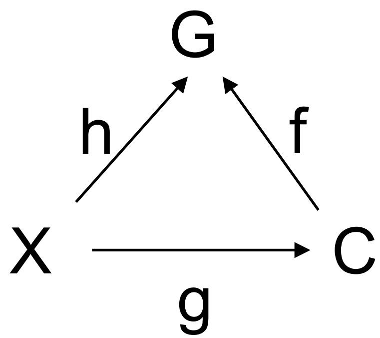
Basic constraints are a special case of conditional constraints (when $X = \emptyset$)

Satisfaction of basic constraints:

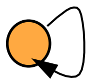
$G \models C$ if $h: C \rightarrow G$

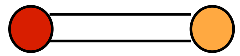
Satisfaction of conditional constraints:

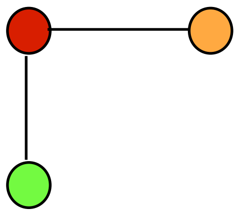
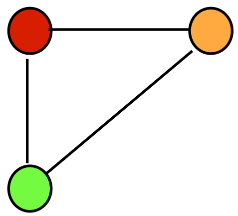
$G \models (g: X \rightarrow C)$ if for every $h: C \rightarrow G$, there is an $f: C \rightarrow G$ such that:



Examples

¬ 

¬ 

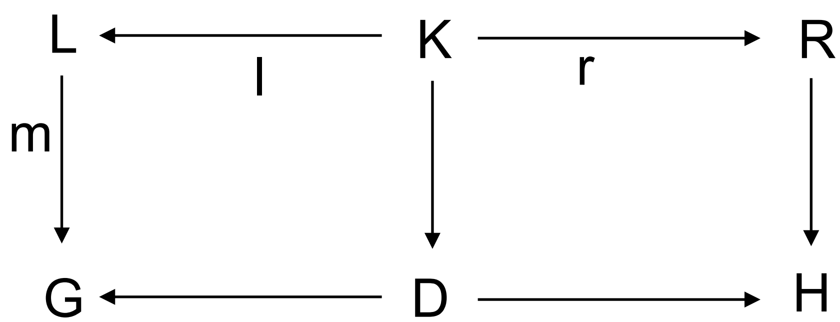
If  then 

Graph Transformation with constraints

Given a set of constraints C, and given

$$p = L \xleftarrow{l} K \xrightarrow{r} R$$

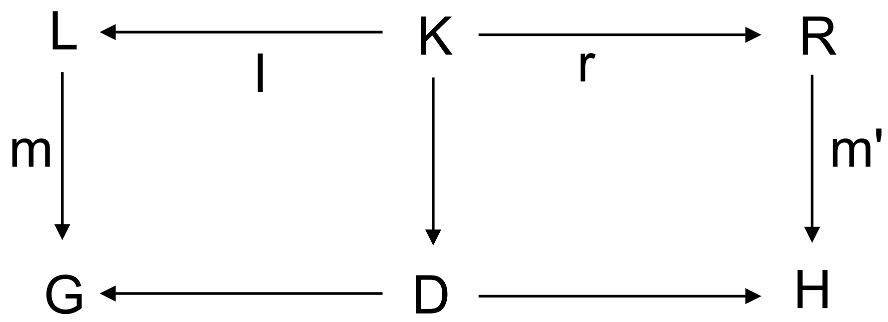
we can apply p to G if



H |= C.

Transformation of constraints into NACs

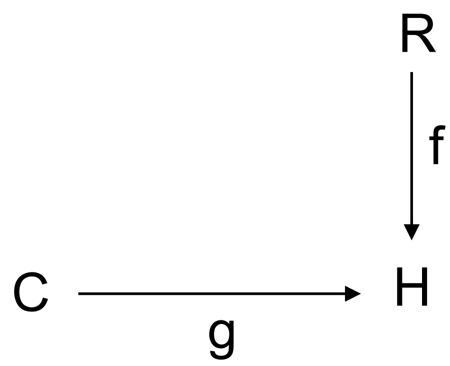
Given a constraint c, there is a set of right NACs Φ such that given:



m' satisfies all the NACs in Φ if and only if H satisfies c.

