Introduction to Algebraic Graph Transformation

Fernando Orejas

Royal Holloway University of London

on leave from Universitat Politècnica de Catalunya, Barcelona

- Lecture 1: Basic Notions in Graph Transformation
- Lecture 2: Fundamental Theory of GT
- Lecture 3: Conditional Graph Transformation
- Lecture 4: Model Transformation by Triple Graph Grammars

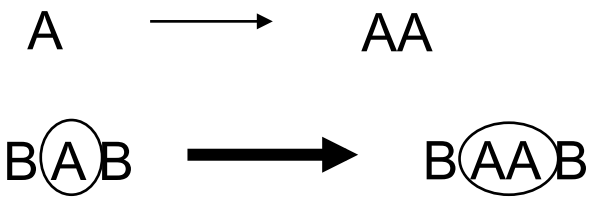
1. Introduction
2. The basic case
3. Generalizing Graph Transformation
4. Variations
5. Extensions

Introduction

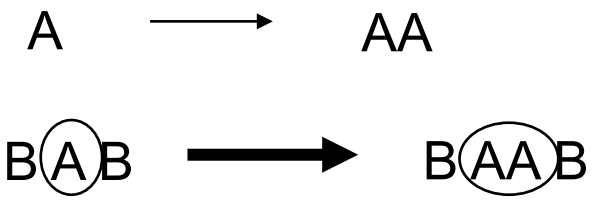
Origins of Graph Transformation (1969):

Generalization of Chomsky Grammars

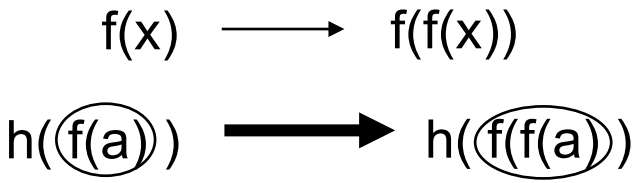
Chomsky Grammars:



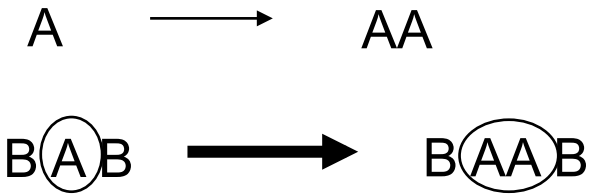
Chomsky Grammars:



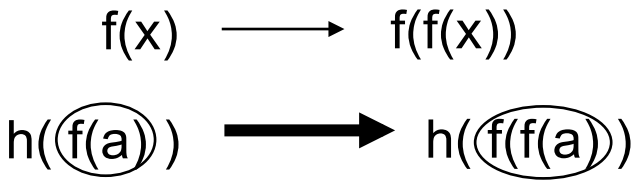
Term Rewriting:



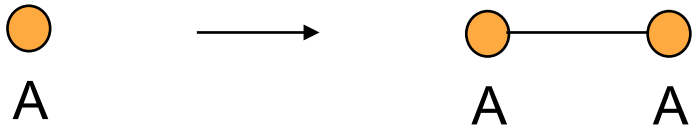
Chomsky Grammars:

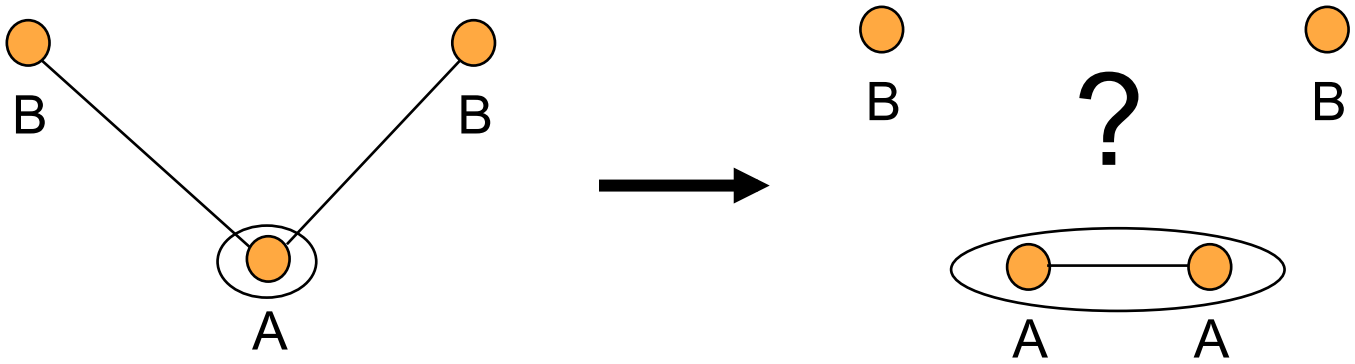


Term Rewriting:



Graph Grammars





- What kind of graphs?
- What kind of transformations
- How do we describe transformations

- What kind of graphs?

  Any kind (and more)

- What kind of transformations

  Replacing subgraphs by subgraphs

- How do we describe transformations

  Algebraic double pushout approach

The basic case

A graph G1 consists of:

- V: nodes
- E: edges
- source: $E \rightarrow V$
- target: $E \rightarrow V$

A graph morphism h: $G_1 \rightarrow G_2$ consists:

- h_V: $V_1 \rightarrow V_2$
- h_E : $E_1 \rightarrow E_2$

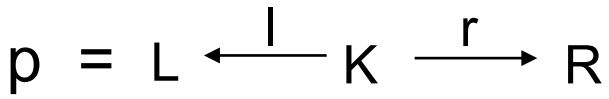
such that

1. $h_V(source_1(e_1)) = source_2(h_E\ (e_1))$
2. $h_V(target_1(e_1)) = target_2(h_E\ (e_1))$

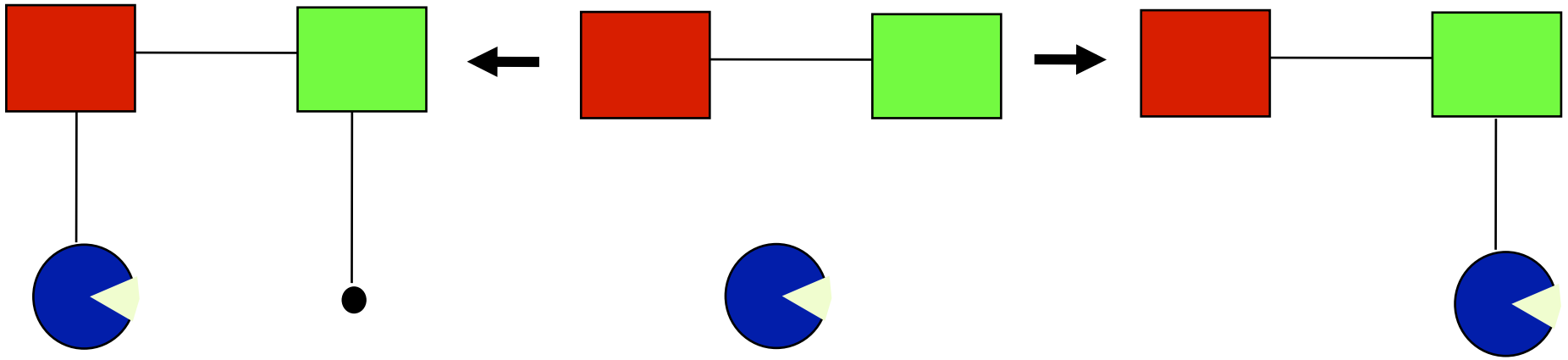
- What is a graph transformation rule?
- How do we apply a rule?

Double Pushout Graph Transformation Rules

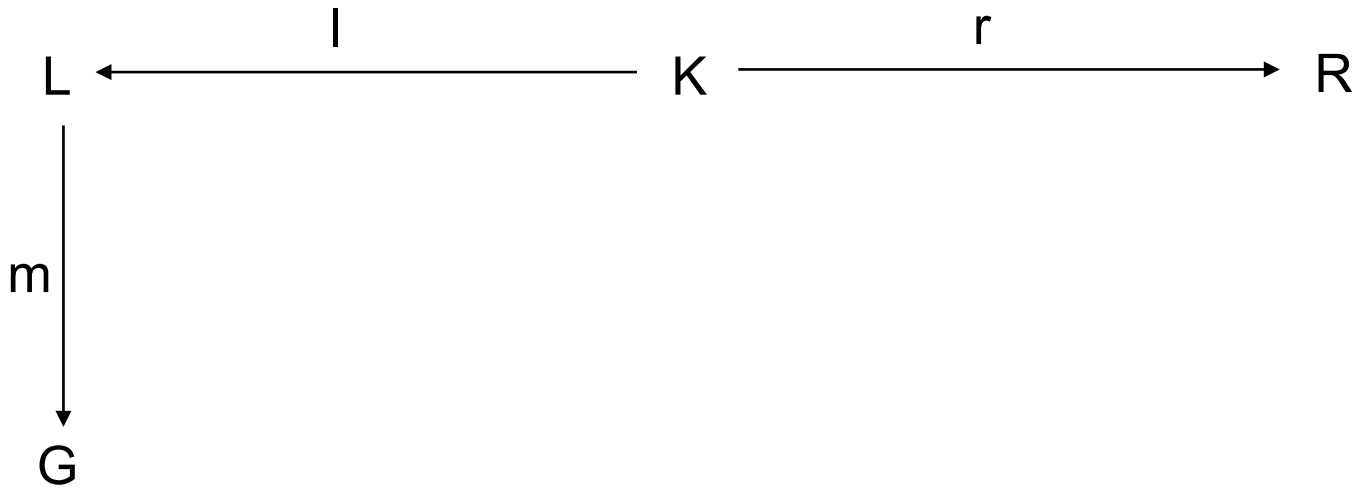
Rules



A graph transformation rule:

