A resolution-based approach to graph transformation

Ionuţ Ţuţu

Department of Computer Science,
Royal Holloway University of London

Workshop on Logic,
Algebra and Category Theory

Melbourne, 2018

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs
typed graphs
labelled graphs
multigraphs
hypergraphs

SPO derivations
DPO derivations
DPB transitions

adhesive categories

relational LP
equational LP
constraint LP
service-oriented LP
hybrid LP

resolution
paramodulation
narrowing

generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed graphs	equational LP
labelled graphs	constraint LP
multigraphs	service-oriented LP
hypergraphs	hybrid LP
SPO derivations	resolution
DPO derivations	paramodulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

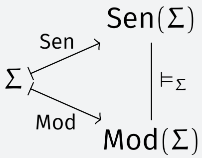
Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple g	tional LP
typed g	tional LP
labellec	traint LP
multigra	ented LP
hypergr	ybrid LP
SPO der	esolution
DPO der	dulation
DPB trar	arrowing
adhesiv	neralized systems

An institution-theoretic approach

$\langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$



rules .... sentences

graph transition systems .... models

witnessing rule application .... satisfaction

Bridging the gap between GT and LP

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

The necessary connection between DS and OS of LP

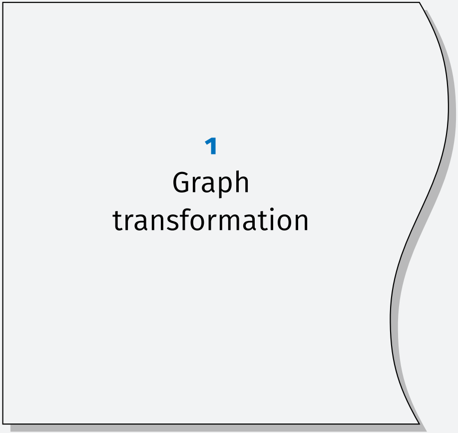
Given a logic program Γ, the answers to an existential query can be found simply by examining a term model – the least Herbrand model – instead of all the models that satisfy Γ.

1. $\Gamma \models_{\Sigma} \exists X \cdot \rho$
2. $\mathcal{O}_{\Sigma,\Gamma} \models_{\Sigma} \exists X \cdot \rho$
3. There exists $\psi \colon X \to Y$ such that $\Gamma \models_{\Sigma} \forall Y \cdot \psi(\rho)$.

substitution systems

Bridging the gap between GT and LP: Outline

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.



Bridging the gap between GT and LP: Outline

- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs
typed graphs
labelled graphs
multigraphs
hypergraphs

SPO derivations
DPO derivations
DPB transitions

adhesive categories

2
Substitution systems

Bridging the gap between GT and LP: Outline

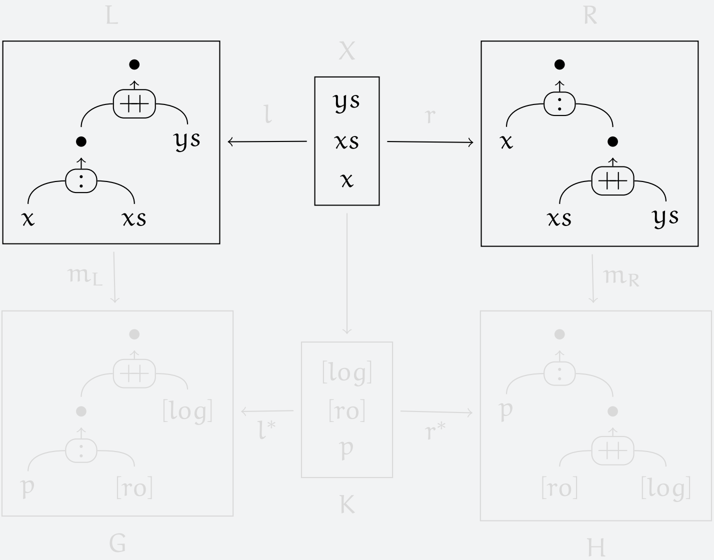
- Both Graph Transformation and Logic Programming have a long tradition, with a well-established theory and rich tool support.

simple graphs	relational LP
typed g	tional LP
labellec	straint LP
multigra	ented LP
hypergr	hybrid LP
SPO der	esolution
DPO der	dulation
DPB transitions	narrowing
adhesive categories	generalized substitution systems