Construction

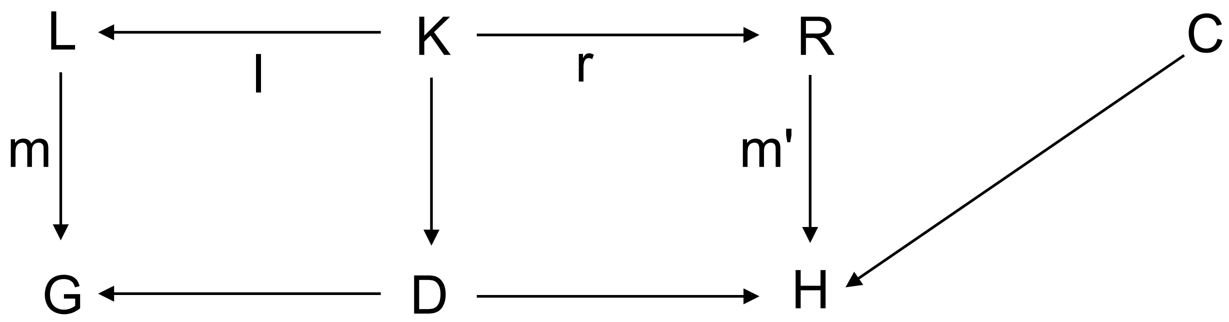
If c = ¬ C, then Φ is the set of right NACs f: R $\rightarrow$ H:



such that f and g are jointly surjective.

