Encoding Graph Transformation in Linear Logic

Paolo Torrini
joint work with Reiko Heckel
pt95@mcs.le.ac.uk

University of Leicester

Graph Transformation

- Graph Transformation Systems (GTS) — high-level approach to system modelling, UML, model-driven development, stochastic simulation
- Existing formalisations — algebraic-categorical (SPO, DPO), 2nd-order predicate logic
- High-level character, strong mathematical foundation
- Double-pushout (DPO) — mature approach, based on category theory

Encoding GT in LL — why?

- LL close to process algebras (Abramsky, Pfenning, Cervesato)
- Parallel composition ($\alpha \otimes \beta$), choice ($\alpha \& \beta$), reachability ($\vdash \alpha \multimap \beta$), replication (!)
- Semantic motivation: taking closer graph transformation and process algebra
- Existing approach: hyperedge replacement
- What we do: logic-based hyperedge replacement
- Practical motivation: making proofs about GTS easier

Typed hypergraphs

- Hypergraph $G = \langle V, E, \mathsf{s} \rangle$
  V set of nodes, E set of hyperedges
  assignment $\mathsf{s} : E \to V^*$
- H-graph morphism — $\langle \phi_V : V_1 \to V_2,\ \phi_E : E_1 \to E_2 \rangle$
  assignment-preserving
- Type h-graph $TG = \langle \mathcal{V}, \mathcal{E}, \mathsf{ar} \rangle$
  $\mathcal{V}$ set of node types, $\mathcal{E}$ set of h-edge types
  $\mathsf{ar}(l) : \mathcal{E} \to \mathcal{V}^*$
- TG-typed h-graph (G, t), with $\mathsf{t} : G \to TG$
- TG-typed h-graph morphism $f : (G_1, \mathsf{t}_1) \to (G_2, \mathsf{t}_2)$
  is h-morphism $f : G_1 \to G_2$ with $\mathsf{t}_2 \circ f = \mathsf{t}_1$

DPO diagram

- Graph transformation rule $p : L \xleftarrow{l} K \xrightarrow{r} R$
  span of typed h-graph morhisms (l, r),
  K interface, L/K to be deleted, R/K to be created,
  rule application determined by match morphism m,
  m determined up to iso by interface morphism d
- DPO conditions — (1) Identification condition:
  (a) m never identifies distinct L/K elements
  (b) m never identifies L/K elements with K ones
  (2) Dangling condition: for each node $n \in L/K$, all
  edges connected to n are in L/K, too

$$\begin{tikzcd}
L \arrow[d, "m"'] \arrow[dr, phantom, "(1)"] & K \arrow[l, "l"'] \arrow[r, "r"] \arrow[d, "d"] \arrow[dr, phantom, "(2)"] & R \arrow[d, "m^*"] \\
G & D \arrow[l, "g"] \arrow[r, "h"'] & H
\end{tikzcd}$$

Overall plan

- Algebraic characterisation of DPO-GTS — monoidal structure with restriction over node names (hyperedge replacement)
- edges as relations (predicates) over nodes, parallel composition (tensor), α-renaming of nodes (restriction, or *hiding*)
- Translation to a quantified extension of ILL
- maps graph algebraic components to derivations, hence proof terms and linear λ-calculus
- terms represent identity of nodes and edges
- formulas represent type and connectivity — enough for reasoning about graphs up to iso

QILL

- $\alpha = A \mid L(N_1, \ldots, N_n) \mid \mathbf{1} \mid \alpha_1 \otimes \alpha_2 \mid \alpha_1 \multimap \alpha_2 \mid !\alpha_1 \mid \alpha_1 \& \alpha_2 \mid \forall x : \beta.\alpha \mid \hat{\exists} x : \beta.\alpha \mid \alpha \downharpoonleft N \mid \alpha = \alpha$
- $M = x \mid p \mid u \mid$ nil $\mid N_1 \otimes N_2 \mid \lambda x.N \mid \hat{\lambda} u.N \mid N_1 \hat{\ } N_2 \mid N_1 N_2 \mid M \mid \langle N_1, N_2 \rangle \mid$ fst $N \mid$ snd N
- $\alpha \hat{\equiv} \beta \ =_{df} \ (\alpha \multimap \beta) \& (\beta \multimap \alpha)$
- Sequent calculus proof rules
- Double-entry sequents — linear premises (Δ) and non-linear ones (Γ, equivalent to $!\Gamma$)