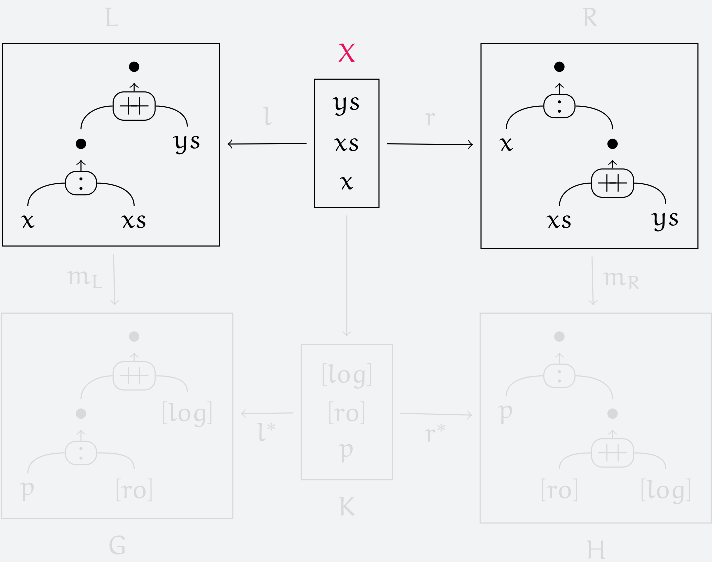
3

A logic-programming semantics for graph transformation

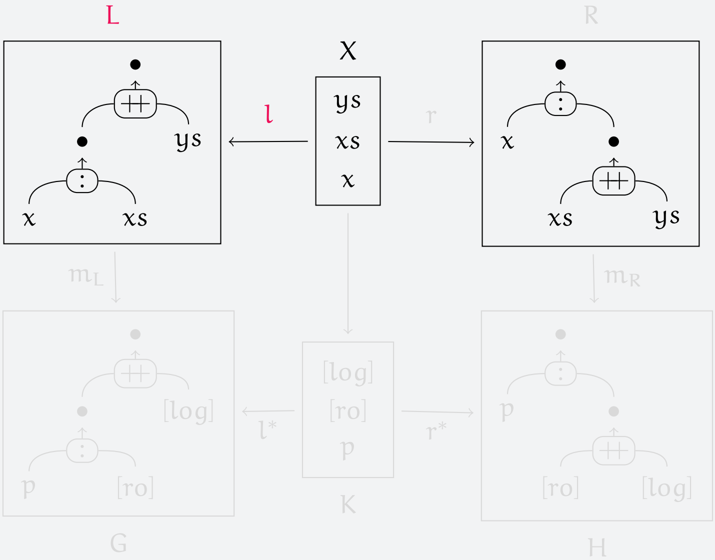
Graph transformation



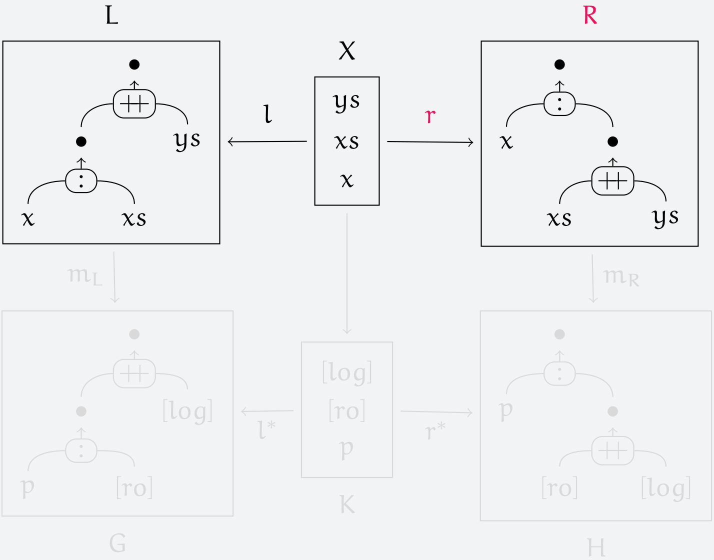
Graph transformation



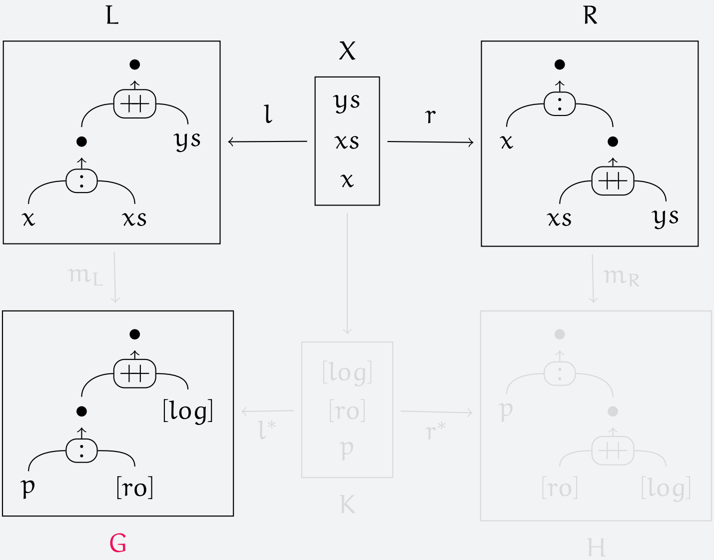
Graph transformation



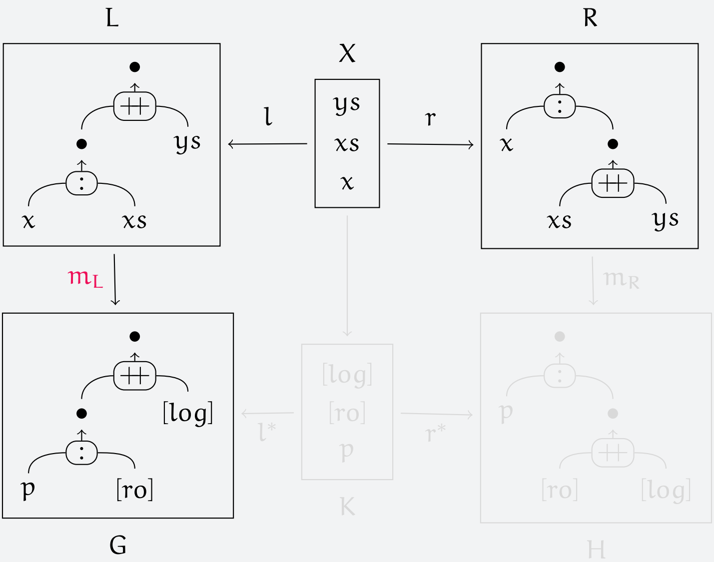
Graph transformation



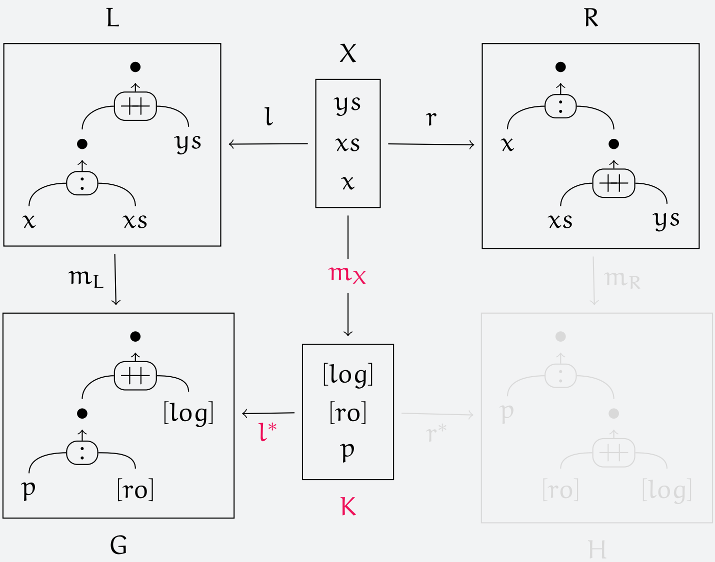
Graph transformation



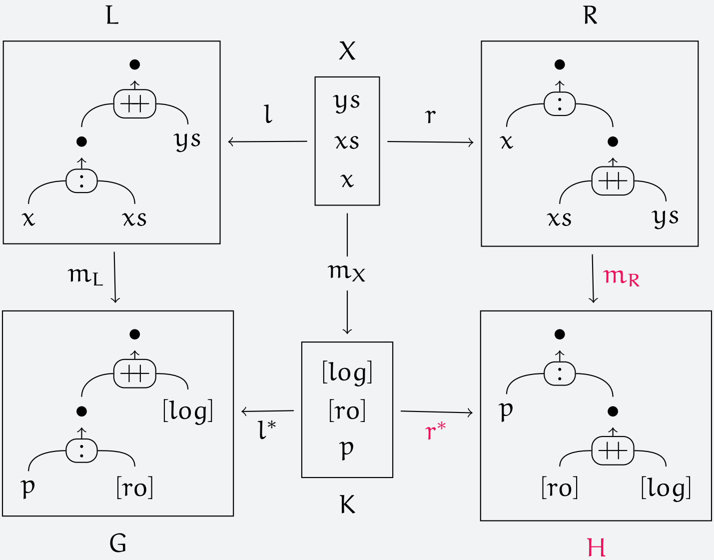
Graph transformation



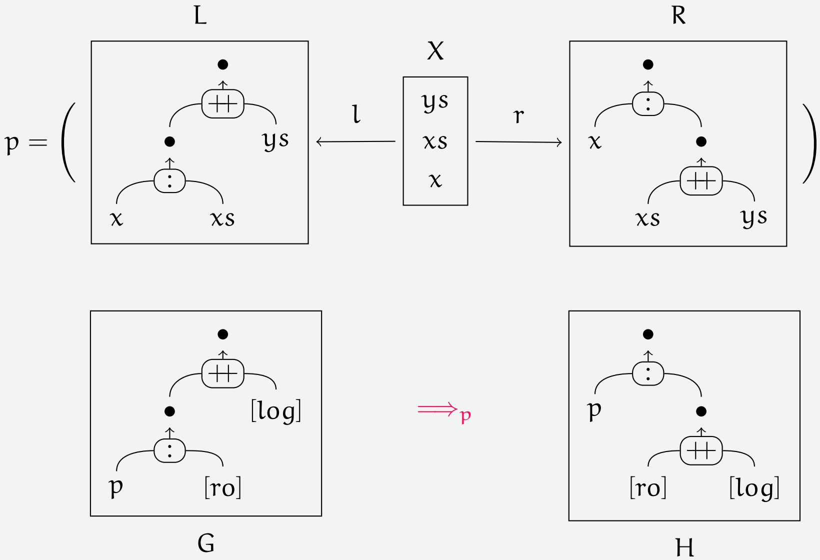
Graph transformation



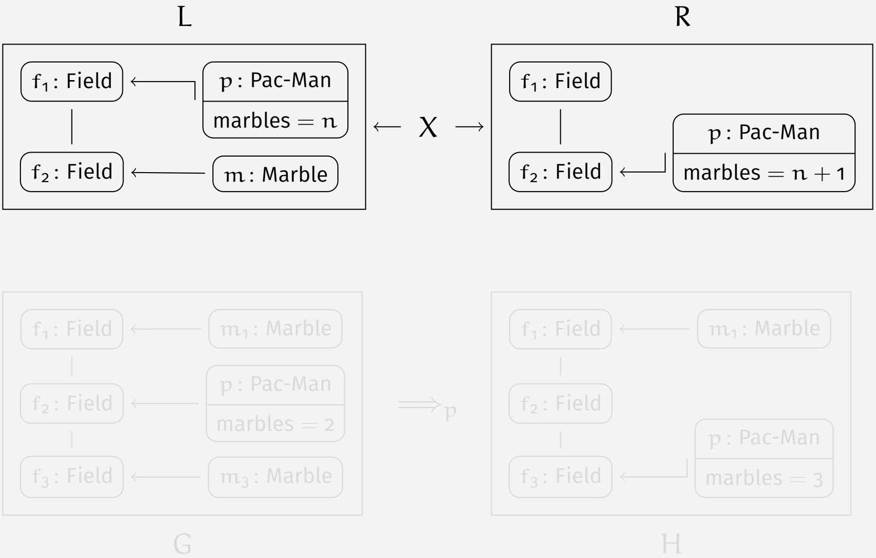
Graph transformation