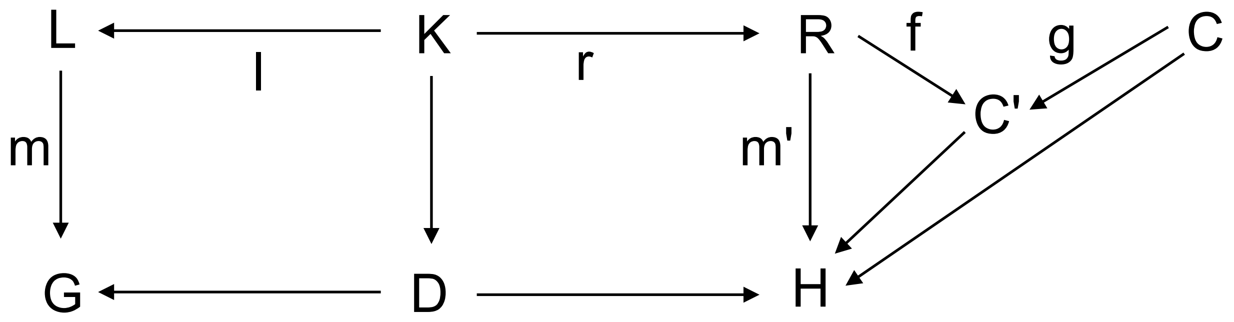
Proof

Let us assume that

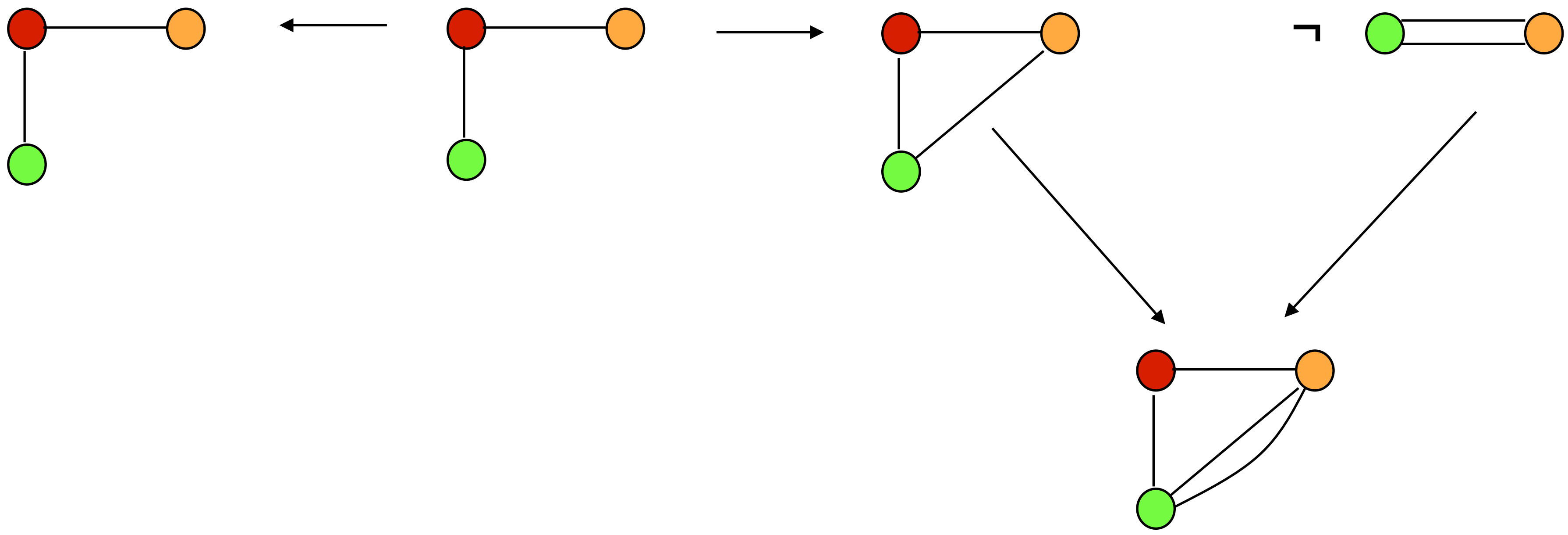


Proof

By pair factorization there are C', f, and g such that f and g are jointly surjective



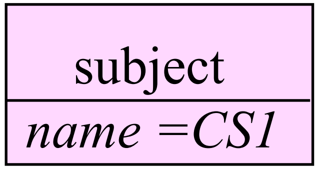
Example

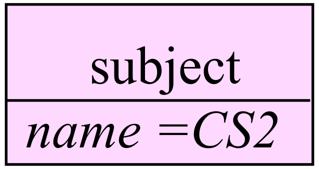


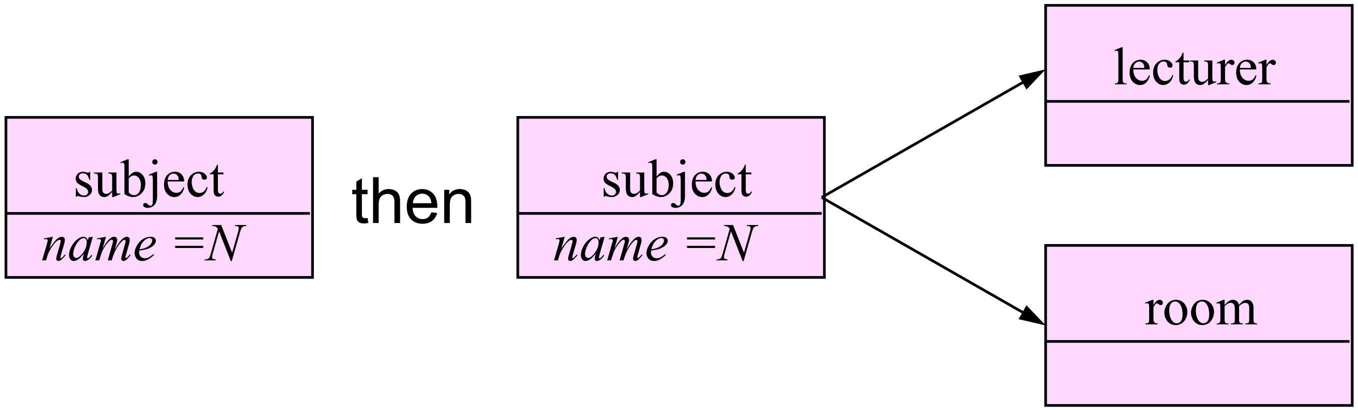
Modelling and Reasoning with Constraints