$$\Gamma; \Delta \vdash N :: \alpha$$

Encoding graphs

- Graph constructors: edge predicates, Nil (empty graph), $\otimes$ (parallel composition), $\hat{\exists}$ (linearly resource-bound quantifier) to type restriction
- (closed) h-graph as (closed) formula

$$\hat{\exists}\overline{x : A}.\gamma$$

  $\overline{x : A}$ sequence of typed variables,
  either $\gamma = \mathbf{1}$ (empty graph) or $\gamma = L_1\ (\overline{x}_1) \otimes \ldots \otimes L_k\ (\overline{x}_k)$
  (a multiset of edge components)
- Adequate graph representation

Encoding rules (node only interfaces)

- $\multimap$ for transformation, $\forall$ for interface nodes
- Transformation rule (α, β graph formulas)

$$\forall \overline{x : A}.\alpha \multimap \beta$$

- rule application schema

$$\frac{\begin{array}{cc} \Gamma; \cdot \Vdash \alpha_G \hat{\equiv} \alpha_{G'} & \Gamma; \cdot \Vdash \alpha_H \hat{\equiv} \alpha_{H'} \\ \multicolumn{2}{l}{\alpha_{G'} = \hat{\exists} \overline{z : A}.\alpha_L[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C} \\ \multicolumn{2}{l}{\alpha_{H'} = \hat{\exists} \overline{z : A}.\alpha_R[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C} \end{array}}{\Gamma; \forall \overline{x : A}.\alpha_L \multimap \alpha_R \Vdash \alpha_G \multimap \alpha_H} \stackrel{p,m}{\Longrightarrow}$$

rules I

$$\frac{}{\Gamma; u :: \alpha \vdash u :: \alpha}\ Id \qquad \frac{}{\Gamma, x :: \alpha; \cdot \vdash x :: \alpha}\ UId$$

$$\frac{\Gamma; \Delta_1 \vdash M :: \alpha \quad \Gamma; \Delta_2 \vdash N :: \beta}{\Gamma; \Delta_1, \Delta_2 \vdash M \otimes N :: \alpha \otimes \beta}\ \otimes R$$

$$\frac{\Gamma; \Delta, u :: \alpha, v :: \beta \vdash N :: \gamma}{\Gamma; \Delta, w :: \alpha \otimes \beta \vdash \mathsf{let}\ u \otimes v = w\ \mathsf{in}\ N :: \gamma}\ \otimes L$$

$$\frac{\Gamma; \Delta, u :: \alpha \vdash M :: \beta}{\Gamma; \Delta \vdash \hat{\lambda} u : \alpha.\ M :: \alpha \multimap \beta}\ \multimap R$$

$$\frac{\Gamma; \Delta_1 \vdash N :: \alpha \quad \Gamma; \Delta_2, u :: \beta \vdash M :: \gamma}{\Gamma; \Delta_1, \Delta_2, w :: \alpha \multimap \beta \vdash \mathsf{let}\ u = w\hat{\ }N\ \mathsf{in}\ M :: \gamma}\ \multimap L$$

RBQ rules I

$$\frac{\begin{array}{ll}\Gamma; \Delta \vdash M :: \alpha[N_\Delta/x] & \Gamma; \cdot \vdash N_\Delta :: \beta \\ \Gamma; \Delta' \vdash n :: \beta\lfloor N_\Delta & \Gamma, x :: \beta; \cdot \vdash \mathsf{nil} :: (\alpha[N_\Delta/x])[x/N_\Delta] = \alpha\end{array}}{\Gamma; \Delta, \Delta' \vdash !N_\Delta \otimes n \otimes M :: \hat{\exists} x : \beta.\alpha}\ \hat{\exists}R$$

$$\frac{\Gamma, x :: \beta; \Delta, n :: \beta\lfloor x, v :: \alpha \vdash N :: \gamma}{\Gamma; \Delta, w :: \hat{\exists} x : \beta.\alpha \vdash N[w/(!x \otimes n \otimes v)] :: \gamma}\ \hat{\exists}L$$