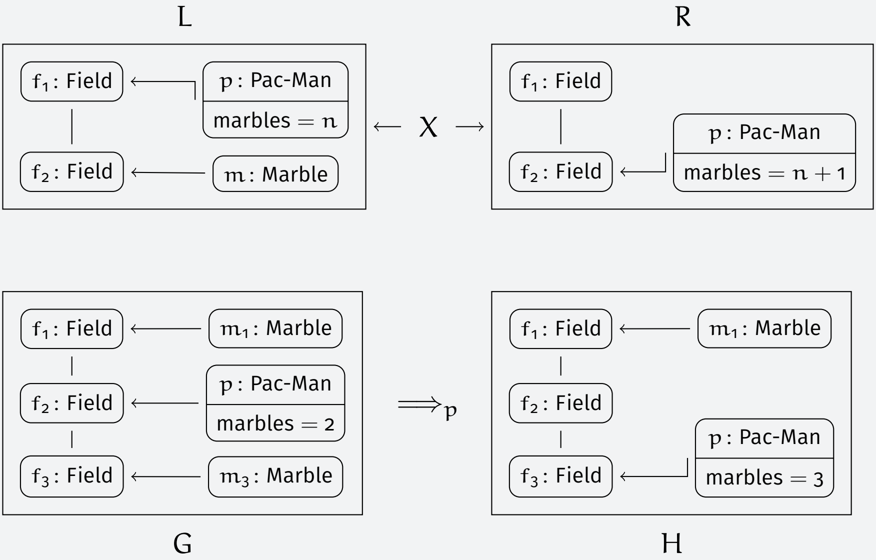
Graph transformation



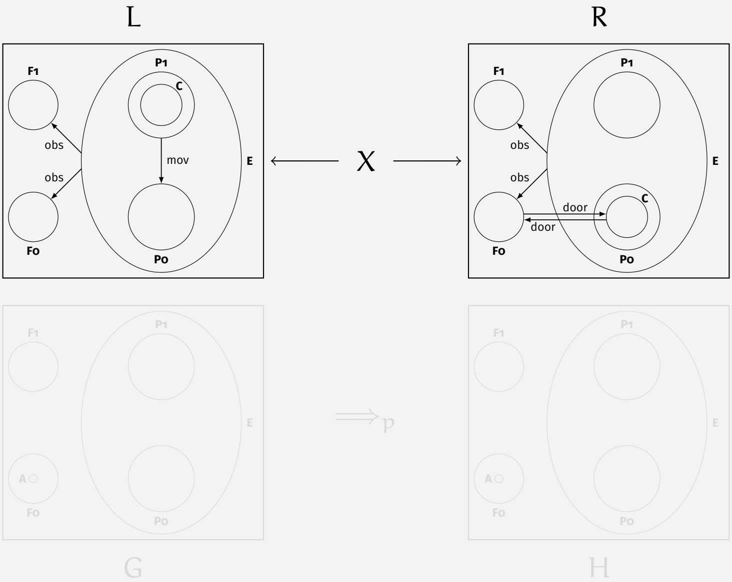
Type & instance graphs



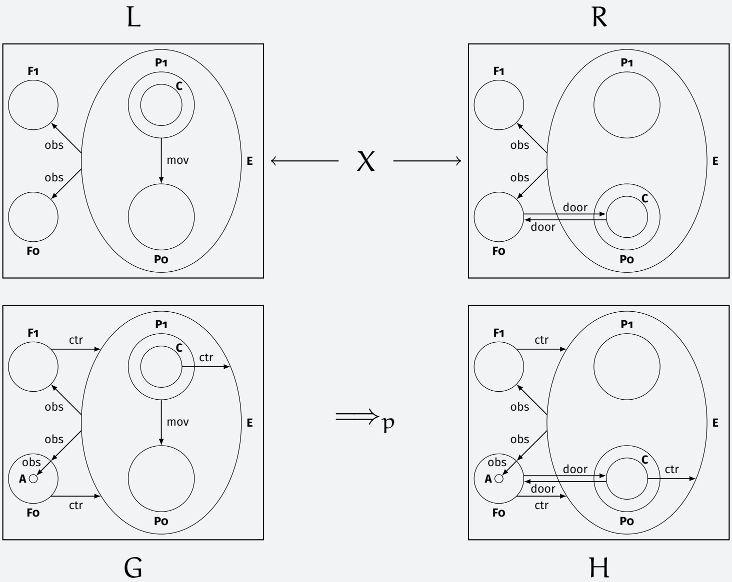
Type & instance graphs



Actor networks



Actor networks



Substitution systems

- Every institution $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$ can be regarded as a functor $\mathcal{I} \colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ into the category of rooms and corridors:

$$
\begin{array}{ccc}
\Sigma & \longmapsto & \langle \mathrm{Sen}(\Sigma), \mathrm{Mod}(\Sigma), \models_\Sigma \rangle \\
\varphi \big\downarrow & & \big\downarrow \langle \mathrm{Sen}(\varphi), \mathrm{Mod}(\varphi) \rangle \\
\Omega & \longmapsto & \langle \mathrm{Sen}(\Omega), \mathrm{Mod}(\Omega), \models_\Omega \rangle
\end{array}
$$

- A substitution system consists of
  1. a category $\mathbb{S}\mathrm{ubst}$ of signatures of variables and substitutions,
  2. a room G of ground sentences and models, and
  3. a functor $\mathcal{S} \colon \mathbb{S}\mathrm{ubst} \to G \,/\, \mathbb{R}\mathrm{oom}$.
- Universally and existentially quantified sentences can be defined:

$$\forall X \cdot \rho \qquad \exists X \cdot \rho$$

  where $X \in |\mathbb{S}\mathrm{ubst}|$ and $\rho \in \mathrm{Sen}(X)$.