Modelling with Constraints

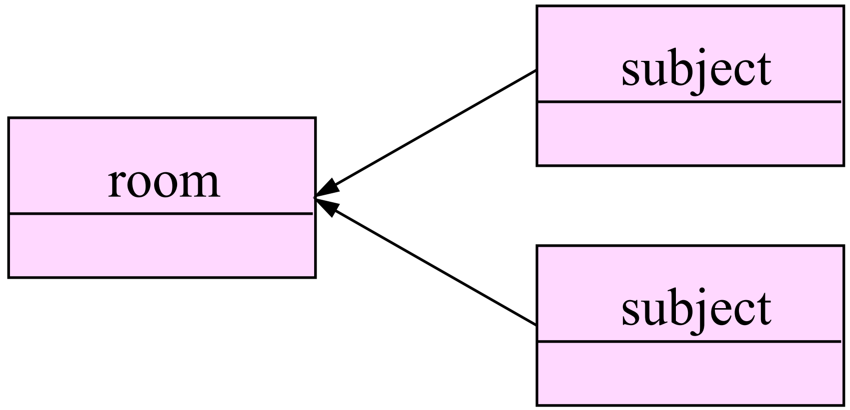
Example

(1) ∃ 

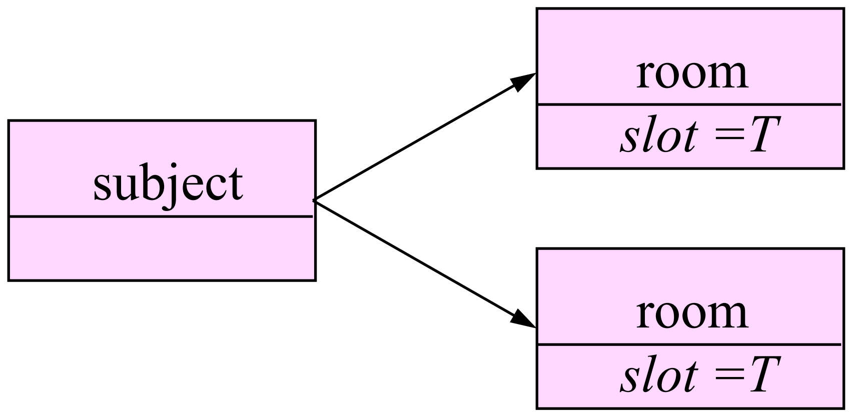
(2) ∃ 

(3) If 

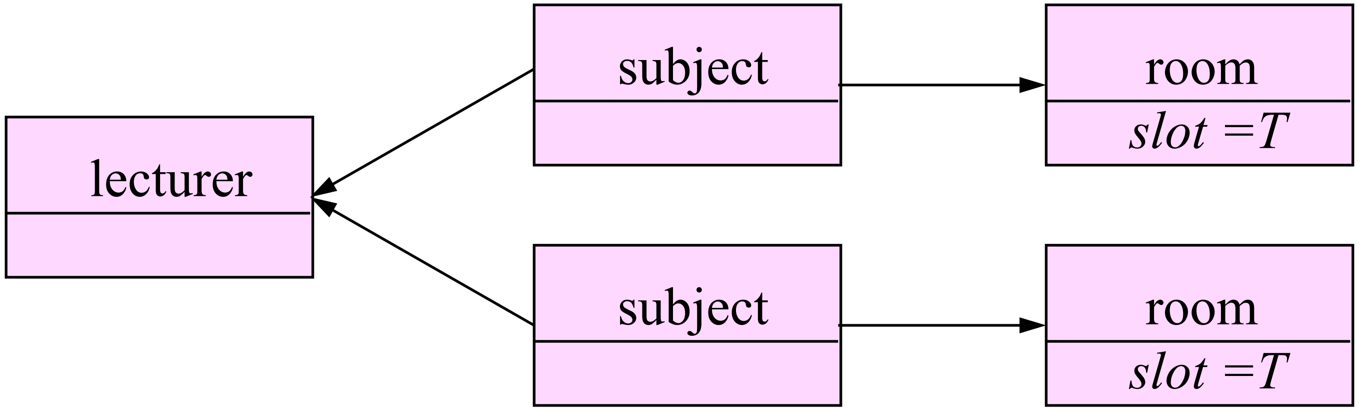
(4) $\neg\exists$



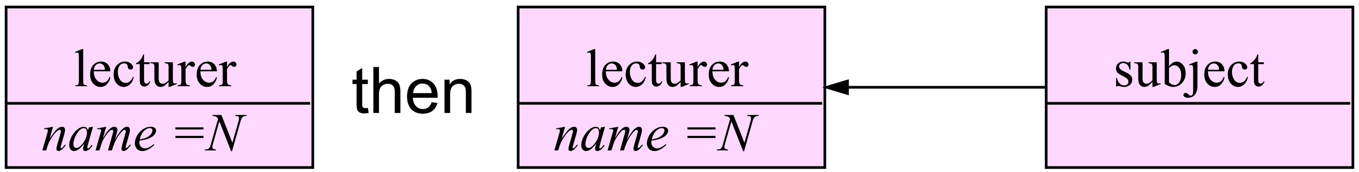
(5) $\neg\exists$



(6) ¬∃


lecturer
subject
room
slot =T
subject
room
slot =T

(7) ¬If


lecturer
name =N
then
lecturer
name =N
subject

Reasoning with constraints

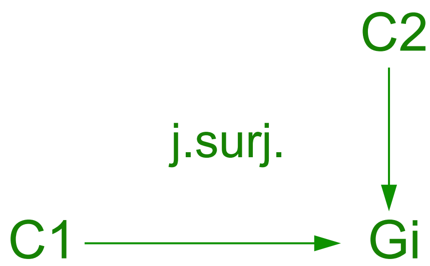
1)

$$\frac{\exists C1 \vee \Gamma 1 \qquad \neg\exists C2}{\Gamma 1}$$

if there is a morphism C2 → C1

2)

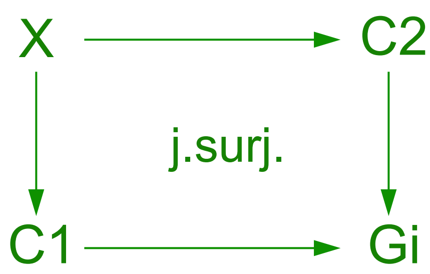