RBQ right intro

$$\frac{\begin{array}{ll}\Gamma;\Delta \vdash M :: \alpha[N_\Delta/x] & \Gamma;\cdot \vdash N_\Delta :: \beta \\ \Gamma;\Delta' \vdash n :: \beta\downharpoonleft N_\Delta & \Gamma, x :: \beta;\cdot \vdash \mathsf{id}_\alpha :: (\alpha[N_\Delta/x])[x/N_\Delta] = \alpha\end{array}}{\Gamma;\Delta,\Delta' \vdash (!N_\Delta \otimes n) \otimes M :: \hat{\exists} x : \beta.\alpha}\ \hat{\exists} I$$

- standard hips. (1) $\alpha[N/x]$ graph with N in place of free x
  (2) N well-typed
- (3) there has to be a node (linear resource) named by N — $\downharpoonleft$ denotes naming reference to term
- (4) N does not occur free in α (unless $x = N$) — a freshness condition, expressed using type equality and substitution
- N (N_Δ) depends on the derivation of $\alpha[N_\Delta/x]$

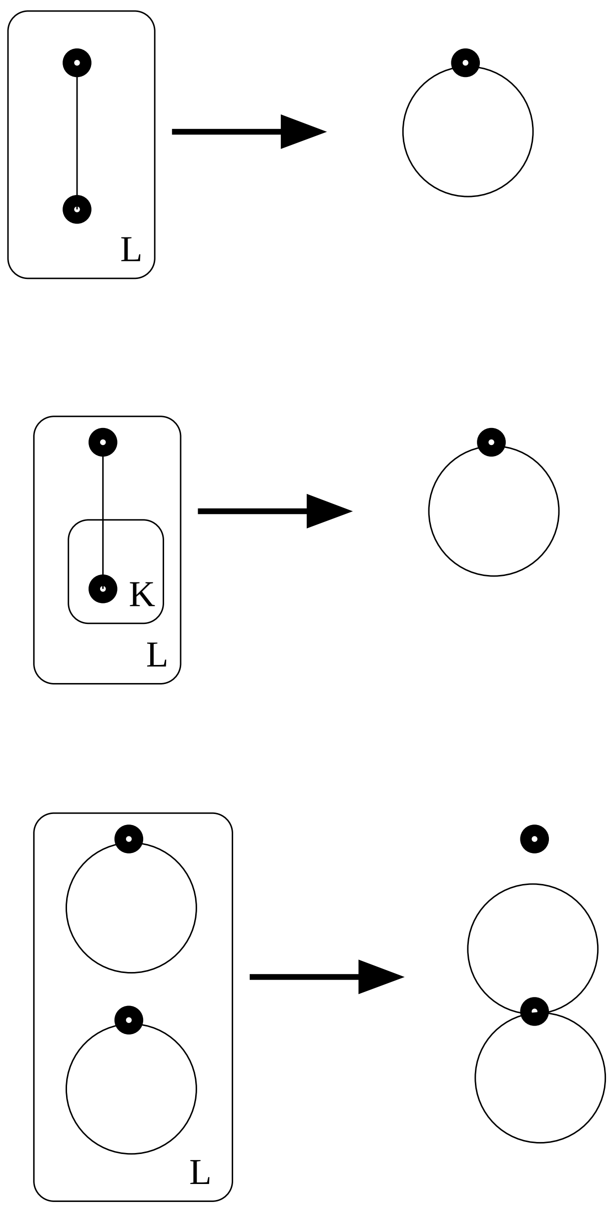
RBQ left intro

$$\frac{\Gamma, x :: \beta; \Delta, n :: \beta \downharpoonleft x, v :: \alpha \vdash N :: \gamma}{\Gamma; \Delta, w :: \hat{\exists} x : \beta.\alpha \vdash \text{let } (!x \otimes n \otimes v) = w \text{ in } N :: \gamma} \ \hat{\exists} L$$