Substitution systems

- Every institution $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$ can be regarded as a functor $\mathcal{I} \colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ into the category of rooms and corridors:

$$\begin{array}{ccc} \Sigma & \longmapsto & \langle \mathrm{Sen}(\Sigma), \mathrm{Mod}(\Sigma), \models_\Sigma \rangle \\ {\scriptstyle \varphi}\downarrow & & \downarrow{\scriptstyle \langle \mathrm{Sen}(\varphi), \mathrm{Mod}(\varphi) \rangle} \\ \Omega & \longmapsto & \langle \mathrm{Sen}(\Omega), \mathrm{Mod}(\Omega), \models_\Omega \rangle \end{array}$$

- A substitution system consists of
  1. a category $\mathbb{S}\mathrm{ubst}$ of signatures of variables and substitutions,
  2. a room G of ground sentences and models, and
  3. a functor $\mathcal{S} \colon \mathbb{S}\mathrm{ubst} \to G \,/\, \mathbb{R}\mathrm{oom}$.
- Universally and existentially quantified sentences can be defined:

$$\forall X \cdot \rho \qquad \exists X \cdot \rho$$

where $X \in |\mathbb{S}\mathrm{ubst}|$ and $\rho \in \mathrm{Sen}(X)$.

Substitution systems

- Every institution $\mathcal{I} = \langle \mathbb{S}\mathrm{ig}, \mathrm{Sen}, \mathrm{Mod}, \models \rangle$ can be regarded as a functor $\mathcal{I} \colon \mathbb{S}\mathrm{ig} \to \mathbb{R}\mathrm{oom}$ into the category of rooms and corridors:

$$\begin{array}{ccc} \Sigma & \longmapsto & \langle \mathrm{Sen}(\Sigma), \mathrm{Mod}(\Sigma), \models_\Sigma \rangle \\ \varphi \downarrow & & \downarrow \langle \mathrm{Sen}(\varphi), \mathrm{Mod}(\varphi) \rangle \\ \Omega & \longmapsto & \langle \mathrm{Sen}(\Omega), \mathrm{Mod}(\Omega), \models_\Omega \rangle \end{array}$$

- A substitution system consists of
  1. a category $\mathbb{S}\mathrm{ubst}$ of signatures of variables and substitutions,
  2. a room G of ground sentences and models, and
  3. a functor $\mathcal{S} \colon \mathbb{S}\mathrm{ubst} \to G \,/\, \mathbb{R}\mathrm{oom}$.
- Universally and existentially quantified sentences can be defined:

$$\forall X \cdot \rho \qquad \exists X \cdot \rho$$

where $X \in |\mathbb{S}\mathrm{ubst}|$ and $\rho \in \mathrm{Sen}(X)$.

A substitution system for GT: Prerequisites

Consider an adhesive category $\mathbb{C}$. Then the following prop. hold:

1. $\mathbb{C}$ has pushouts along monomorphisms, and monomorphisms are stable under pushouts.
2. $\mathbb{C}$ has pullbacks, and pushouts along monomorphisms are also pullbacks.
3. Given a commutative cube diagram, if the arrows in the top and bottom squares are mono, the top square is a pullback, and the front and back squares are pushouts, then the bottom square is a pullback too.

A substitution system for GT: Prerequisites

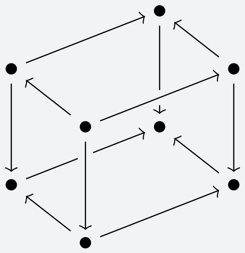
Consider an adhesive category $\mathbb{C}$. Then the following prop. hold:

1. $\mathbb{C}$ has pushouts along monomorphisms, and monomorphisms are stable under pushouts.
2. $\mathbb{C}$ has pullbacks, and pushouts along monomorphisms are also pullbacks.
3. Given a commutative cube diagram, if the arrows in the top and bottom squares are mono, the top square is a pullback, and the front and back squares are pushouts, then the bottom square is a pullback too.

A substitution system for GT: Prerequisites

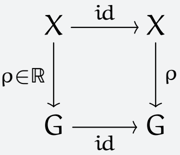
Consider an adhesive category $\mathbb{C}$. Then the following prop. hold:

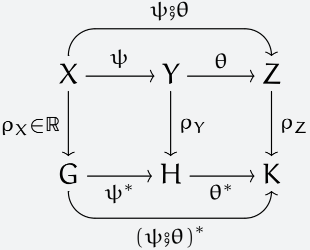
1. $\mathbb{C}$ has pushouts along monomorphisms, and monomorphisms are stable under pushouts.
2. $\mathbb{C}$ has pullbacks, and pushouts along monomorphisms are also pullbacks.
3. Given a commutative cube diagram, if the arrows in the top and bottom squares are mono, the top square is a pullback, and the front and back squares are pushouts, then the bottom square is a pullback too.



A substitution system for GT: Prerequisites (continued)

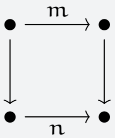
- We assume that $\mathbb{C}$ is equipped with a monic quantification space $\mathbb{R}$, i.e. a class of monomorphisms for which there exists a functorial choice of pushouts along any arrow in $\mathbb{C}$.





A substitution system for GT: Prerequisites (continued)

- In addition, we assume a class $\mathbb{T}$ of commutative squares



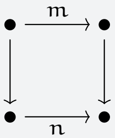
(for which the arrows m and n are monomorphisms)

such that

1. all pushouts along monomorphisms are in $\mathbb{T}$, and
2. for any commutative two-square diagram such that the top square is a pushout along monomorphisms, the outer square belongs to $\mathbb{T}$ if and only if the bottom square belongs to $\mathbb{T}$.

A substitution system for GT: Prerequisites (continued)

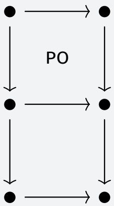
- In addition, we assume a class $\mathbb{T}$ of commutative squares



(for which the arrows $\mathfrak{m}$ and $\mathfrak{n}$ are monomorphisms)

such that

1. all pushouts along monomorphisms are in $\mathbb{T}$, and
2. for any commutative two-square diagram such that the top square is a pushout along monomorphisms, the outer square belongs to $\mathbb{T}$ if and only if the bottom square belongs to $\mathbb{T}$.