$$\frac{\exists C1 \vee \Gamma 1 \qquad \exists C2}{\exists G1 \vee \ldots \vee \exists Gk \vee \Gamma 1}$$



3)

$$\frac{\exists C1 \vee \Gamma \qquad \forall(g':X \to C2)}{\exists G1 \vee \ldots \vee \exists Gk \vee \Gamma}$$

If X → C1 and:

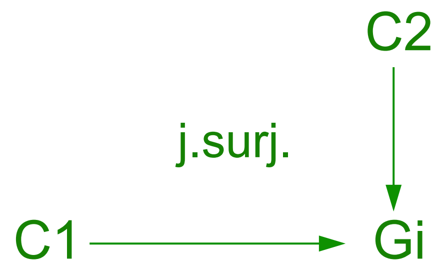


Example

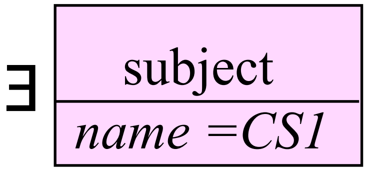
(1) ∃ subject | *name = CS1*

(2) ∃ subject | *name = CS2*

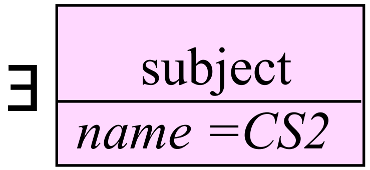
$$\frac{\exists C1 \vee \Gamma 1 \qquad \exists C2}{\exists G1 \vee \ldots \vee \exists Gk \vee \Gamma 1}$$



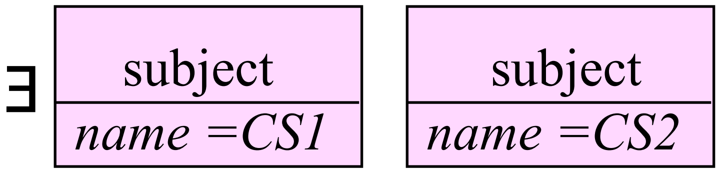
(1)


∃
subject
name =CS1

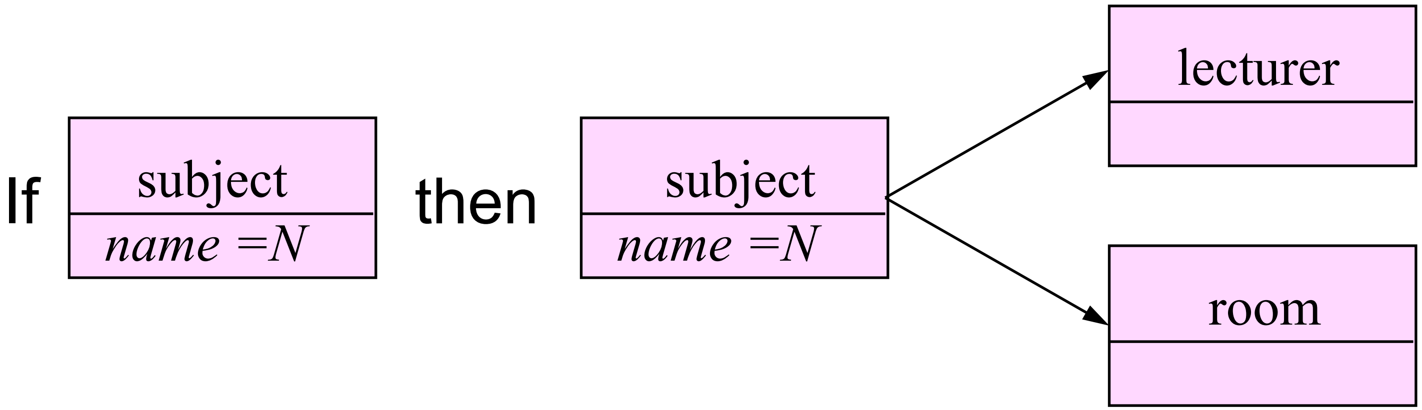
(2)