- (almost) standard rule
- $\hat{\exists}$ can be used to type restriction in graph expressions
- $\downharpoonleft$ introduction by axiom

$$\frac{\Gamma; \cdot \vdash N :: \alpha}{\Gamma; n :: \alpha \downharpoonleft N \vdash n :: \alpha \downharpoonleft N} \ \downharpoonleft A$$

Incorrect matches



Quantifier and DPO conditions

- $\nvdash (\hat{\exists} x : \beta.\ \alpha(x,x)) \multimap \hat{\exists} xy : \beta.\ \alpha(x,y)$
  the resource for x cannot suffice for x and y.
- $\nvdash \forall x : \beta.\ \beta{\downharpoonleft}x \otimes \alpha(x,x) \multimap \hat{\exists} y : \beta.\alpha(y,x)$
  y and x should be instantiated with the same term — blocked by the freshness condition in $\hat{\exists}$ introduction
- $\nvdash (\hat{\exists} yx : \beta.\ \alpha_1(x) \otimes \alpha_2(x)) \multimap (\hat{\exists} x : \beta.\alpha_1(x)) \otimes \hat{\exists} x : \beta.\alpha_2(x)$
  the two bound variables in the consequence require distinct resources and refer to distinct occurrences

Examples I

$$\dfrac{x \vdash x \qquad \dfrac{}{x, \alpha(x,x) \vdash \hat{\exists} y.\alpha(x,y)}\ \hat{\exists}R\ \mathit{fails}}{\dfrac{x, \alpha(x,x) \vdash \hat{\exists} xy.\alpha(x,y)}{\hat{\exists} x.\alpha(x,x) \vdash \hat{\exists} xy.\alpha(x,y)}\ \hat{\exists}L}\ \hat{\exists}R$$

$$\dfrac{x \vdash x \qquad x, \alpha(x,x) \vdash \alpha(x/y, x)}{x, \alpha(x,x) \vdash \hat{\exists} y.\alpha(y,x)}\ \hat{\exists}R\ \mathit{fails}$$

$$\dfrac{\dfrac{\dfrac{\ldots}{x, \alpha(x) \vdash \hat{\exists} x.\alpha(x)}\ \hat{\exists}R \qquad \dfrac{}{y, \alpha(x) \vdash \hat{\exists} x.\alpha(x)}\ \hat{\exists}R\ \mathit{fails}}{x, y, \alpha(x), \alpha(x) \vdash (\hat{\exists} x.\alpha(x)) \otimes \hat{\exists} x.\alpha(x)}\ \otimes R}{\hat{\exists} xy.\alpha(x) \otimes \alpha(x) \vdash (\hat{\exists} x.\alpha(x)) \otimes \hat{\exists} x.\alpha(x)}\ \hat{\exists}L(2)$$

Examples II — problem with Cut

$$\dfrac{\dfrac{\ldots}{\Gamma; \alpha \downharpoonleft x, \alpha \vdash \hat{\exists} x.\alpha(x)} \quad \dfrac{\ldots}{\Gamma; \alpha \downharpoonleft x, \alpha \vdash \hat{\exists} x.\alpha(x)}}{\Gamma; \alpha \downharpoonleft x, \alpha \downharpoonleft x, \alpha, \alpha \vdash (\hat{\exists} x.\alpha(x)) \otimes \hat{\exists} x.\alpha(x)} \otimes R$$

$$\Gamma; (\hat{\exists} x.\alpha(x)) \otimes \hat{\exists} x.\alpha(x) \vdash \hat{\exists} xy.\alpha(x) \otimes \alpha(y)$$

$$\Gamma; \alpha \downharpoonleft x, \alpha \downharpoonleft x, \alpha, \alpha \nvdash \hat{\exists} xy.\alpha(x) \otimes \alpha(y)$$