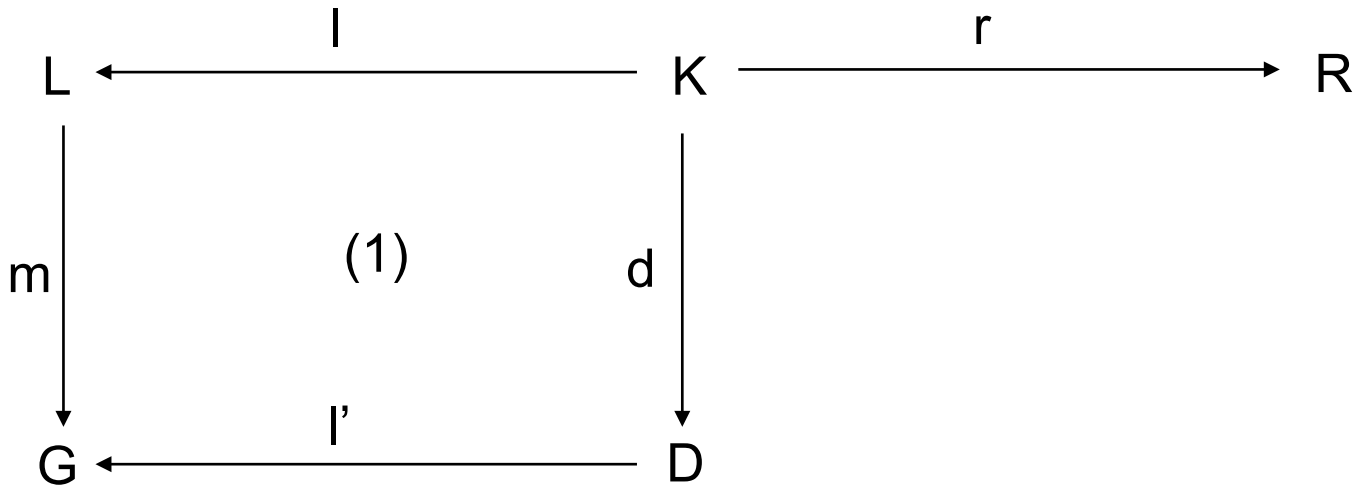


Rule Application



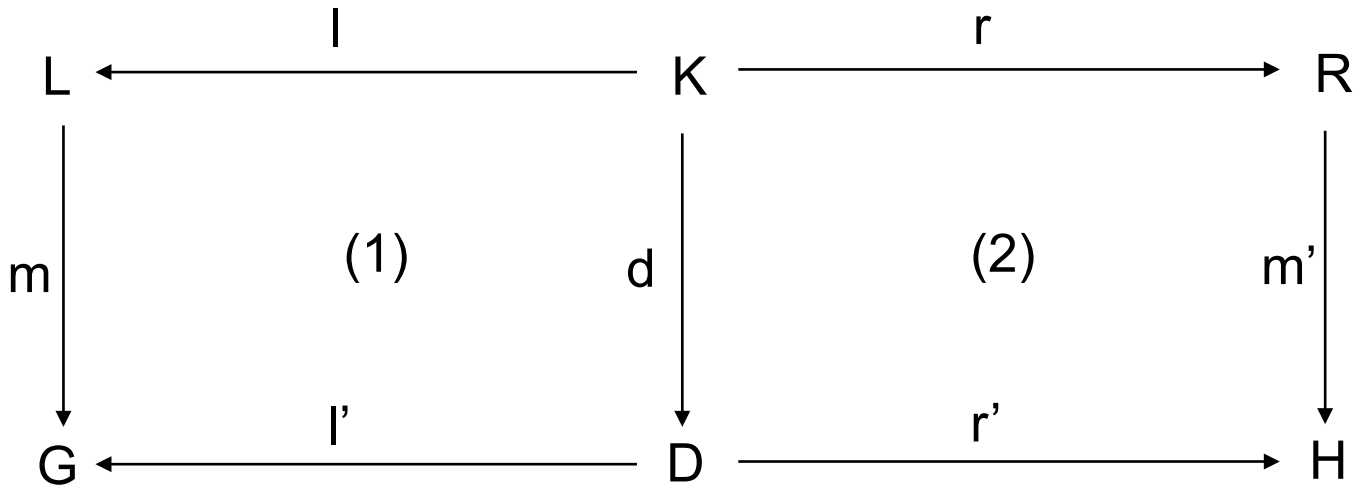
Rule Application

1. D = G \ m(L\K) is a *pushout complement*

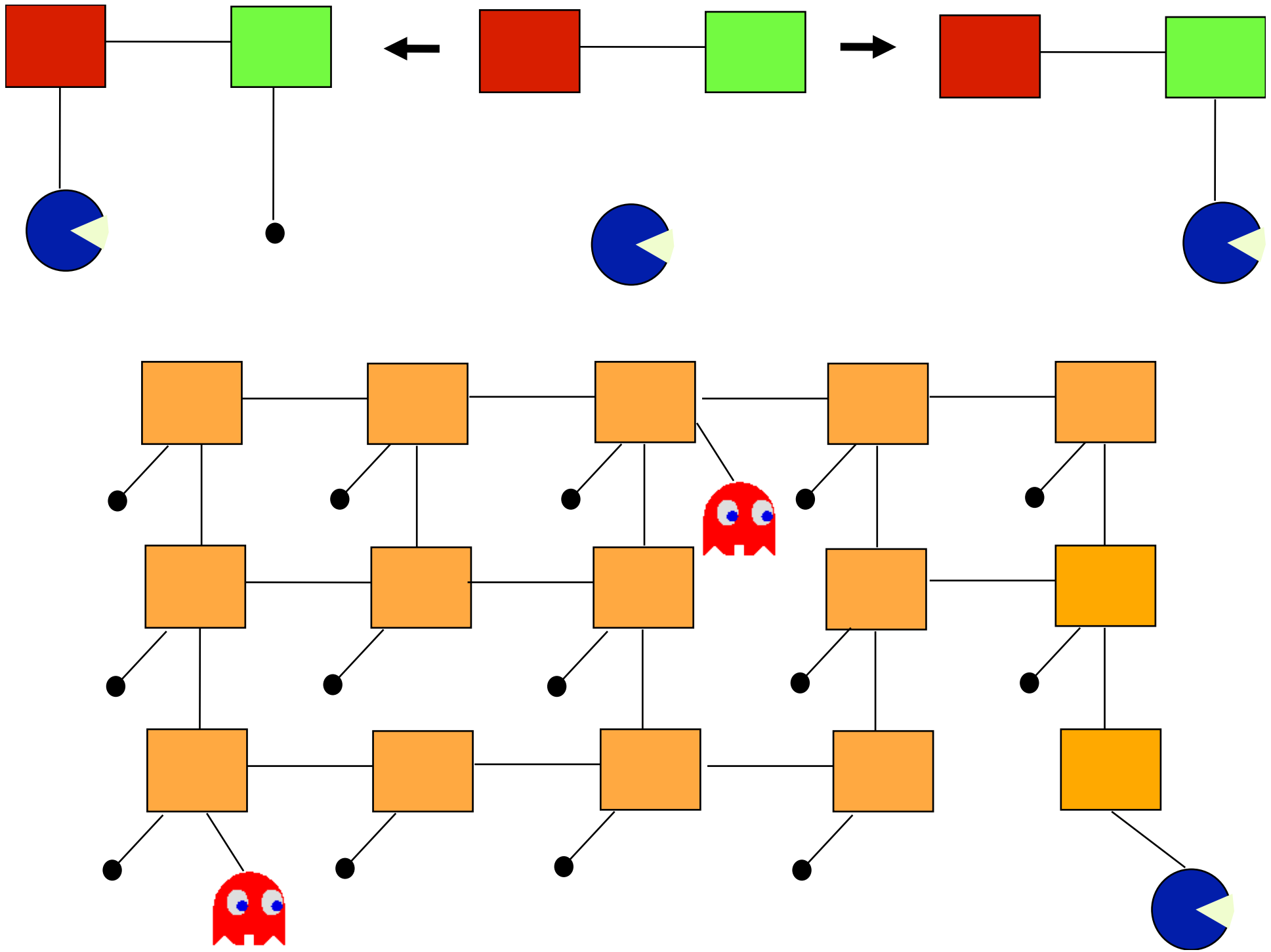


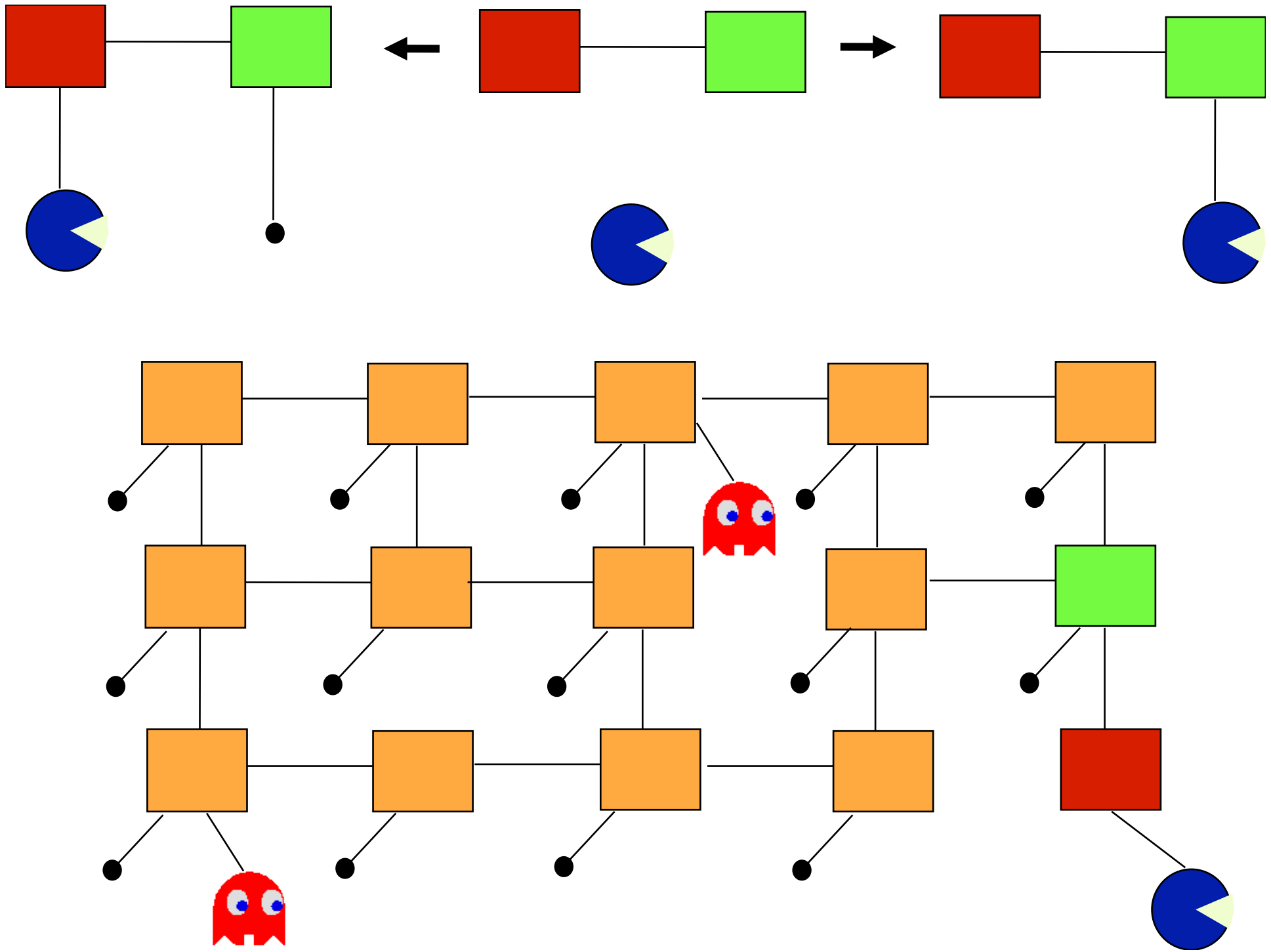
Rule Application

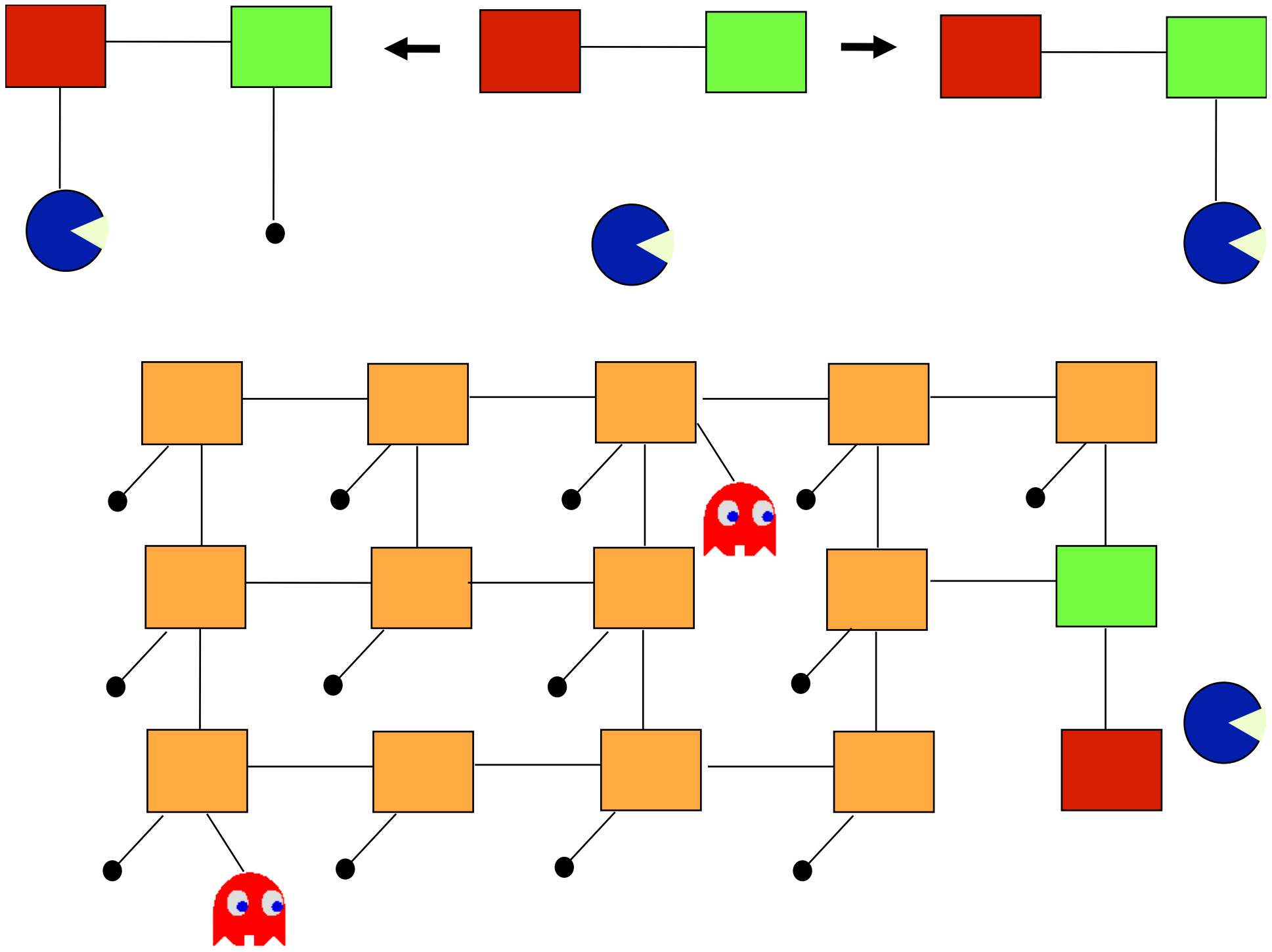
2. The result is H = D + (R\K)