∃
subject
name =CS2

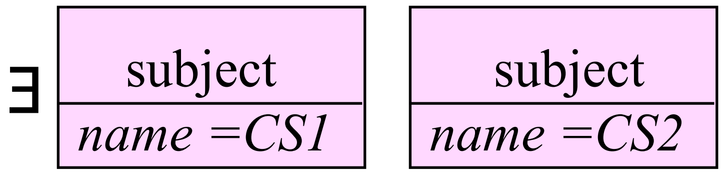
(8)


∃
subject
name =CS1
subject
name =CS2

(3)

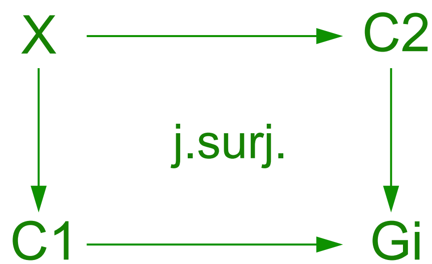


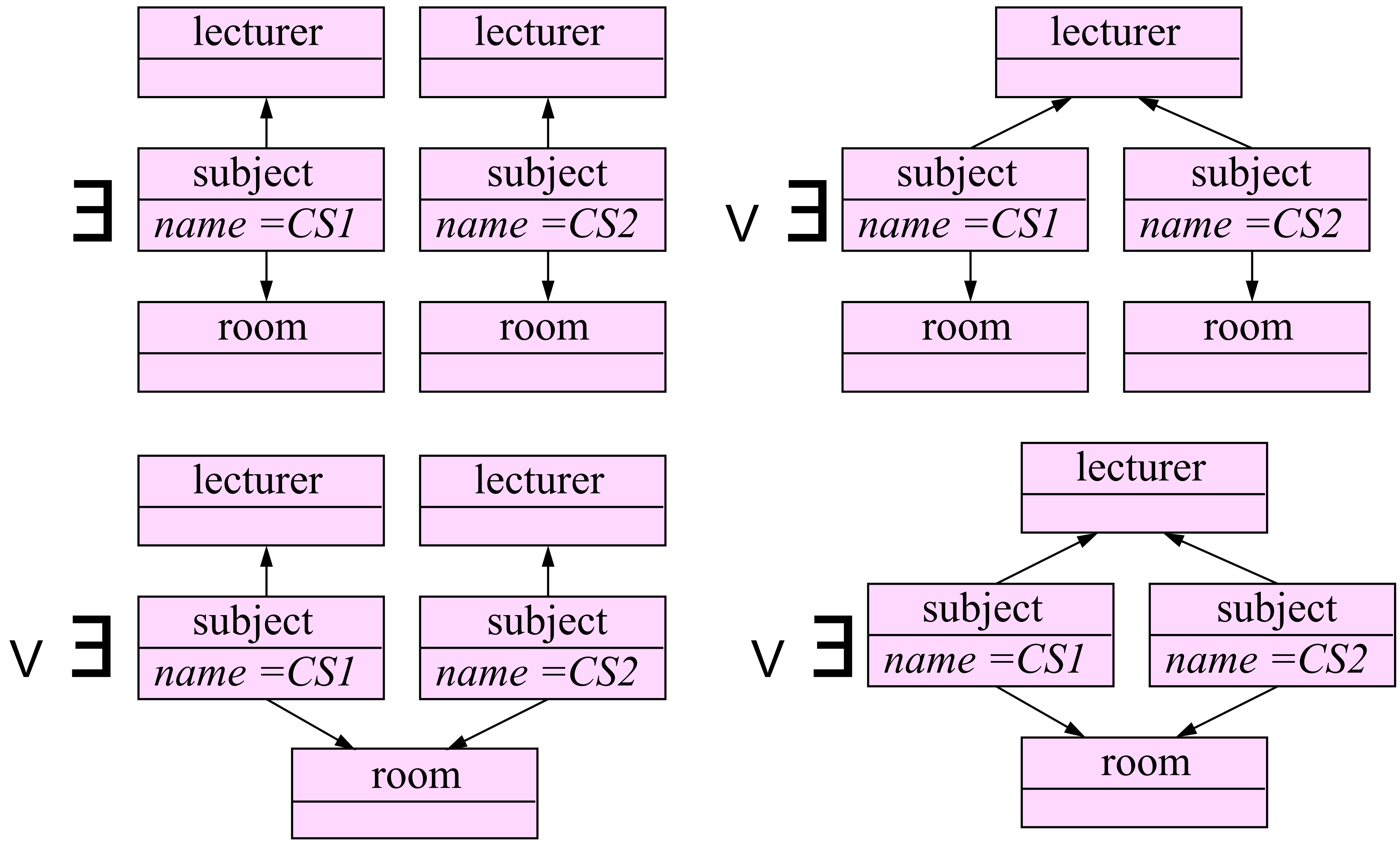
(8)