Solution:

$$\dfrac{\dfrac{\ldots}{\Gamma_1; \alpha \downharpoonleft x, \alpha \vdash \hat{\exists} x.\alpha(x)} \quad \dfrac{\ldots}{\Gamma_2; \alpha \downharpoonleft x, \alpha \vdash \hat{\exists} x.\alpha(x)}}{\Gamma_1, \Gamma_2; \alpha \downharpoonleft x, \alpha \downharpoonleft y, \alpha, \alpha[y/x] \vdash (\hat{\exists} x.\alpha(x)) \otimes \hat{\exists} x.\alpha(x)} \otimes R$$

rules II

$$\frac{}{\Gamma; u :: \alpha \vdash u :: \alpha}\ Id \qquad \frac{}{\Gamma, x :: \alpha; \cdot \vdash x :: \alpha}\ UId$$

$$\frac{\Gamma_1; \Delta_1 \vdash M :: \alpha \quad \Gamma_2; \Delta_2 \vdash N :: \beta}{\Gamma_1, \Gamma_2; \Delta_1, \Delta_2 \vdash M \otimes N :: \alpha \otimes \beta}\ \otimes R$$

$$\frac{\Gamma; \Delta, u :: \alpha, v :: \beta \vdash N :: \gamma}{\Gamma; \Delta, w :: \alpha \otimes \beta \vdash \mathsf{let}\ u \otimes v = w\ \mathsf{in}\ N :: \gamma}\ \otimes L$$

$$\frac{\Gamma; \Delta, u :: \alpha \vdash M :: \beta}{\Gamma; \Delta \vdash \hat{\lambda} u : \alpha.\ M :: \alpha \multimap \beta}\ \multimap R$$

$$\frac{\Gamma_1; \Delta_1 \vdash N :: \alpha \quad \Gamma_2; \Delta_2, u :: \beta \vdash M :: \gamma}{\Gamma_1, \Gamma_2; \Delta_1, \Delta_2, w :: \alpha \multimap \beta \vdash \mathsf{let}\ u = w\hat{\ }N\ \mathsf{in}\ M :: \gamma}\ \multimap L$$

RBQ rules II

$$\frac{\begin{array}{ll}\Gamma; \Delta \vdash M :: \alpha[N/x] & \Gamma; \cdot \vdash N :: \beta \\ \Gamma; \Delta' \vdash n :: \beta\!\downharpoonleft & \\ \Gamma, x :: \beta; \cdot \vdash \mathsf{nil} :: (\alpha[N/x])[x/N] = \alpha & \end{array}}{\Gamma, x :: \beta; \Delta, \Delta' \vdash !N \otimes n \otimes M :: \hat{\exists} x : \beta.\alpha}\ \hat{\exists}R$$

$$\frac{\Gamma, x :: \beta; \Delta, n :: \beta\!\downharpoonleft, v :: \alpha \vdash N :: \gamma}{\Gamma; \Delta, w :: \hat{\exists} x : \beta.\alpha \vdash N[w/(!x \otimes n \otimes v)] :: \gamma}\ \hat{\exists}L$$

Reachability

- Tansformation — G_0, G_1 closed h-graphs, G_0 initial, $P_1, \ldots, P_k$ rules
  - G_1 reachable from by some application of the rules

$$!P_1, \ldots, !P_k, G_0 \vdash G_1$$

  - G_1 reachable by applying each rule once

$$P_1, \ldots, P_k, G_0 \vdash G_1$$

- Translation complete with respect to reachability (sequent provable if graph reachable)
- Soundness — work in progress,
  general idea — logically valid implications are "read-only" transformations

Conclusion and further work

- Proof theory-driven approach to GT
- uses resource logic
- new quantifier to deal with restriction
- Extension to generalised interfaces ($\hat{\forall}$)
- two-level embedding approach
- Interest in mechanised theorem proving

Other rules

$$\frac{}{\Gamma; \cdot \vdash \mathsf{nil} :: \mathbf{1}}\ \mathbf{1}I \qquad \frac{\Gamma; \Delta \vdash M :: \mathbf{1} \quad \Gamma; \Delta' \vdash N :: \alpha}{\Gamma; \Delta, \Delta' \vdash \mathsf{let\ nil} = M\ \mathsf{in}\ N :: \alpha}\ \mathbf{1}E$$

$$\frac{\Gamma; \Delta \vdash M :: \alpha \quad \Gamma; \Delta \vdash N :: \beta}{\Gamma; \Delta \vdash \langle M, N \rangle :: \alpha \& \beta}\ \& I$$