A substitution system for graph transformation

- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}$rans($\mathbb{C}$) be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;

  - the composition of arrows is defined by means of pullbacks.

Then a ground model is a broad subcategory of $\mathbb{T}$rans($\mathbb{C}$).

A substitution system for graph transformation

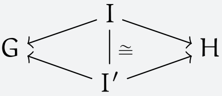
- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}$rans($\mathbb{C}$) be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;

  - the composition of arrows is defined by means of pullbacks.

Then a ground model is a broad subcategory of $\mathbb{T}$rans($\mathbb{C}$).

A substitution system for graph transformation

- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}$rans$(\mathbb{C})$ be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;

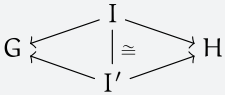


  - the composition of arrows is defined by means of pullbacks.

Then a ground model is a broad subcategory of $\mathbb{T}$rans$(\mathbb{C})$.

A substitution system for graph transformation

- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}\text{rans}(\mathbb{C})$ be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;



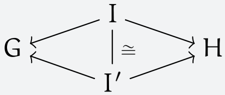
  - the composition of arrows is defined by means of pullbacks.

$$G \longleftarrow I \longrightarrow H \longleftarrow J \longrightarrow K$$

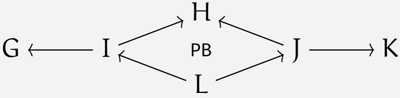
Then a ground model is a broad subcategory of $\mathbb{T}\text{rans}(\mathbb{C})$.

A substitution system for graph transformation

- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}\mathrm{rans}(\mathbb{C})$ be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;



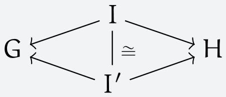
  - the composition of arrows is defined by means of pullbacks.



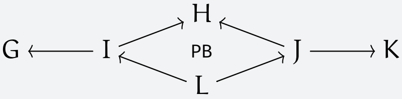
Then a ground model is a broad subcategory of $\mathbb{T}\mathrm{rans}(\mathbb{C})$.

A substitution system for graph transformation

- The category $\mathbb{S}$ubst of variables and substitutions is just $\mathbb{C}$.
- For simplicity, we define the set of ground sentences to be empty. (Hence, the ground satisfaction relation is empty also.)
- Concerning ground models, let $\mathbb{T}\mathrm{rans}(\mathbb{C})$ be the category in which:
  - the objects are objects of $\mathbb{C}$;
  - the arrows are abstract spans of monomorphisms in $\mathbb{C}$;



  - the composition of arrows is defined by means of pullbacks.



Then a ground model is a broad subcategory of $\mathbb{T}\mathrm{rans}(\mathbb{C})$.

A substitution system for graph transformation (continued)

- A sentence over $X \in |\mathbb{S}\mathrm{ubst}|$ is a span of morphisms in $\mathbb{R}$.

$$L \xleftarrow{\ l\ } X \xrightarrow{\ r\ } R$$

- A valuation for X consists of a graph transition system M and a morphism $\nu \colon X \to K$ in $\mathbb{C}$.
- $\langle M, \nu \rangle \vDash_X l \to r$ if and only if every graph transition generated from the span (l, r) along the morphism ν is in M:

$$\begin{array}{ccccc} L & \xleftarrow{\ l\ } & X & \xrightarrow{\ r\ } & R \\ \downarrow & \mathbb{T} & \downarrow \nu & \mathbb{T} & \downarrow \\ G & \longleftarrow & K & \longrightarrow & H \end{array}$$

A substitution system for graph transformation (continued)

- A sentence over $X \in |\mathbb{S}\mathrm{ubst}|$ is a span of morphisms in $\mathbb{R}$.

$$L \xleftarrow{\;l\;} X \xrightarrow{\;r\;} R$$

- A valuation for X consists of a graph transition system M and a morphism $\nu \colon X \to K$ in $\mathbb{C}$.
- $\langle M, \nu \rangle \models_X l \to r$ if and only if every graph transition generated from the span (l, r) along the morphism ν is in M:

$$\begin{array}{ccccc} L & \xleftarrow{\;l\;} & X & \xrightarrow{\;r\;} & R \\ \downarrow & \mathbb{T} & \downarrow \nu & \mathbb{T} & \downarrow \\ G & \longleftarrow & K & \longrightarrow & H \end{array}$$

A substitution system for graph transformation (continued)

- A sentence over $X \in |\mathbb{S}\mathrm{ubst}|$ is a span of morphisms in $\mathbb{R}$.

$$L \xleftarrow{\ l\ } X \xrightarrow{\ r\ } R$$

- A valuation for X consists of a graph transition system M and a morphism $\nu \colon X \to K$ in $\mathbb{C}$.
- $\langle M, \nu \rangle \vDash_X l \to r$ if and only if every graph transition generated from the span (l, r) along the morphism ν is in M:

$$\begin{array}{ccccc} L & \xleftarrow{\ l\ } & X & \xrightarrow{\ r\ } & R \\ \downarrow & \mathbb{T} & \downarrow \nu & \mathbb{T} & \downarrow \\ G & \longleftarrow & K & \longrightarrow & H \end{array}$$

A substitution system for graph transformation (continued)

- Sentences (over signature of variables X) are translated along substitutions $\psi\colon X \to Y$ by means of dedicated pushouts:

$$\begin{array}{ccccc} L & \xleftarrow{\;l_X\;} & X & \xrightarrow{\;r_X\;} & R \\ \downarrow & \mathbb{R} & \downarrow \psi & \mathbb{R} & \downarrow \\ G & \xleftarrow[\;l_Y\;]{} & Y & \xrightarrow[\;r_Y\;]{} & H \end{array}$$

- The reduction of valuations is defined by precomposition with ψ:

$$\langle M, \nu \rangle \mapsto \langle M, \psi \mathbin{;} \nu \rangle$$

Proposition. The satisfaction of graph-transformation sentences is invariant with respect to substitutions.