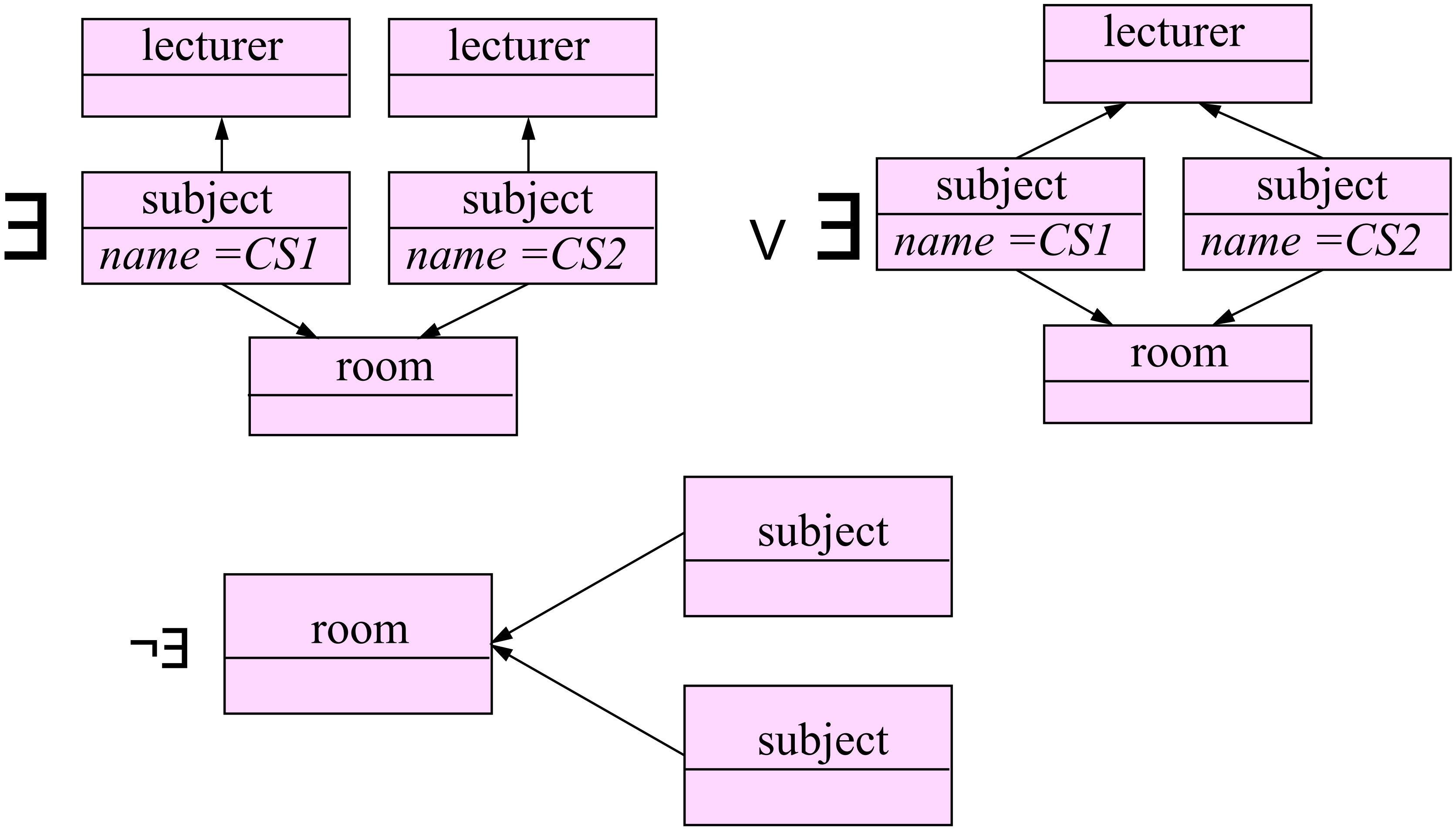


$$\frac{\exists C1 \vee \Gamma \qquad \forall(g':X \to C2)}{\exists G1 \vee \dots \vee \exists Gk \vee \Gamma}$$

If X → C1 and:

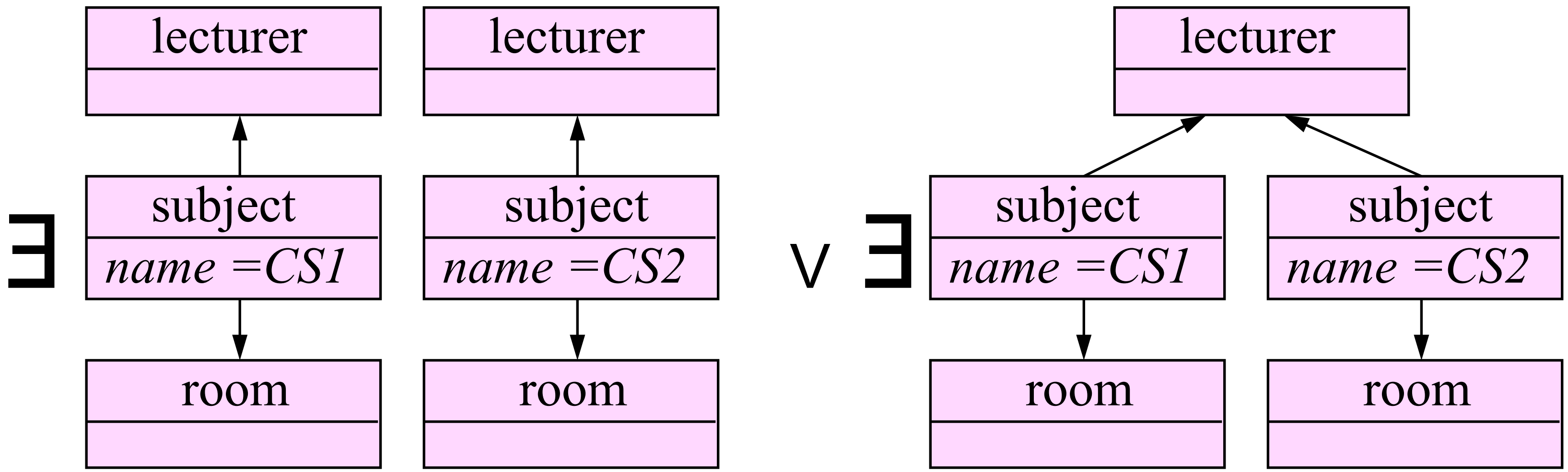



∃
lecturer
subject
name =CS1
room
lecturer
subject
name =CS2
room
∨ ∃
lecturer
subject
name =CS1
room
subject
name =CS2
room
∨ ∃
lecturer
subject
name =CS1
lecturer
subject
name =CS2
room
∨ ∃
lecturer
subject
name =CS1
subject
name =CS2
room



$$\frac{\exists C1 \vee \Gamma 1 \qquad \neg\exists C2}{\Gamma 1}$$

if there is a morphism C2 → C1



∃
lecturer
subject
name =CS1
room
lecturer
subject
name =CS2
room
∨ ∃
lecturer
subject
name =CS1
room
subject
name =CS2
room

¡Thank You!