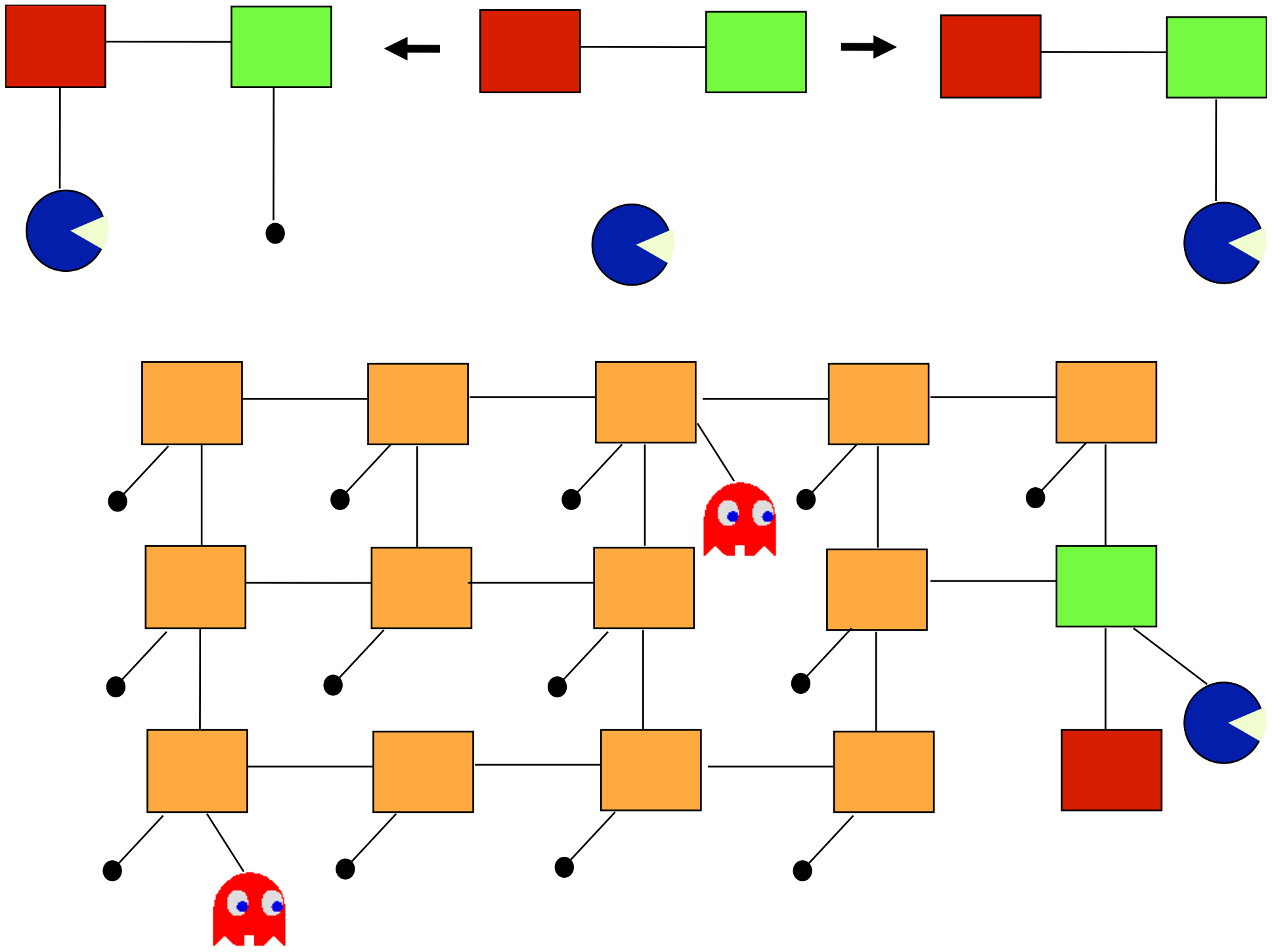


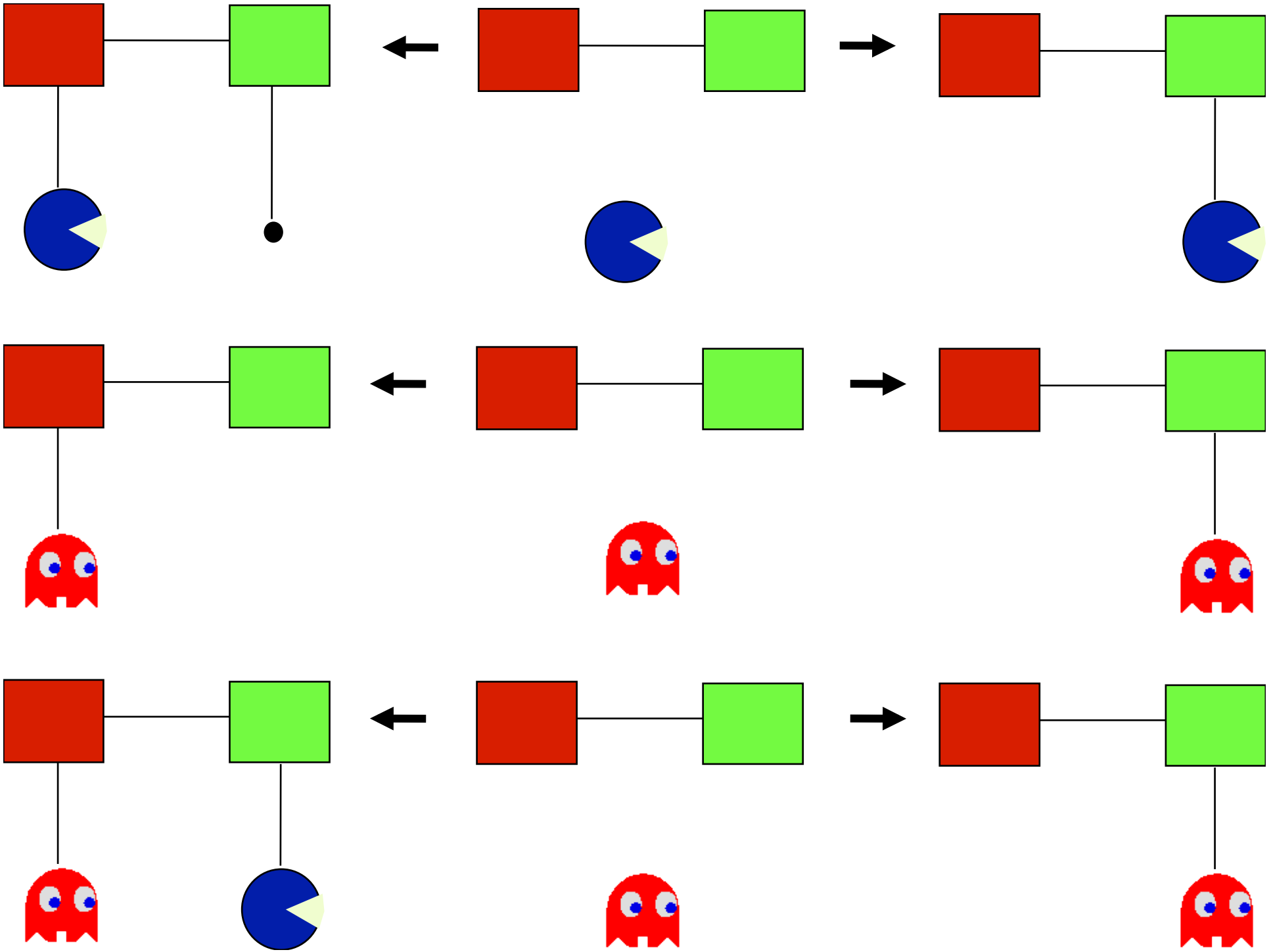
(1) and (2) are pushouts



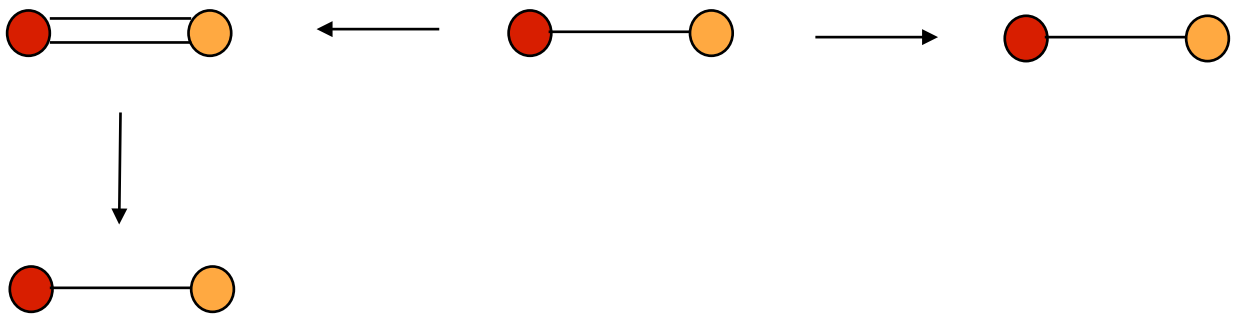




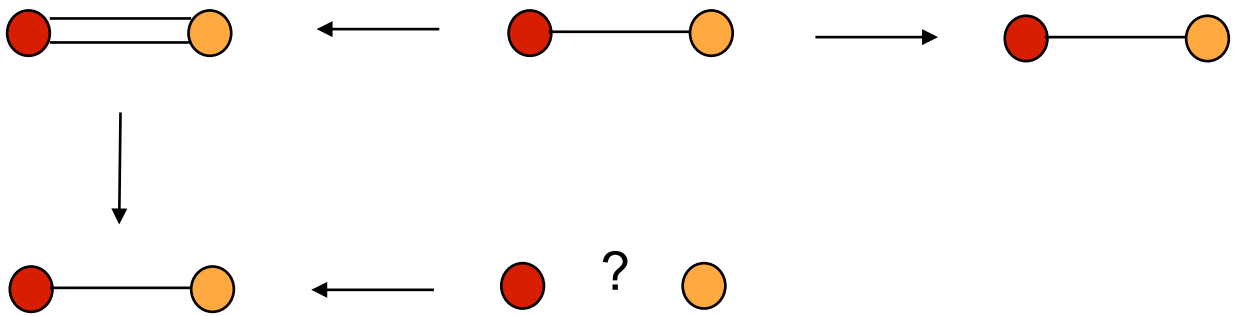




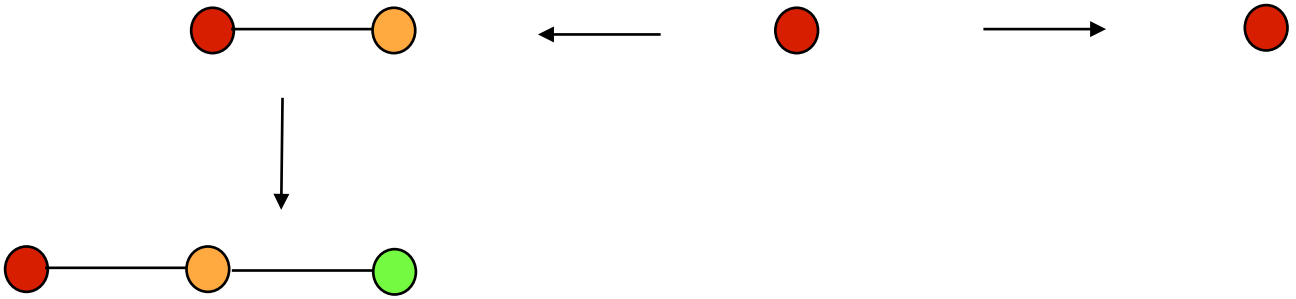
Gluing Conditions



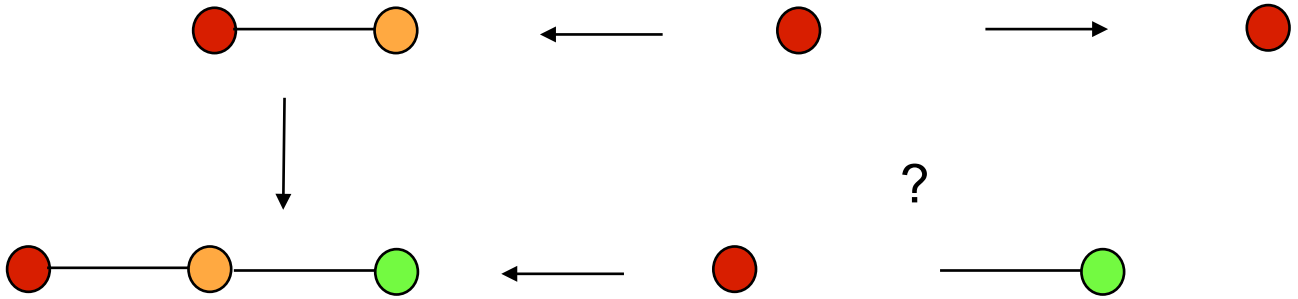
Gluing Conditions



Gluing Conditions



Gluing Conditions



- Identification Condition

For every x_1, x_2 in $V_L \cup E_L$, if $m(x_1) = m(x_2)$ then:

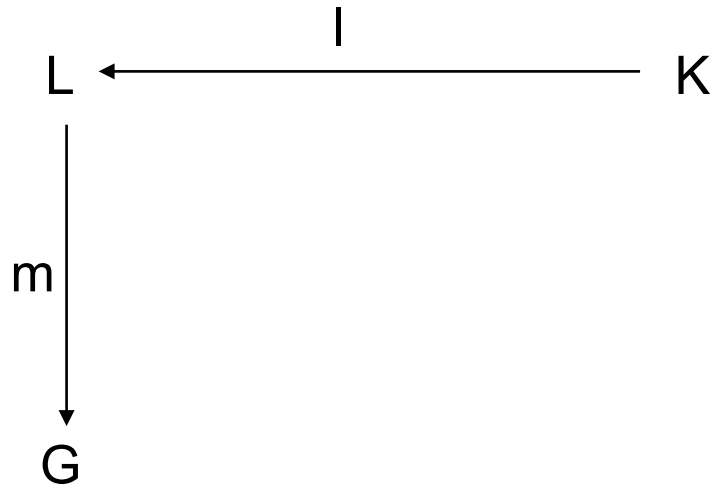
$$x_1 \in K \text{ iff } x_2 \in K.$$

- Dangling Edge Condition

For every e in $E_G \backslash m(E_L)$ if source(e) or target(e) have a pre-image in V_L then they also have it in K_L

Theorem:

Given



There is a pushout complement if and only if the identification and the dangling edge conditions hold. Moreover, this pushout complement is unique.