A substitution system for graph transformation (continued)

- Sentences (over signature of variables X) are translated along substitutions $\psi\colon X \to Y$ by means of dedicated pushouts:

$$
\begin{array}{ccccc}
L & \xleftarrow{\;l_X\;} & X & \xrightarrow{\;r_X\;} & R \\
\downarrow & \mathbb{R} & \downarrow{\scriptstyle \psi} & \mathbb{R} & \downarrow \\
G & \xleftarrow[\;l_Y\;]{} & Y & \xrightarrow[\;r_Y\;]{} & H
\end{array}
$$

- The reduction of valuations is defined by precomposition with ψ:

$$\langle M, \nu \rangle \mapsto \langle M, \psi \mathbin{;} \nu \rangle$$

Proposition. The satisfaction of graph-transformation sentences is invariant with respect to substitutions.

A substitution system for graph transformation (continued)

- Sentences (over signature of variables X) are translated along substitutions $\psi \colon X \to Y$ by means of dedicated pushouts:

$$\begin{array}{ccccc} L & \xleftarrow{\ l_X\ } & X & \xrightarrow{\ r_X\ } & R \\ \downarrow & \mathbb{R} & \downarrow{\scriptstyle \psi} & \mathbb{R} & \downarrow \\ G & \xleftarrow[\ l_Y\ ]{} & Y & \xrightarrow[\ r_Y\ ]{} & H \end{array}$$

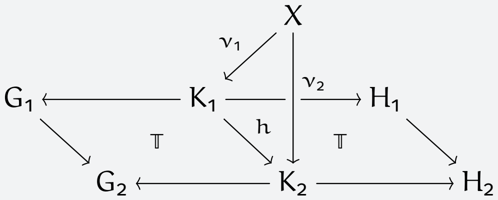
- The reduction of valuations is defined by precomposition with ψ:

$$\langle M, \nu \rangle \mapsto \langle M, \psi \mathbin{;} \nu \rangle$$

Proposition. The satisfaction of graph-transformation sentences is invariant with respect to substitutions.

Graph transformation as logic programming

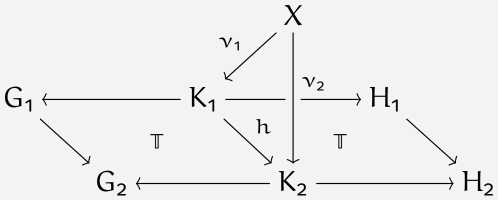
- A homomorphism between two valuations $\langle M_1, \nu_1 \colon X \to K_1 \rangle$ and $\langle M_2, \nu_2 \colon X \to K_2 \rangle$ is a morphism $h \colon K_1 \to K_2$ such that



Proposition. For any sentence $l \to r$ over X and any homomorphism $h \colon \langle M_1, \nu_1 \rangle \to \langle M_2, \nu_2 \rangle$ such that $\langle M_1, \nu_1 \rangle \vDash l \to r$, $\langle M_2, \nu_2 \rangle \vDash l \to r$.

Graph transformation as logic programming

- A homomorphism between two valuations $\langle M_1, \nu_1 \colon X \to K_1 \rangle$ and $\langle M_2, \nu_2 \colon X \to K_2 \rangle$ is a morphism $h \colon K_1 \to K_2$ such that



Proposition. For any sentence $l \to r$ over X and any homomorphism $h \colon \langle M_1, \nu_1 \rangle \to \langle M_2, \nu_2 \rangle$ such that $\langle M_1, \nu_1 \rangle \models l \to r$, $\langle M_2, \nu_2 \rangle \models l \to r$.

Graph transformation as logic programming (continued)

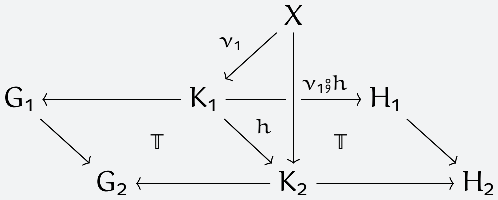
Theorem. Every set Γ of universally quantified GT sentences has an initial model O_Γ: the broad subcategory of $\mathbb{T}\mathrm{rans}(\mathbb{C})$ generated by those transitions that witness the application of a rule from Γ.

Proposition. The model O_Γ is X-reachable for every $X \in |\mathbb{S}\mathrm{ubst}|$: for every valuation $\langle \mathrm{O}_\Gamma, \nu_1 \rangle$ of X, every pair of morphisms $\mathrm{O}_\Gamma \to M$ and $h \colon K_1 \to K_2$ can be lifted to a homomorphism $\langle \mathrm{O}_\Gamma, \nu_1 \rangle \to \langle M, \nu_1 ; h \rangle$.

Graph transformation as logic programming (continued)

Theorem. Every set Γ of universally quantified GT sentences has an initial model O_Γ: the broad subcategory of $\mathbb{T}\mathrm{rans}(\mathbb{C})$ generated by those transitions that witness the application of a rule from Γ.

Proposition. The model O_Γ is X-reachable for every $X \in |\mathbb{S}\mathrm{ubst}|$: for every valuation $\langle O_\Gamma, \nu_1 \rangle$ of X, every pair of morphisms $O_\Gamma \to M$ and $h\colon K_1 \to K_2$ can be lifted to a homomorphism $\langle O_\Gamma, \nu_1 \rangle \to \langle M, \nu_1 \fatsemi h \rangle$.