$$\frac{\Gamma; \Delta \vdash M :: \alpha \& \beta}{\Gamma; \Delta \vdash \mathsf{fst}\ M :: \alpha}\ \& E1 \qquad \frac{\Gamma; \Delta \vdash M :: \alpha \& \beta}{\Gamma; \Delta \vdash \mathsf{snd}\ M :: \beta}\ \& E2$$

$$\frac{\Gamma; \cdot \vdash M :: \alpha}{\Gamma; \cdot \vdash !M :: !\alpha}\ !I \qquad \frac{\Gamma; \Delta_1 \vdash M :: !\alpha \quad \Gamma, p :: \alpha; \Delta_2 \vdash N :: \beta}{\Gamma; \Delta_1, \Delta_2 \vdash \mathsf{let}\ p = M\ \mathsf{in}\ N :: \beta}\ !E$$

$$\frac{\Gamma, x :: \beta; \Delta \vdash M :: \alpha}{\Gamma; \Delta \vdash \lambda x.\ M :: \forall x : \beta.\ \alpha}\ \forall I \qquad \frac{\Gamma; \Delta \vdash M :: \forall x : \beta.\ \alpha \quad \Gamma; \cdot \vdash N :: \beta}{\Gamma; \Delta \vdash MN :: \alpha[N/x]}\ \forall E$$

Encoding rules II

- $\multimap$ for transformation, $\forall$ (first order, standard) for interface nodes, $\hat{\forall}$ (second order, non-linearly resource bound) for interface edges
- Transformation rule as closed formula

$$\forall \overline{x : A}.\hat{\forall}\overline{y : \gamma_i}.\alpha \multimap \beta$$

  with α, β, γ_i graph formulas
- applying a rule gives a transformation where α is deleted (consumed), β is created, after instantiating first nodes $\overline{x}$ and then edges $\overline{y}$ — the rule interface

Rule application schema

$$\frac{\begin{array}{ll}\Gamma; \cdot \Vdash \alpha_G \hat{\equiv} \alpha_{G'} & \Gamma; \cdot \Vdash \alpha_H \hat{\equiv} \alpha_{H'} \\ \multicolumn{2}{l}{\alpha_{G'} = \hat{\exists}\overline{z : A}.(\alpha_L \otimes (\bigotimes[\overline{\gamma_i} \xleftarrow{d_e} \overline{v : \gamma_i}]))[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C} \\ \multicolumn{2}{l}{\alpha_{H'} = \hat{\exists}\overline{z : A}.(\alpha_R \otimes (\bigotimes[\overline{\gamma_i} \xleftarrow{d_e} \overline{v : \gamma_i}]))[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C}\end{array}}{\Gamma; \forall\overline{x : A}.\hat{\forall}\overline{v : \gamma_i}.\alpha_L \multimap \alpha_R \Vdash \alpha_G \multimap \alpha_H} \stackrel{p,m}{\Longrightarrow}$$

For rules with node-only interfaces

$$\frac{\begin{array}{ll}\Gamma; \cdot \Vdash \alpha_G \hat{\equiv} \alpha_{G'} & \Gamma; \cdot \Vdash \alpha_H \hat{\equiv} \alpha_{H'} \\ \multicolumn{2}{l}{\alpha_{G'} = \hat{\exists}\overline{z : A}.\alpha_L[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C} \\ \multicolumn{2}{l}{\alpha_{H'} = \hat{\exists}\overline{z : A}.\alpha_R[\overline{z : A} \xleftarrow{d_n} \overline{x : A}] \otimes \alpha_C}\end{array}}{\Gamma; \forall\overline{x : A}.\alpha_L \multimap \alpha_R \Vdash \alpha_G \multimap \alpha_H} \stackrel{p,m}{\Longrightarrow}$$