Generalizing Graph Transformation

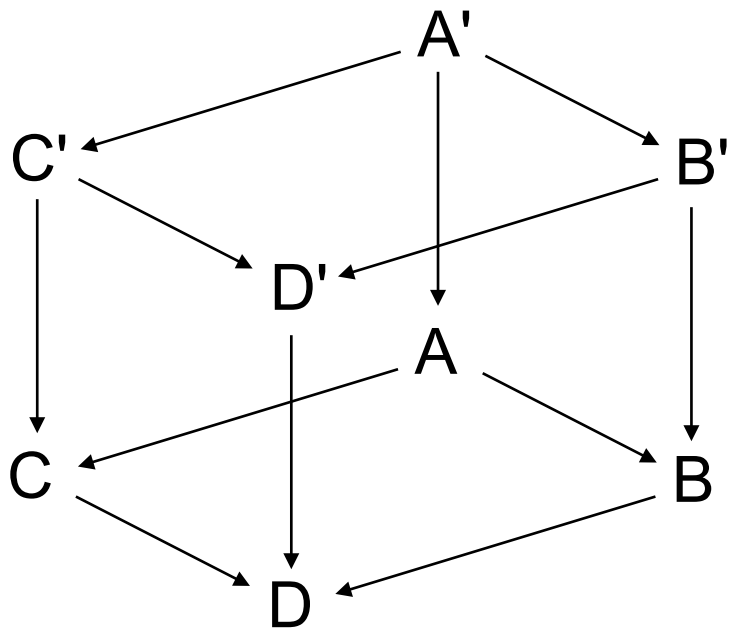
Adhesive Categories (Lack and Sobocinski)

A category **C** is adhesive if:

1. **C** has pushouts over monomorphisms. Moreover monomorphisms are closed by pullbacks and pushouts.
2. **C** has pullbacks
3. Pushouts over monomorphisms are Van Kampen squares.

Adhesive Categories (Lack and Sobocinski)

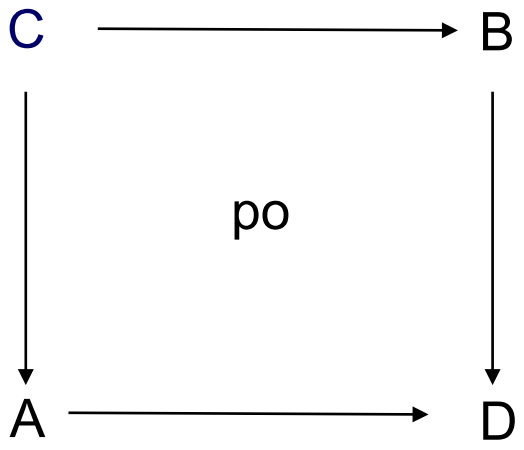
A pushout is a Van Kampen Square if for every commutative cube, where we have the pushout in the bottom square and the back squares are pullbacks, we have that the top square is a pushout if and only if the front squares are pullbacks:

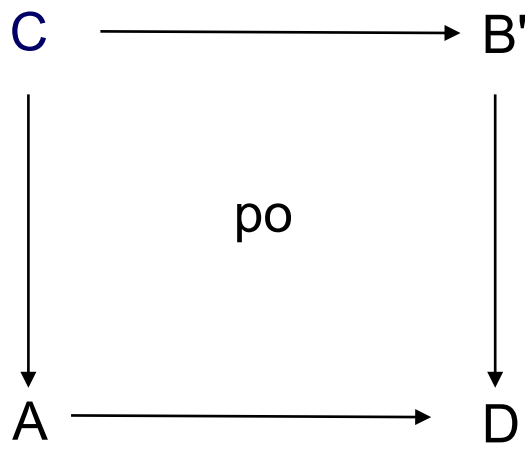


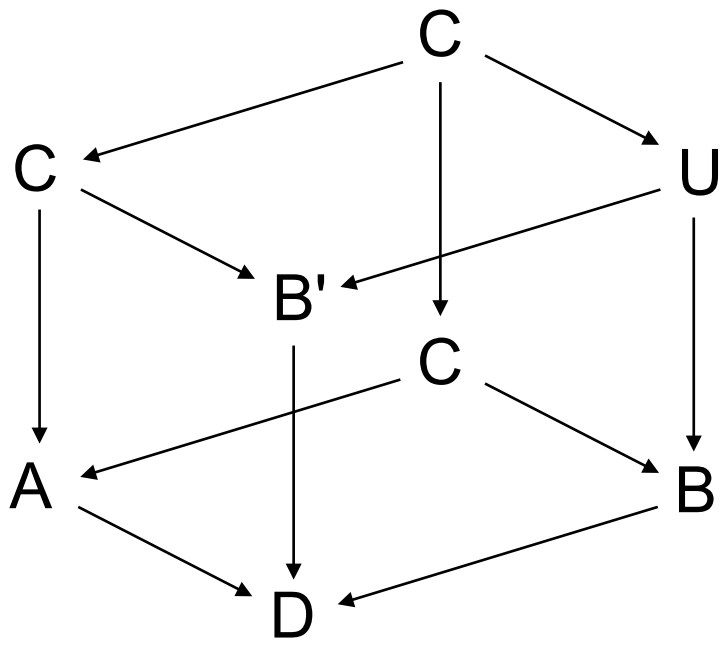
Properties of Adhesive Categories

- Complements of pushout along monos are unique
- Pushouts along monos are pullbacks
- If **C1** and **C2** are adhesive so is **C1xC2**
- If **C** is adhesive so are the slice, co-slice and functor categories over **C**

Complements of pushout along monos are unique

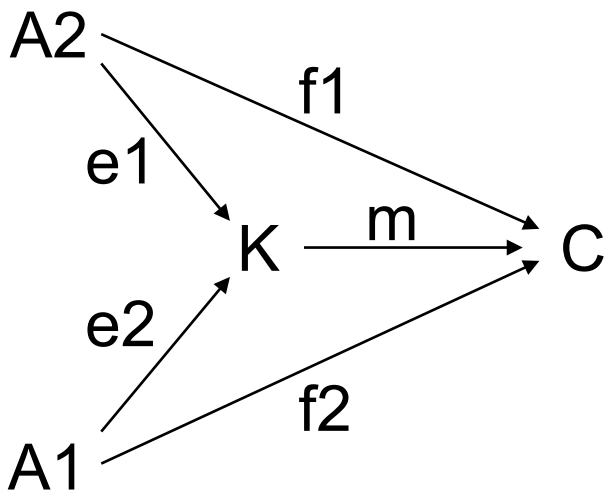






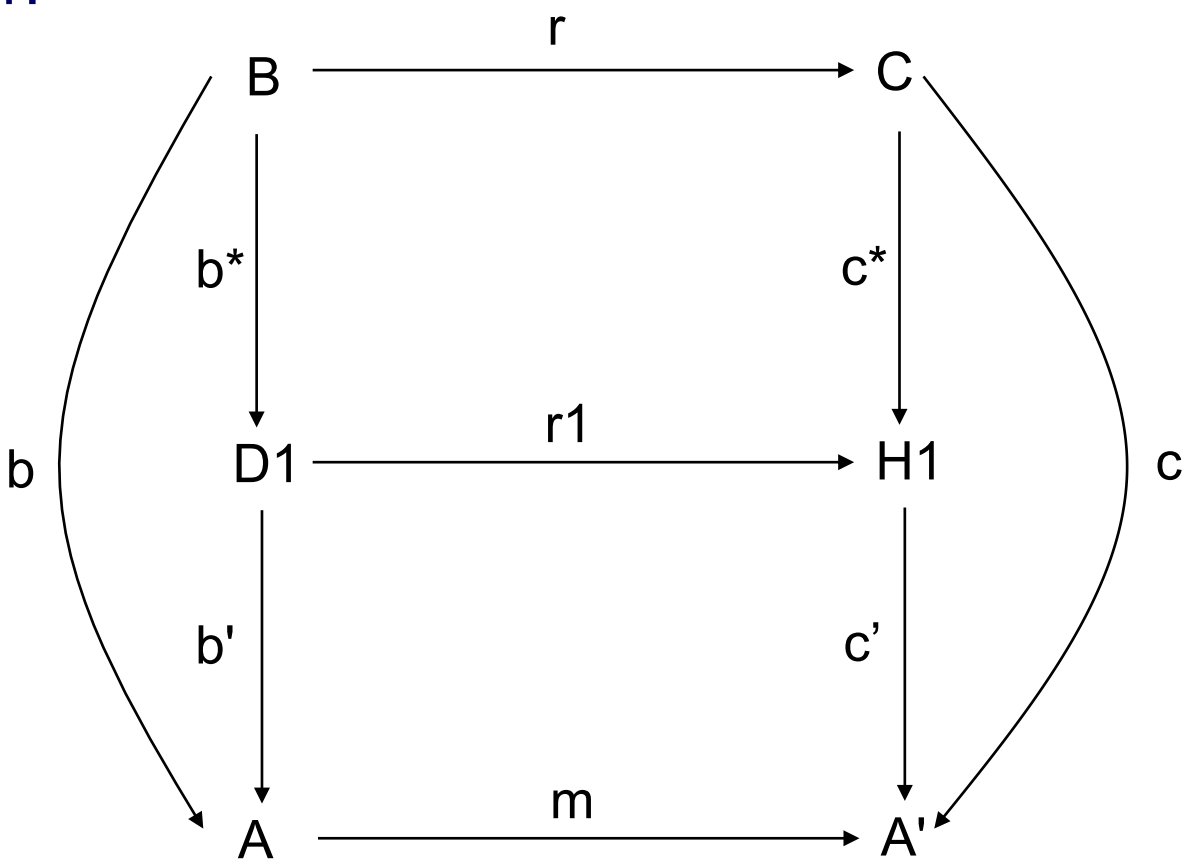
Constructions over Adhesive Categories

- **C** has epi-mono pair factorization if for every pair morphisms f1 and f2, there is an object K, jointly epimorphic morphisms e1 and e2, and a monomorphism m, such that the diagram above commutes

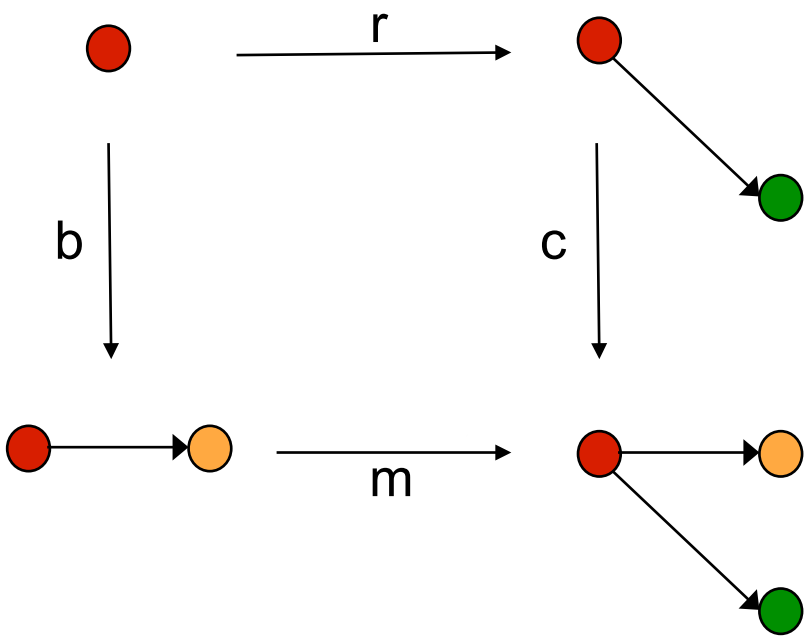


Constructions over Adhesive Categories

- **C** has initial pushouts along monomorphisms if for every morphism m, there is an initial pushout BCAA' along a monomorphism r:

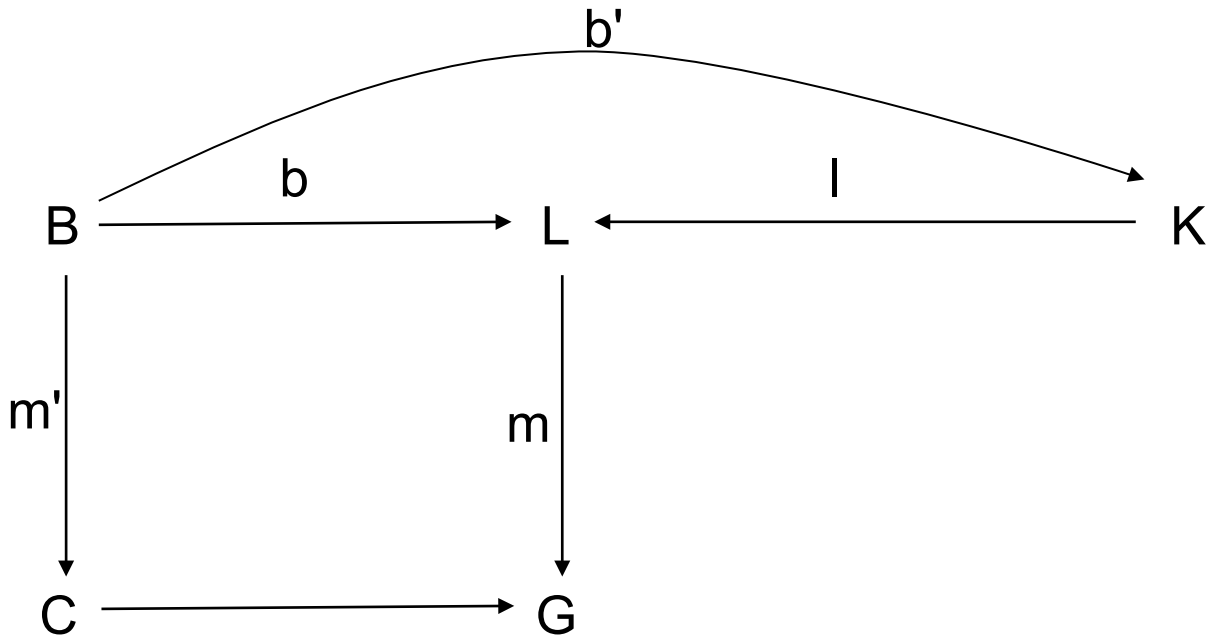


Initial Pushout



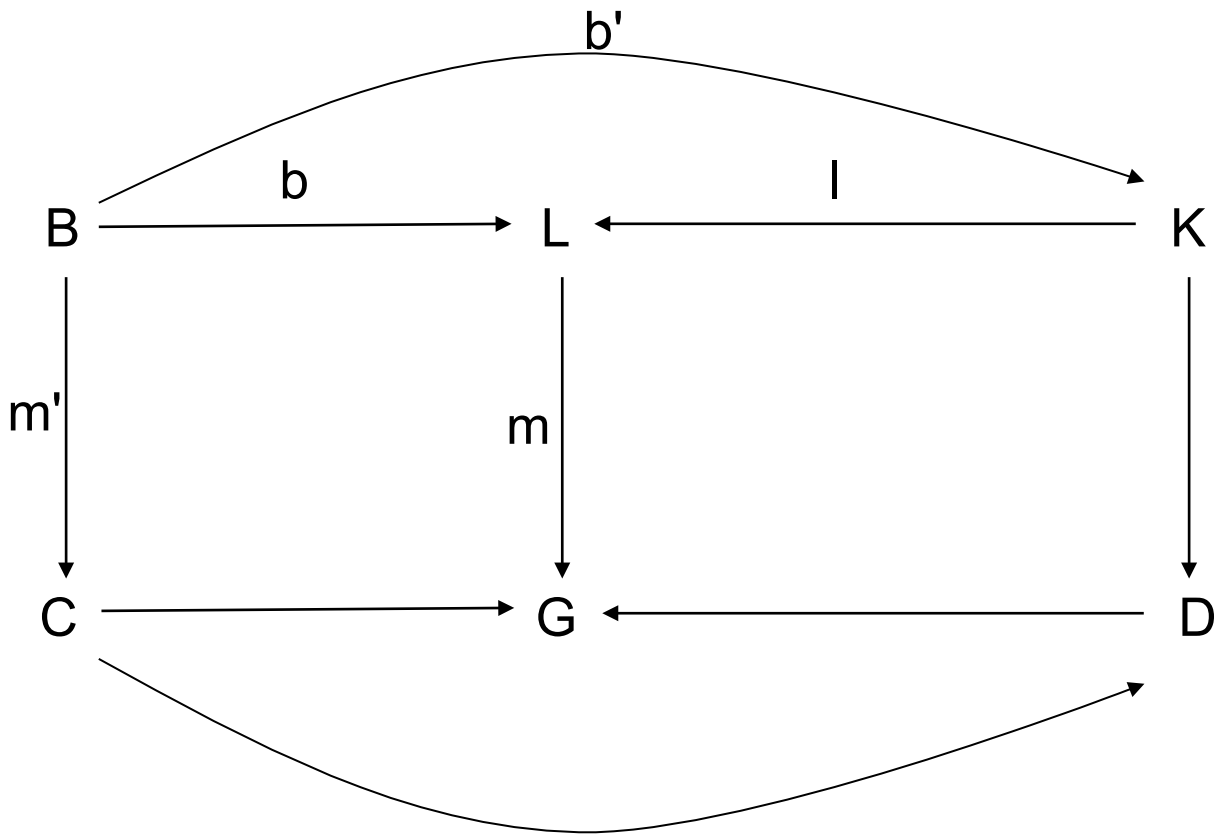
Gluing Conditions for Adhesive Categories

m is boundary consistent with respect to a monomorphism m below if there exists an initial pushout and a morphism b' such that $b = l \cdot b'$:



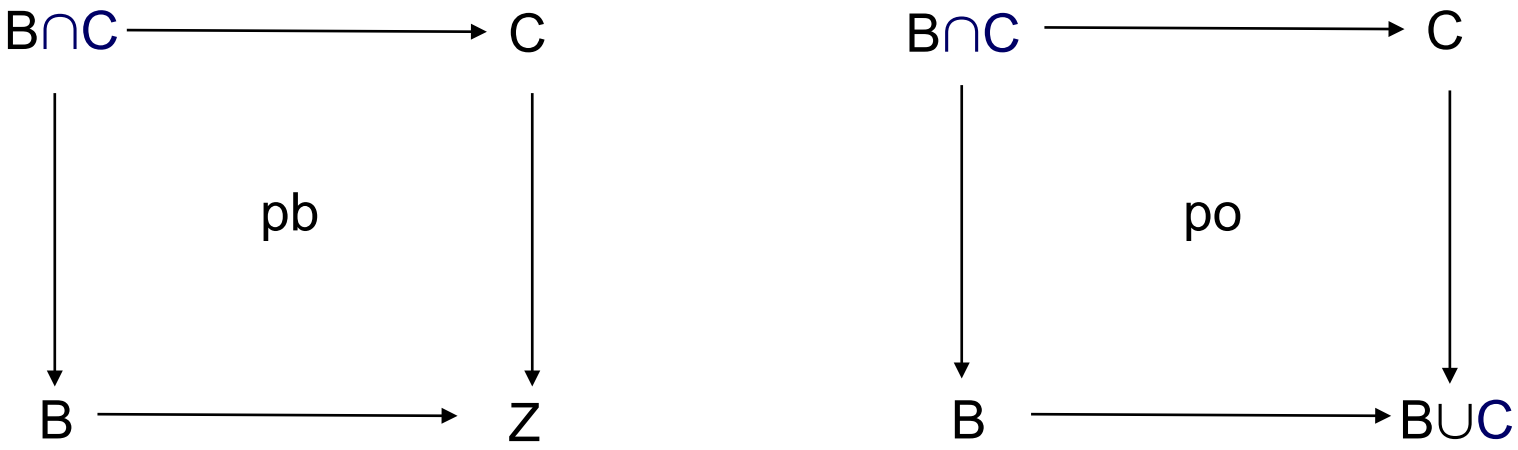
Gluing Conditions for Adhesive Categories

The morphisms l and m below have a pushout complement iff m is boundary consistent with respect to l.



Algebra of Subobjects

Given an element Z in **C** we can define an algebra of its subobjects:

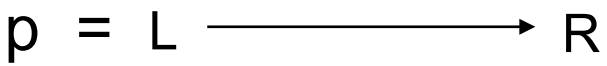


The class of subobjects of Z is a distributive lattice

Variations

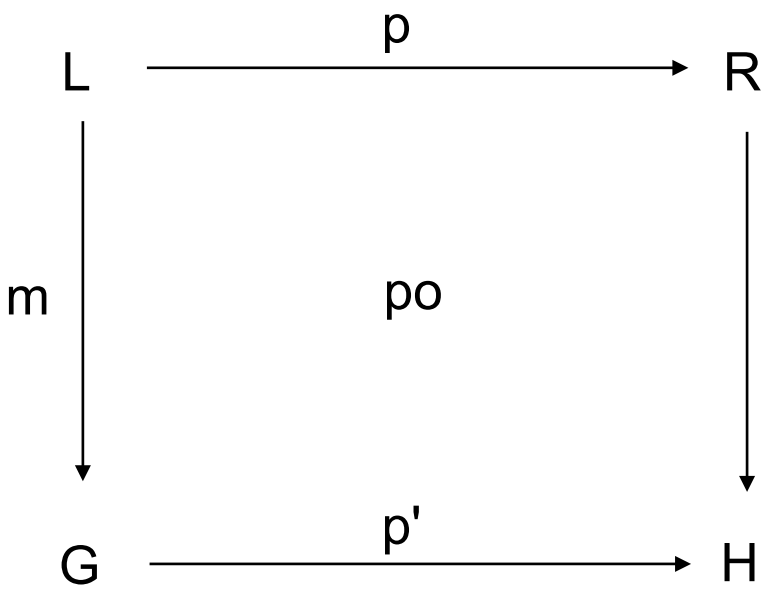
Single pushout transformations

Rules are partial monomorphisms (inclusions):



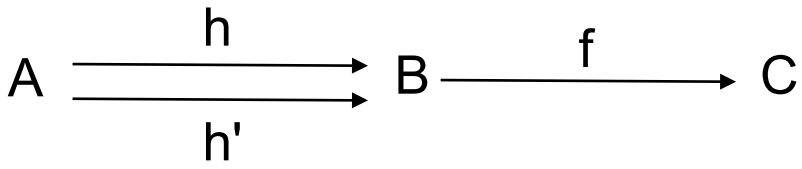
Rule Application

Roughly, single pushout graph transformation removes from G the elements that are in m(L\dom(p)) and adds to G the elements in R\p(L) but, in case of conflict, removes the conflicting elements



Rule Application

Given h, h':



the co-equalizer (C,f) is constructed in the category of graphs with partial morphisms as follows:

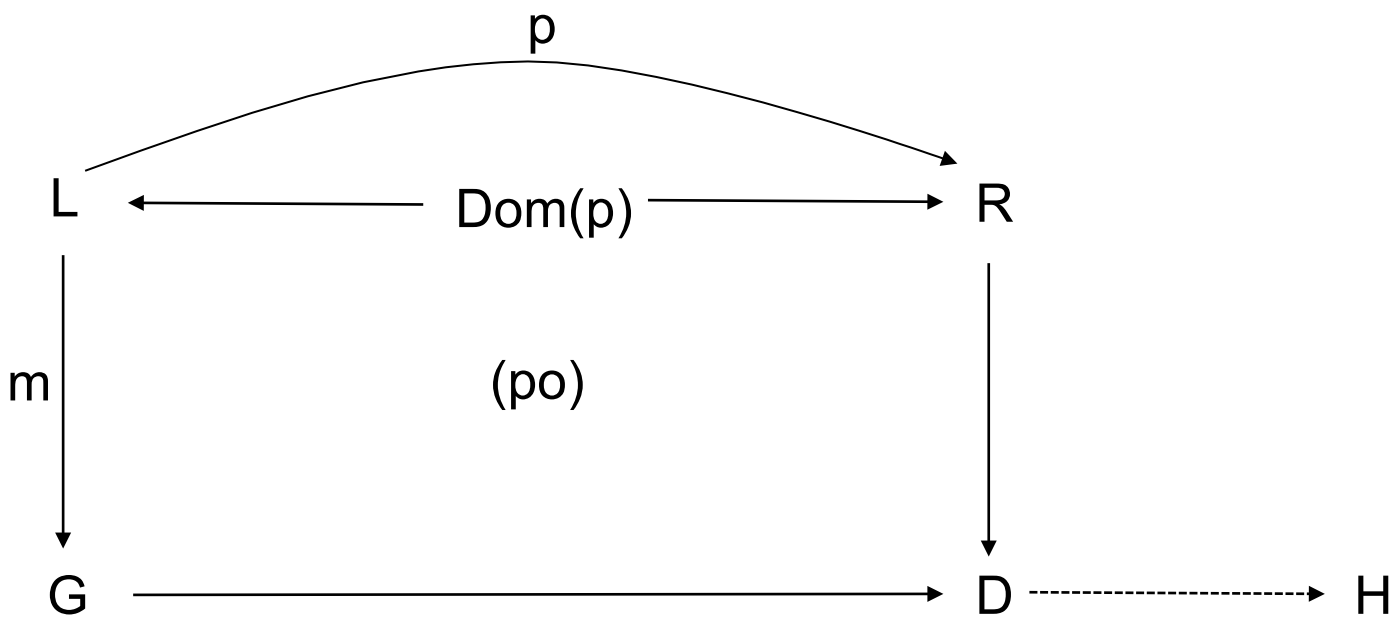
C is the largest subgraph of B and of

$$(h(A) \cap h'(A)) \cup (\overline{h(A)} \cap \overline{h'(A)})$$

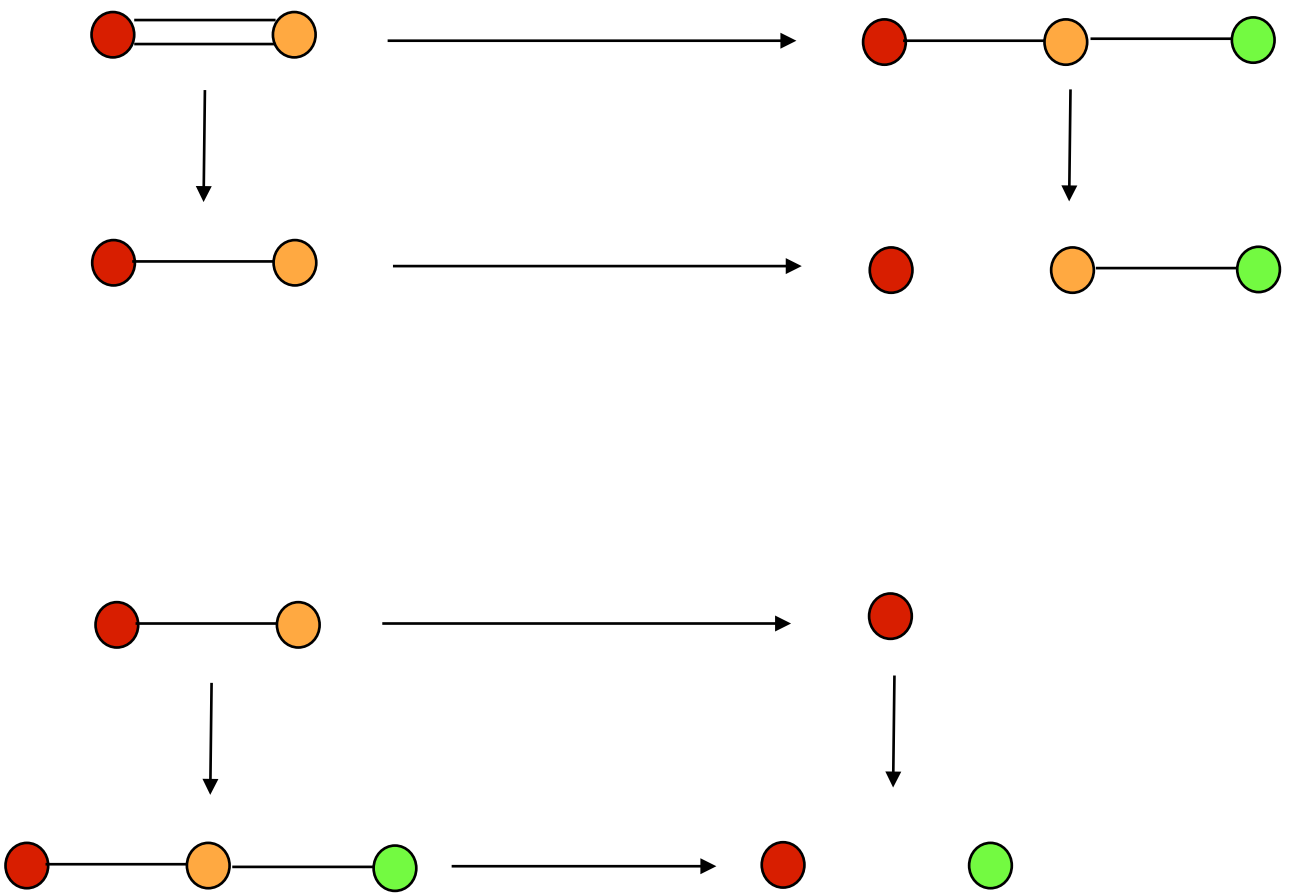
Rule Application

1. Build D as the pushout of Dom(p) → G and Dom(p) → R

2. Build H as the co-equalizer of LRD and LGD

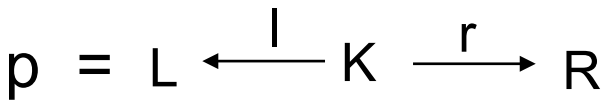


Examples

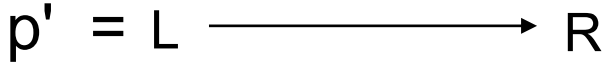


SPO and DPO

Given



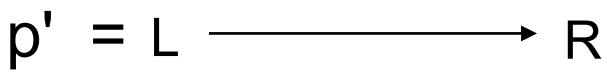
we can define



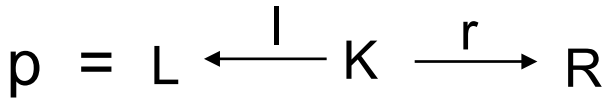
with dom(p') = K and p'(x) = r(x).

SPO and DPO

Given



we can define



with dom(p') = K and p'(x) = r(x).

SPO and DPO

If m: L → G satisfies the gluing conditions then DPO and SPO transformations coincide.

Single/Double Push-Out Furnaces

Inductotherm Group

Contact ➡

Enlarge

Product Categories

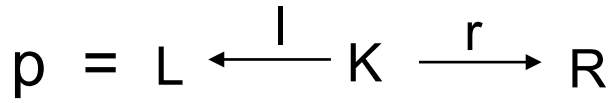
- Furnaces, Ovens & Accessories

Product Profile

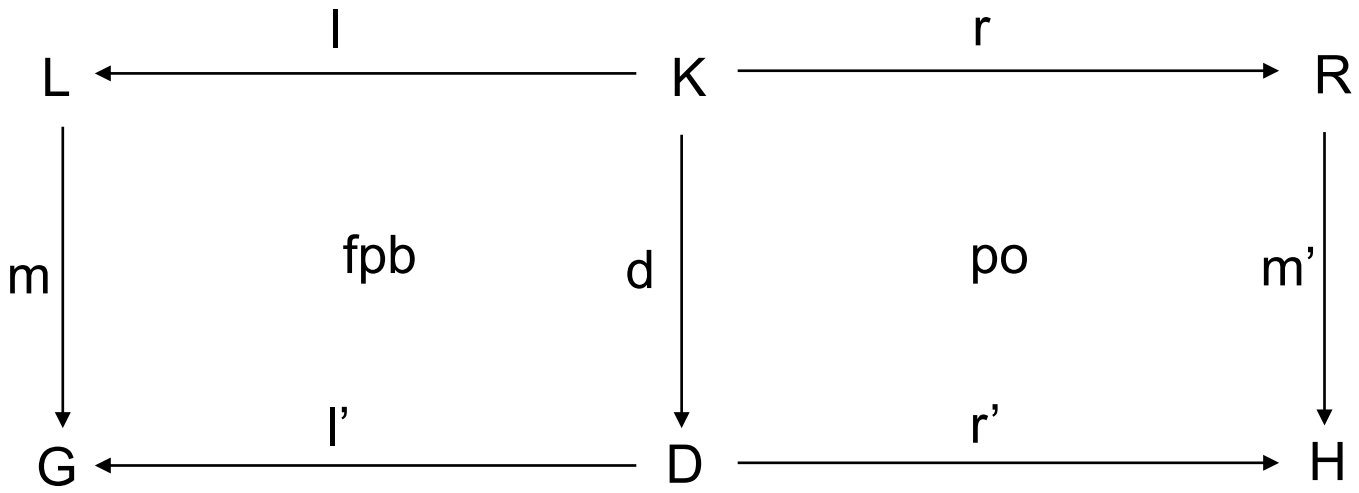
Inductotherm offers Single/Double Push-Out Induction Furnaces, which provide nonferrous and precious metal casters with clean, compact and highly productive crucible melting systems. In double push-out furnaces, power is switched from coil-to-coil in seconds and enhancing productivity.

Sesqui-pushout transformations

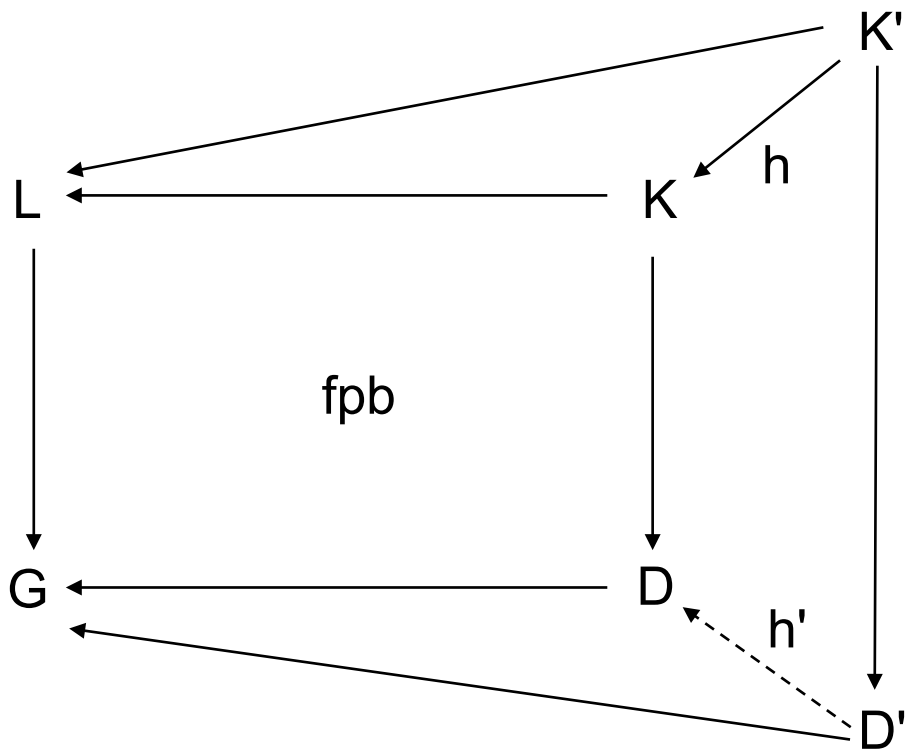
Rules are spans, like for DPO transformations:



Rule Application



Final Pullbacks



KLDG is a final pullback if for every other pullback K'LD'G and every morphism h there is a unique h such that the diagram commutes

Properties of Final Pullbacks

1. Final pullback complements are unique when they exist, even if the morphisms in the pullback are not monos.

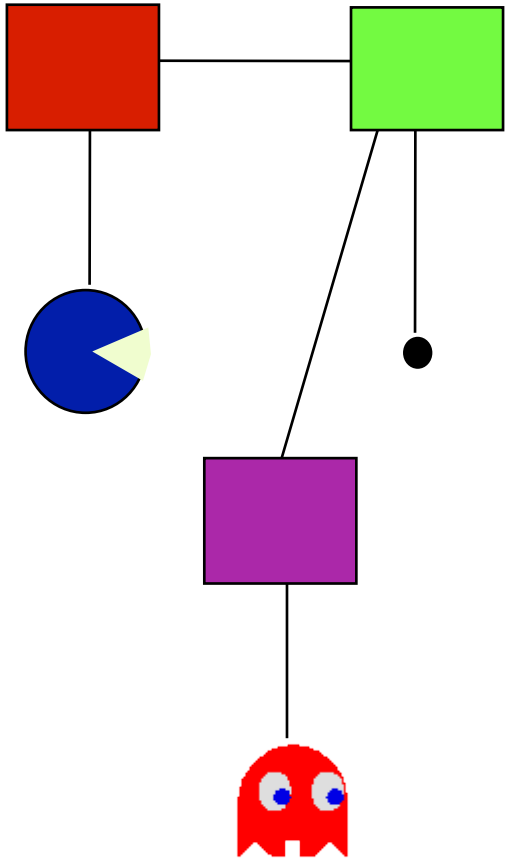
2. Final pullbacks along monos are pushouts.

3. Final pullback complements exist if and only if the identification condition holds.

4. When the morphism $K \rightarrow L$ is not a mono, the rules clone the identified elements.

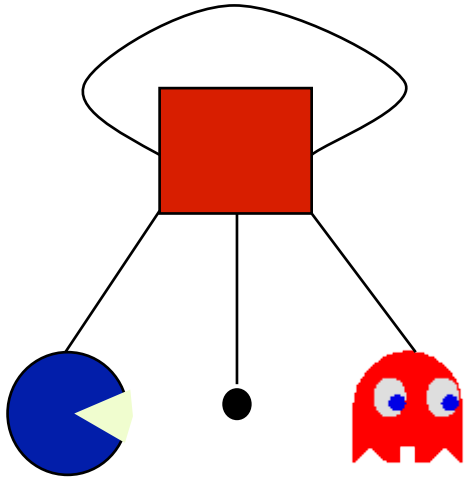
Extensions

Typed Graphs

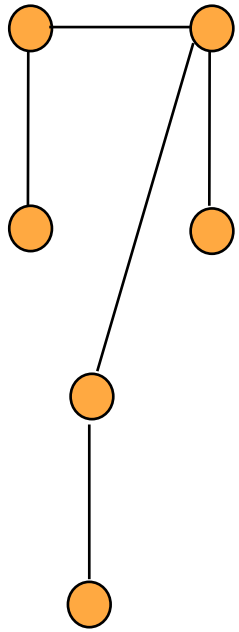


A typed graph consists of:

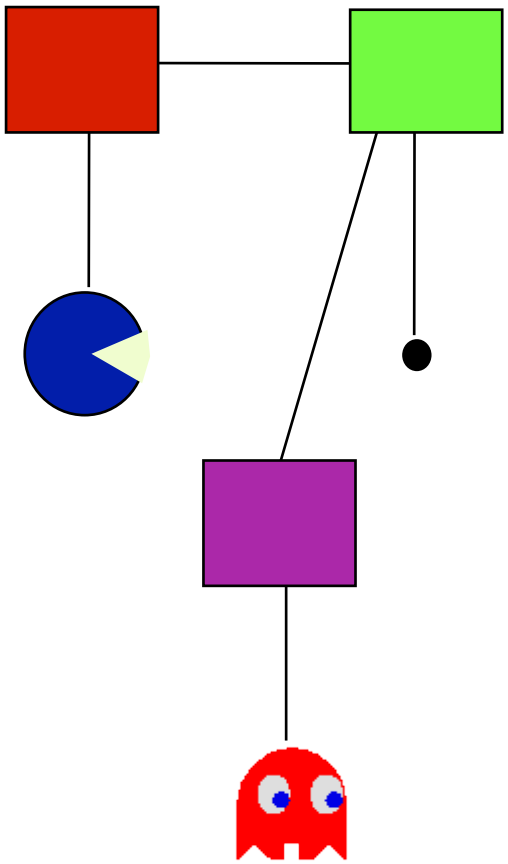
1. A type graph G_T.