Graph transformation as logic programming (conclusion)

Theorem (Herbrand's theorem). For every graph-transformation logic program Γ and every query $\exists X \cdot l \to r$ such that

- the satisfaction of $l \to r$ is preserved by X-homomorphisms,
- Γ has an X-reachable initial model 0_Γ,

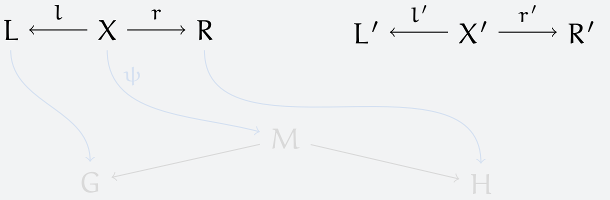
the following statements are equivalent:

1. $\Gamma \vDash \exists X \cdot l \to r$.
2. $0_\Gamma \vDash \exists X \cdot l \to r$.
3. $\exists X \cdot l \to r$ admits a Γ-solution.

Graph rewriting as (abstract) resolution

$\exists X \cdot l \to r$

$\forall X' \cdot l' \to r'$

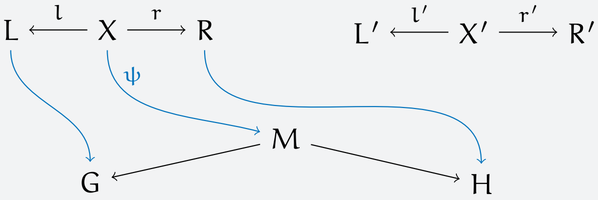


Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Graph rewriting as (abstract) resolution

$\exists X \cdot l \to r$

$\forall X' \cdot l' \to r'$



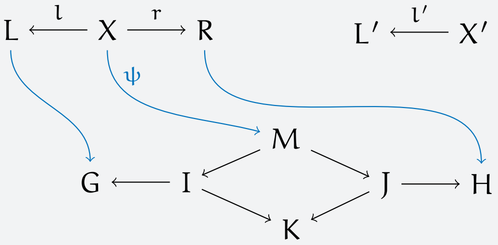
Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Graph rewriting as (abstract) resolution

$\exists X \cdot l \to r$

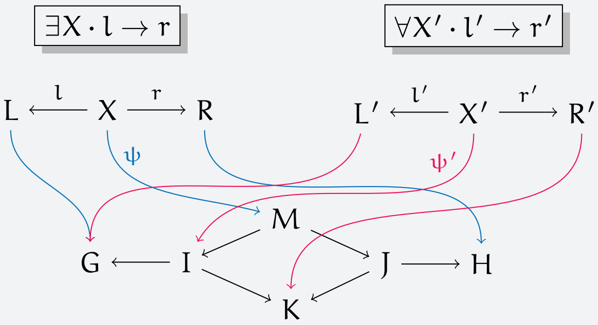
$\forall X' \cdot l' \to r'$

$$L' \xleftarrow{l'} X' \xrightarrow{r'} R'$$



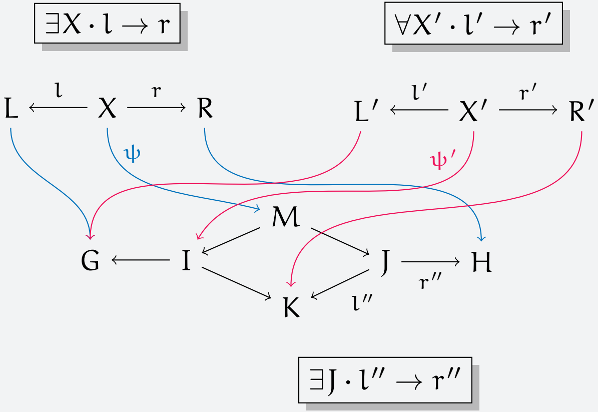
Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Graph rewriting as (abstract) resolution



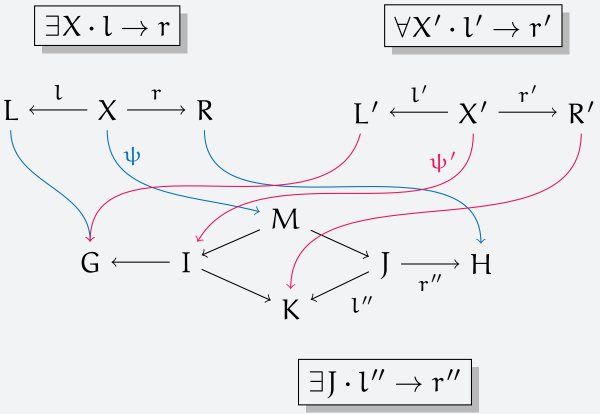
Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Graph rewriting as (abstract) resolution



Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Graph rewriting as (abstract) resolution



Proposition. Resolution is sound and complete with respect to the denotational semantics of the graph-transformation subst. system.

Bridging the gap between GT and LP: Conclusions

A substitution-system-based approach

rules .... sentences
graph transition systems .... models
witnessing rule application .... satisfaction
graph rewriting .... resolution

Further challenges

- a concrete operational semantics of GT LP
- accommodating GT systems with (negative) application conditions

Thank you!

Further Reading

- Reiko Heckel, Hartmut Ehrig, Uwe Wolter, Andrea Corradini.
  Double-Pullback Transitions and Coalgebraic Loose Semantics for Graph Transformation Systems.
  Applied Categorical Structures 9(1): 83-110 (2001)
- Stephen Lack and Pawel Sobocinski.
  Adhesive and quasiadhesive categories.
  ITA 39(3): 511-545 (2005)
- Hartmut Ehrig, Julia Padberg, Ulrike Prange, Annegret Habel.
  Adhesive High-Level Replacement Systems: A New Categorical Framework for Graph Transformation.
  Fundam. Inform. 74(1): 1-29 (2006)
- Andrea Corradini, Fabio Gadducci, Leila Ribeiro.
  An Institution for Graph Transformation.
  WADT 2010: 160-174
- Ionuţ Ţuţu and José L. Fiadeiro.
  From conventional to institution-independent logic programming.
  JLC (2017)