Translation — I

Constituents

$$
\begin{array}{l}
[\![e_i(m,\ldots,n) : L_i(A_m,\ldots,A_n)]\!] \;=_{df}\; Id\ [\Gamma;;\quad c_i :: L_i(x_m,\ldots,x_n)] \\
[\![\mathsf{Nil}]\!] \;=_{df}\; \mathbf{1}I\ [\Gamma] \\
[\![M \parallel N]\!] \;=_{df}\; \otimes I\ [[\![M]\!];;\quad [\![N]\!]] \\
[\![\nu n : A.N]\!] \;=_{df}\; \hat{\exists} I\ [[\![N]\!];; \\
\quad UId\ [\Gamma;;\ x_n :: A];; \\
\qquad\qquad Id\ [\Gamma;;\ n :: A\downharpoonleft x_n];; \\
\qquad\qquad \Gamma, y :: A; \cdot \vdash \mathsf{id} : MainType([\![N]\!])[y/x_n]\#(y, x_n)]
\end{array}
$$

Translation — II

Graph interfaces

$$
\begin{aligned}
[\![n : A]\!] &=_{df} Id\ [\Gamma, x :: A;;\quad n :: A \downharpoonleft x] \\
[\![\{n : A\}]\!] &=_{df} [\![n : A]\!] \\
[\![\{n_1 : A_1\} \cup X]\!] &=_{df} \otimes I\ [[\![\{n_1 : A_1\}]\!];;\ [\![X]\!]]
\end{aligned}
$$

Graph expressions

$$
[\![X \vDash C]\!] =_{df} \otimes I\ [[\![X]\!]_I;;\ [\![C]\!]]
$$

Graph derivations

- *graph formulas* — $\mathbf{1}, \otimes, \hat{\exists}, \downharpoonleft$ fragment of the logic containing only primitive graph types (node and edge types)
- *graph context* — multiset of typed nodes and typed edge components.
- *graph derivation* — derivable sequent $\Gamma; \Delta \vdash N :: \gamma$, where γ is a graph formula, Δ is a graph context, Γ the environment, N a normal derivation.
- Uses only axioms and the introduction rules $\mathbf{1}I$, $\otimes I$, $\hat{\exists}I$.

Quantifier and congruence

$\hat{\exists}$ satisfies properties of renaming, exchange and distribution over $\otimes$

- $\vdash (\hat{\exists} x : \alpha.\beta(x)) \mathrel{\hat{\equiv}} (\hat{\exists} y : \alpha.\beta(y))$
- $\vdash (\hat{\exists} xy : \alpha.\gamma) \mathrel{\hat{\equiv}} (\hat{\exists} yx : \alpha.\gamma)$
- $\vdash (\hat{\exists} x : \alpha.\beta \otimes \gamma(x)) \mathrel{\hat{\equiv}} (\beta \otimes \hat{\exists} x : \alpha.\gamma(x))$ (x not in α)

Equivalence between α and $\hat{\exists} x.\ \alpha$ generally fails in both directions, even when x does not occur free in α

Graphs and types — adequacy

- Isomorphism between graph expressions and graph derivations
- Isomorphism between graphs (congruence classes of graph expressions) and graph formulas modulo linear equivalence
- Curry-Howard style correspondence
- Possibility to implement hypergraphs and to reason about them

Graph transformation

- Less interested in component identity, higher-level translation, based on logic formulas
- Linear implication as transformation
- Standard quantifier for interface nodes
- Rule names as non-linear resources (unlimited application)

$$[\![M \Longrightarrow N]\!]^T =_{df} [\![M]\!]^T \multimap [\![N]\!]^T$$
$$[\![\Lambda x : A.N]\!]^T =_{df} \forall x : A.[\![N]\!]^T$$

$$[\![\pi(p)]\!] \ =_{df} \ FId\ [\Gamma;; \quad p :: \forall \overline{x : A_x}.[\![L]\!]^T \multimap [\![R]\!]^T]$$

Completeness and soundness

- Let $\Gamma_P = \Sigma \cup [\rho | \rho = [\![\pi(p)]\!]^T, p \in P]$, then for each reachable h-graph G

$$\Gamma_P; [\![G_0]\!]^T \vdash [\![G]\!]^T$$

- Let R be a multiset of transformations, $\Delta_R = [\tau | \tau = [\![t]\!]^T, t \in R]$, then for each h-graph G reachable from G_0 by executing R

$$\Sigma; [\![G_0]\!]^T, \Delta_R \vdash [\![G]\!]^T$$

- This is for completeness
- Soundness requires more work on the interpretation of linear implication