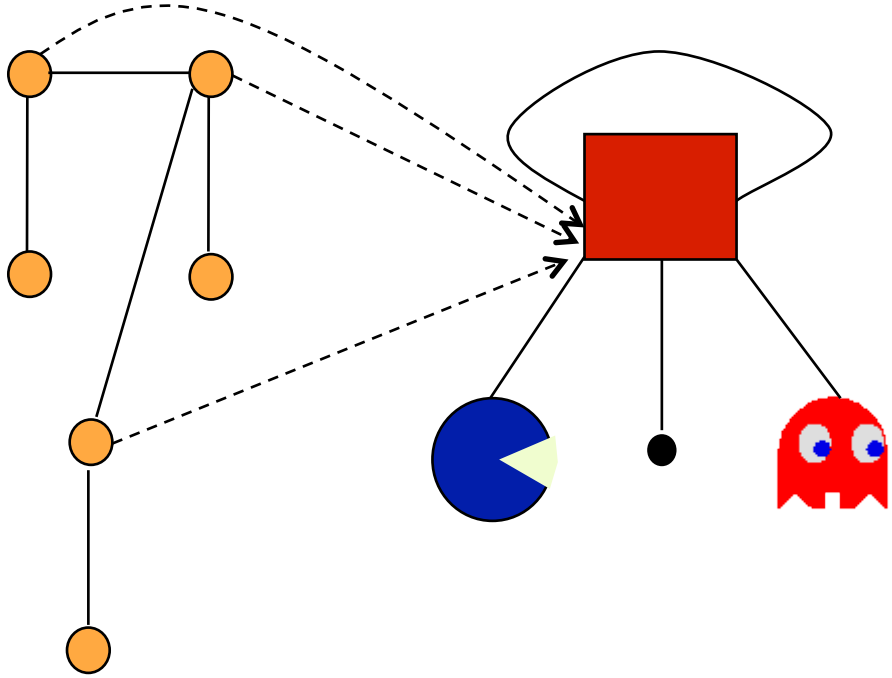


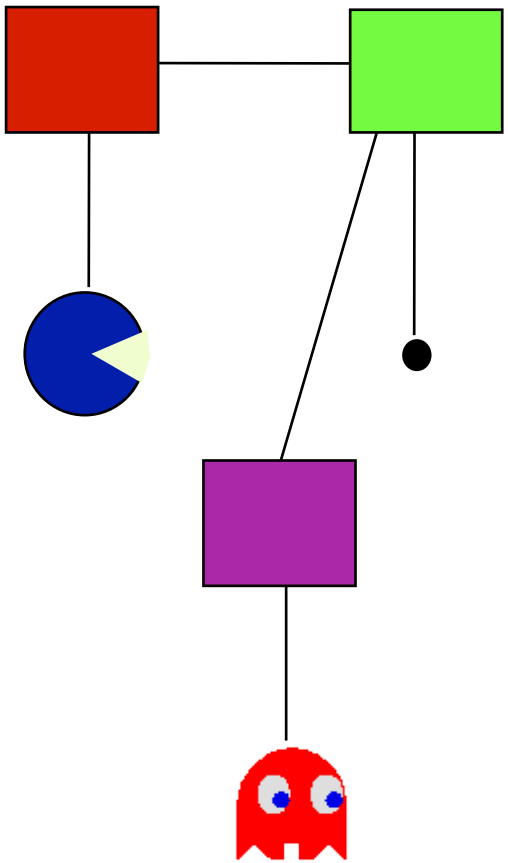
2. A graph G

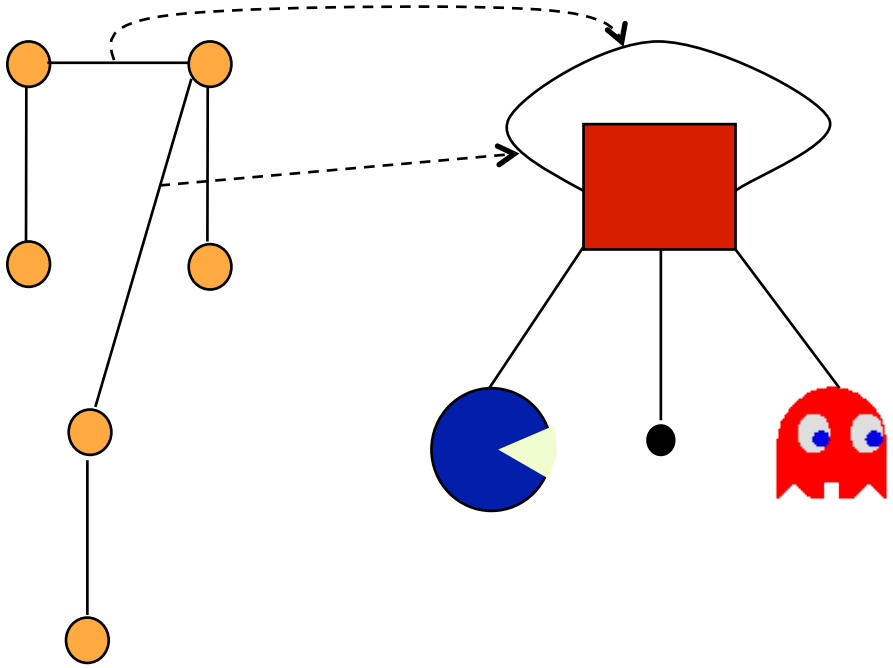


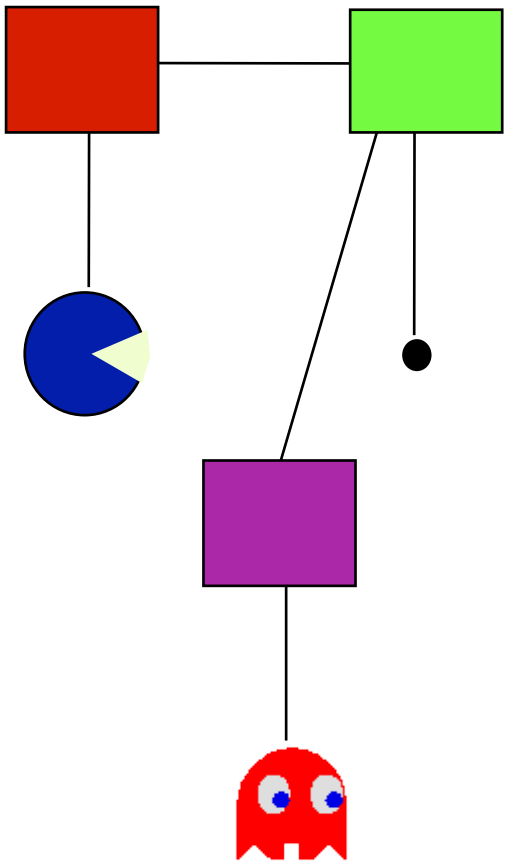
3. A morphism TG: G → G_T, assigning its type to each element in G

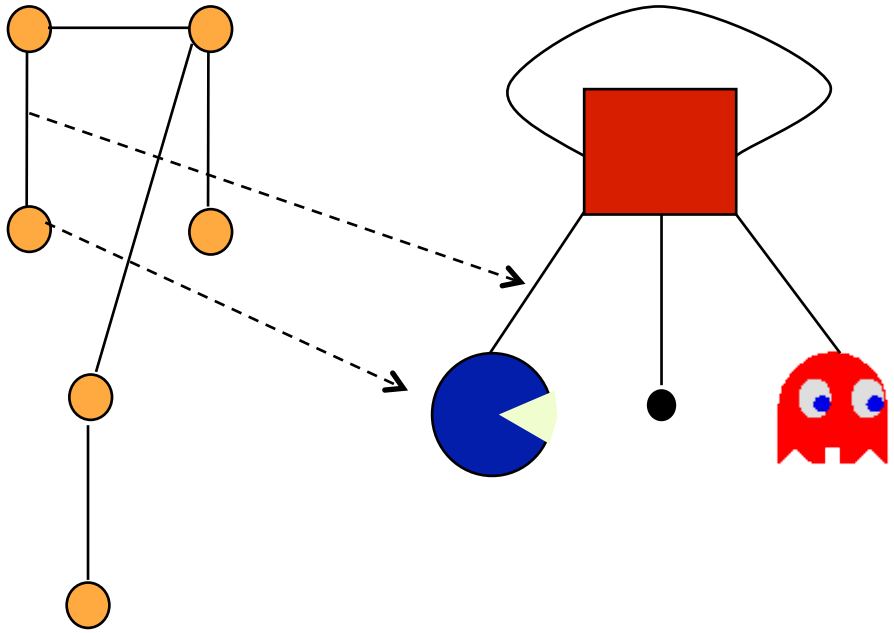




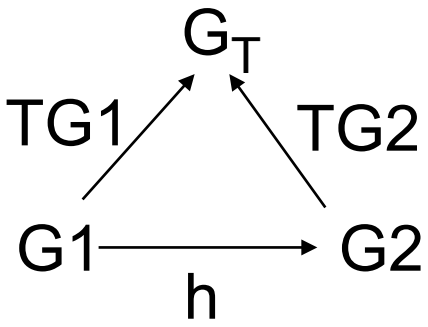








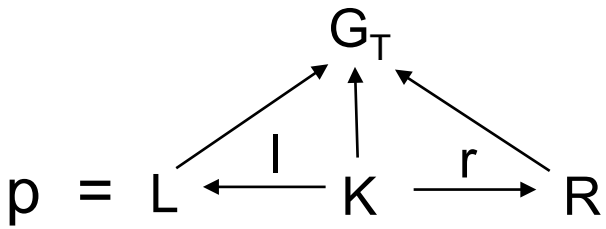
Morphisms



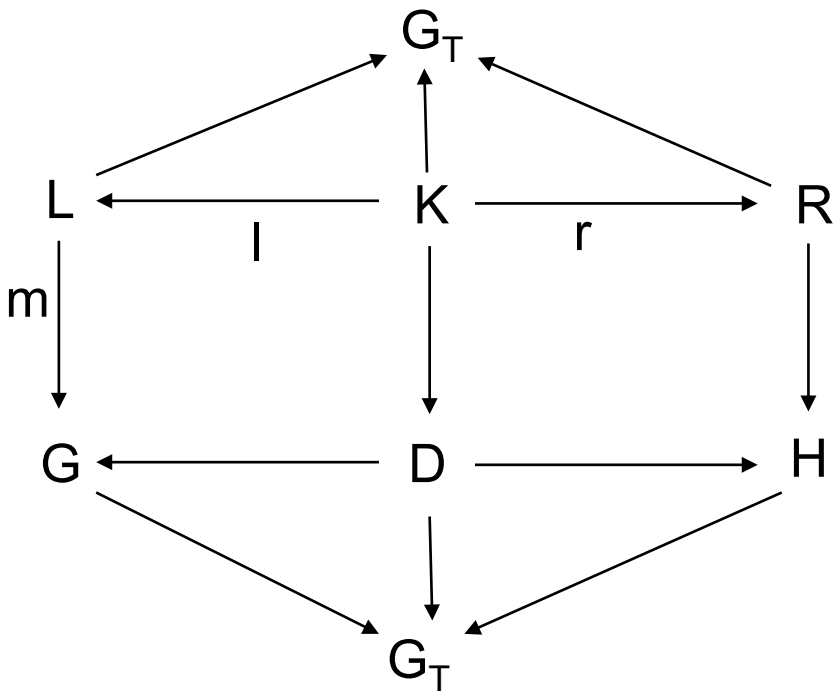
Typed Graphs Transformation

Typed graphs over a given type graph G_T are an adhesive category.

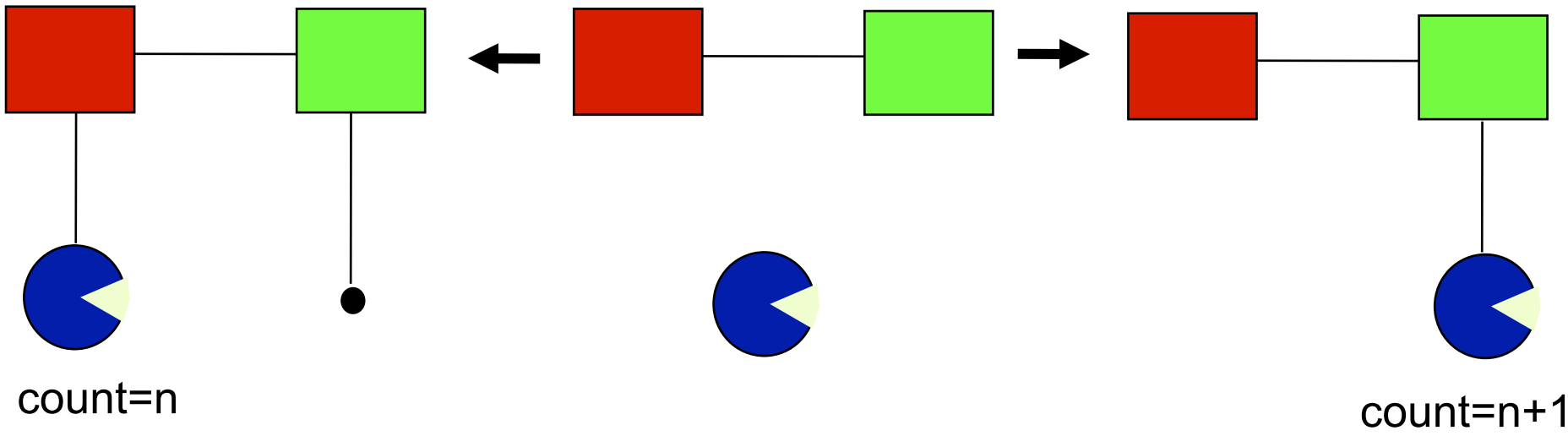
A typed graph transformation rule:



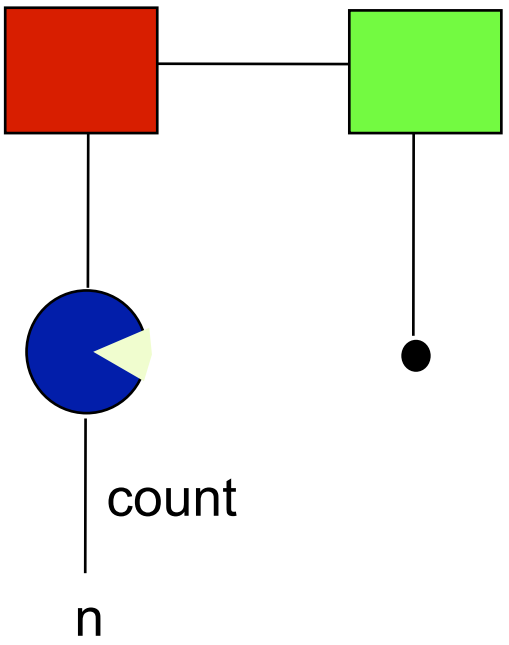
Rule Application



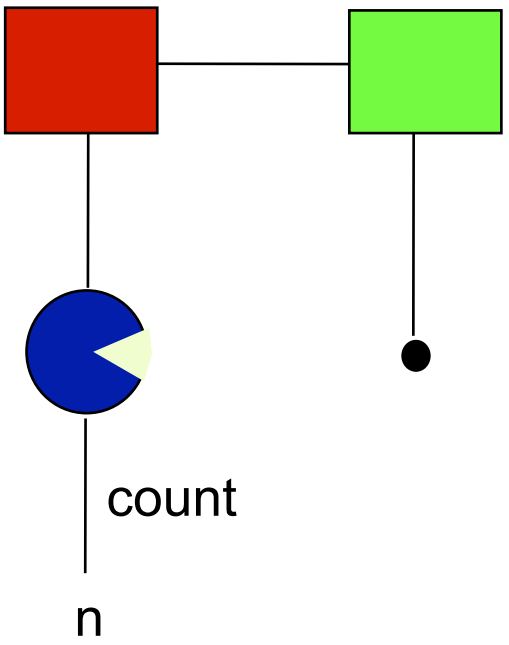
Attributed graphs



E-graphs



Attributed (Symbolic) graphs



with n ≥ 0

Attributed (Symbolic) graphs

- Symbolic graphs are pairs (G,Φ), where G is an E-graph over a set of variables X and Φ is a set of formulas over X and the operations, predicates and data of the given data algebra D.
- A symbolic graph morphism h: $(G,\Phi) \rightarrow (G',\Phi')$ is an E-graph morphism h: $G \rightarrow G'$ such that

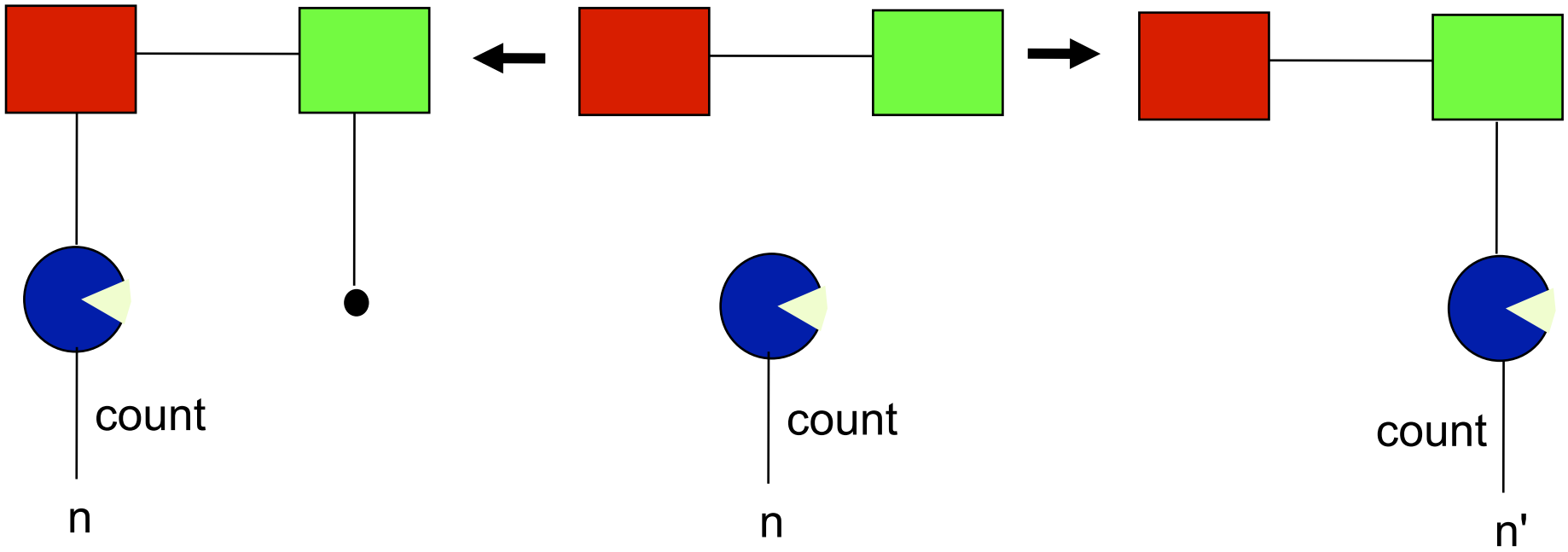
$$D \models \Phi' \Rightarrow h(\Phi)$$

Attributed (Symbolic) graphs

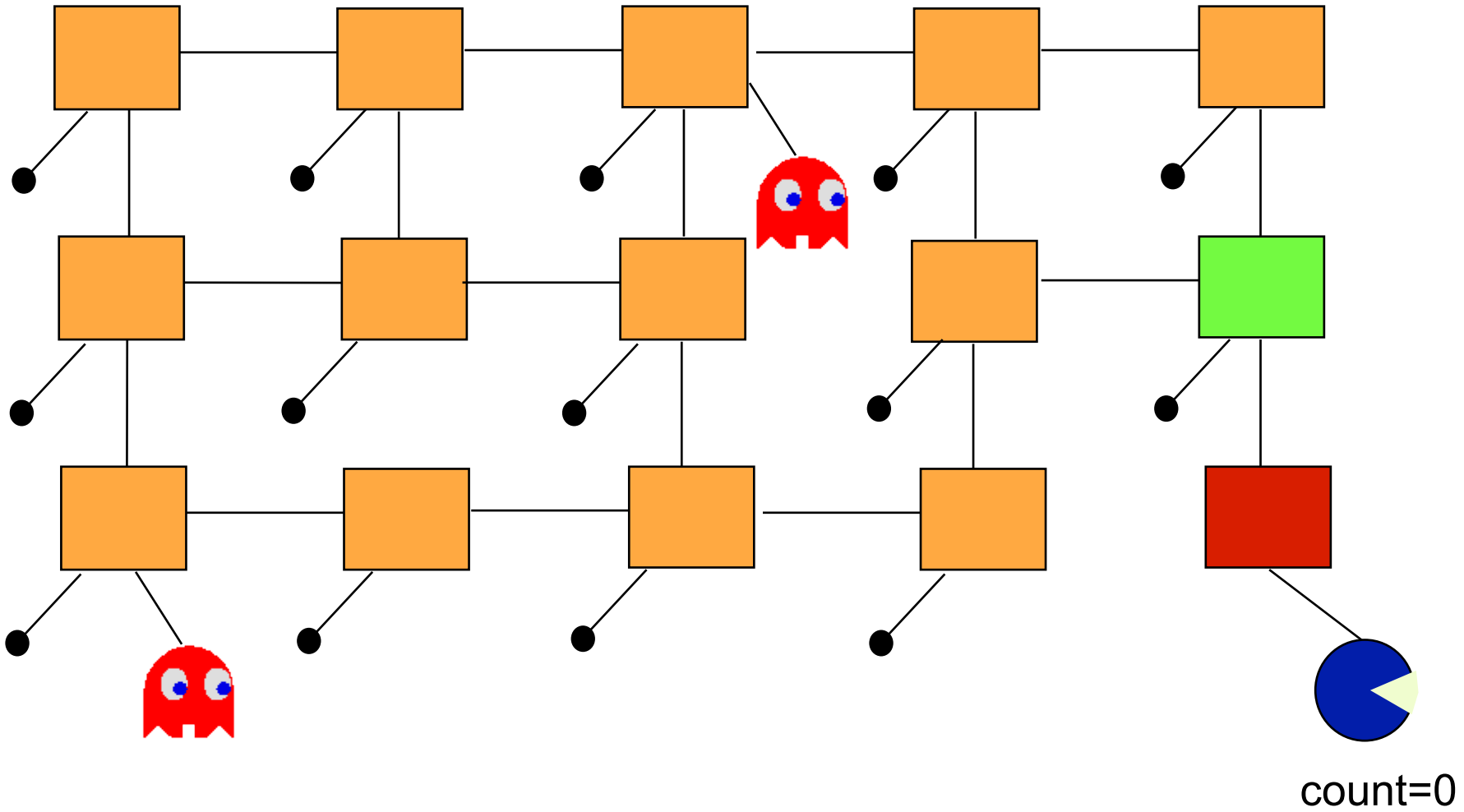
- The category of symbolic graphs is not adhesive, but it is M-adhesive, where an M-morphism $(G,\Phi) \to (G',\Phi')$ is a monomorphism such that

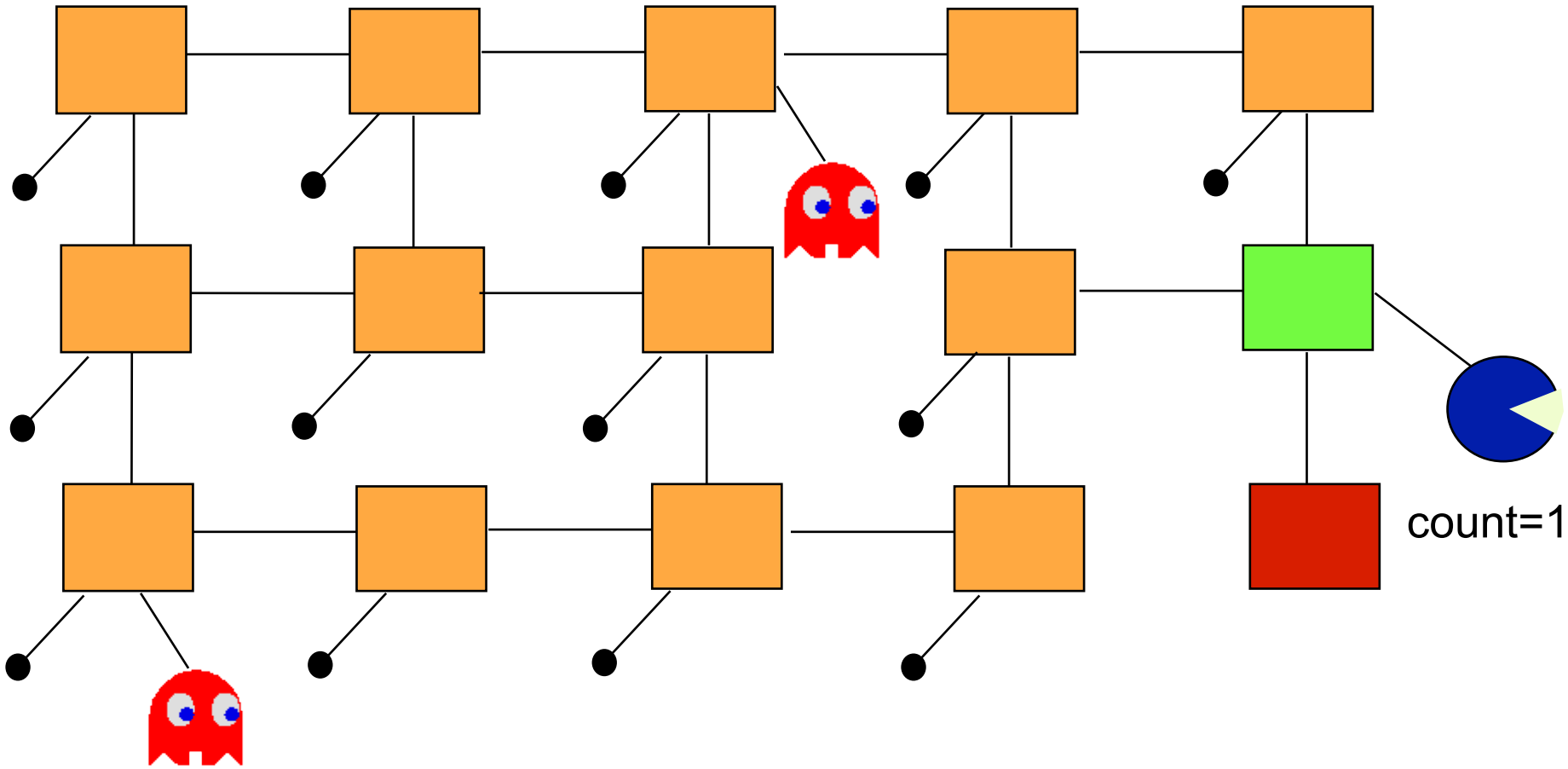
$$D \models \Phi' \equiv h(\Phi)$$

Attributed (Symbolic) graphs



with n' = n+1


count=0


count=1

¡Thank You!