Graph Transformation with Topological Constraints

Paolo Torrini, Reiko Heckel

pt95, reiko@mcs.le.ac.uk

WADT 2010
Graph transformation

- GT: declarative modelling technique that extends term rewriting
- states as graphs, transitions as application of transformation rules (transformation steps), indeterministic character
- structured graphs
- use of preorders and derived topologies to define higher-level views of graphs
Algebraic GT

category of graphs — sets of edges and vertexes with source and target functions

transformation rule $L \implies R$:
  - partial injective morphism — single-pushout (SPO)
  - spans of total injective morphisms (from an interface) — double-pushout (DPO)

graph $G$, rule match $m : L \to G$:
  - total morphism that does not identify preserved elements and deleted ones
  - extra requirement (DPO): leaves no dangling edges
Rules and actions

- suggested intuition: rules as actions, rule application as action event, led by an agent, in a context

- dangling condition (DPO): all the effects of the transformation step are included in the match (modularity wrt rule application)

- interface — subgraph of the match that is preserved

- locality of formulation (SPO): the specification of an action event depends on the agent, not on the context

- boundary — includes the edges that may be deleted as side-effect
Distributed systems

- DS: structure made of multiple components and connections
- GT approach based on two-layer hierarchy (local/global) and DPO (Taentzer et al.)
  - global network architecture and reconfiguration
  - local components and their behaviour
  - interfaces between components
Loose distributed approach

- network architecture — multiple levels of connections
- architecture preservation — graphs with constraints
  - local: disabling rule matches
  - global: invalidating transformation steps
- simple representation: transitive associations — a preorder
- looser than a hierarchy (e.g. containment, a poset)
LD systems

- behaviour modelled using SPO — dangling condition dropped locally
- network architecture as preorder ofplaces
- underlying graph elements mapped to places
- preorder-derived network topology
- region of application, implied by the topology, including the boundary — modularity recovered at the architectural level, in a looser sense wrt DPO
summarising

- SPO rule: specification of the action of an agent
- region of application: affected subgraph
- association as abstract connection relation
- modelling of granular systems
- possible use: high-level search for connection paths by pattern-matching
SPO approach, initial graph (with two components) and transformation rule

two applications of the rule, one to each component, may lead to disconnected components

different ways to rule disconnection out — e.g. by negative application condition

here we look at a direct one — an invariant stating that the two components stay connected
Spatial graphs

- structured graph $G$ (decoupled presentation)
- underlying graph $U_G$
- network architecture (spatial type graph): distinguished directed graph $P_G$ in which nodes are places and edges express association.
  stronger requirement (proper hierarchy): $P_G$ is acyclic (a containment hierarchy)
- total surjective location map ($loc$) from elements of $U_G$ to places, represented as coupling edges
- coupling as containment, no empty places
- algebraic characterisation
Location

- minimal requirements on $P_G$
  - refl-trans closure of association gives $\leq$ as a preorder
  - $\text{loc}$ is a function
  - for each edge $e$ between $n_1, n_2$ in $U_G$, it holds $\text{loc}(n_1), \text{loc}(n_2) \leq \text{loc}(e)$
  - for each $n_3$ s.t. $\text{loc}(n_3) \leq \text{loc}(e)$, either $\text{loc}(n_3) \leq \text{loc}(n_1)$ or $\text{loc}(n_3) \leq \text{loc}(n_2)$
- stronger requirements:
  - $\leq$ is a poset
  - there is a (unique) prime join in the preorder for each pair of places that are connected by an edge
Places and components

- $\leq$ induces a subcomponent relation
- can be used to express pre/post-conditions in rules (using non-emptiness)
- $\text{ctl}(p) = \{x : \text{loc}(x) = p\}$ is not generally a subgraph (e.g. an edge)
- hierarchy of components (vertical)
- given $p_1 < p_2$, then $\text{ctl}(p_2)$ is a subcomponent of $\text{ctl}(p_1)$
- connections (horizontal)
- given $p_1 \leq q, p_2 \leq q$, then $\text{ctl}(q)$ is a connection (common connecting subcomponent) between $\text{ctl}(p_1)$ and $\text{ctl}(p_2)$
shapes

A and B are associated to the same component

E is embedded in A

E is independent

A and B are associated to the same component

A is in the interface of the component associated to B
A and B are associated to components that share a connector

A and B are in mirror components
Derived topology

≤ is the specialisation preorder of an Alexandroff topology

upper-closed sets of places are the open sets
downward-closed sets of places are the closed sets

the topology of the network (abstract regions) can be mapped back to the underlying graph (concrete regions)
closed concrete regions are subgraphs
closed view — gluing aspect

open concrete regions are complements of subgraphs
open view — connecting aspect
Regions and matches

- each rule match gives an open abstract region
  \[ \text{reg}(m) = \bigcup_{x \in m} \{ p : \text{loc}(x) \leq p \} \] (subbase)

- the open concrete region \( \text{ctl}(m) = \{ x : \text{loc}(x) \in \text{reg}(m) \} \)
  includes all the dangling edges

- the closed concrete region \( \text{proj}(m) \) (closure of \( \text{ctl}(m) \))
  — the smallest subgraph wrt the topology that includes all the side-effects (the boundary)
examples: incomplete view

(Left) component $A$ connected to $B$ through $C$
(Right) component $A$ connected to $B$ that includes $C$
boxes as places, inclusion as reverse order
examples: open view

- prime join added, map extended to edges
- boxes as upper-closed sets (open sets)
examples: closed view

boxes as downward-closed sets (closed sets)
examples: closure view

boxes as non-point-like closed sets
Relations over regions

- relations between regions to express constraints on the model
  - inclusion between abstract regions: abstract subcomponent relation
  - intersection of open abstract regions (hence of concrete regions) is not empty: components are connected — they share a connector, e.g. a boundary edge
  - previous one false, intersection of closure of open abstract regions is not empty: components are adjacent — there is a component (concrete) that connects with both
GT example - revised

- initial graph with closed view and transformation rule
- prime join added to the original hierarchy
- mapping extended to edges
- constraint as rule invariant (places are non-empty)
SPO and DPO

from a formal point of view

each match can be completed with an open abstract region

therefore with a concrete region, which includes all the side effects

possible to extract a DPO rule (minimal wrt the topology) that corresponds to the application of the SPO one — the closure region gives its match

this can be shown diagrammatically
SPO diagram

- SPO rule application diagram with hierarchy
- pushouts 2, \([L, R, G, H]\)
- pullbacks 1,3,4,5
Diagram scheme

derived rule

proj(L) K' proj(R)

original rule

L R

P L P R

P G P D P H

topological constraint

K reg(L) reg(R)

derived regions

dag transformation

graph transformation

SPO match and derived DPO rule
embedded SPO diagram

pullbacks 1–4, 7, 8, \([K, K', \text{reg}(L), \text{proj}(L)], [K, K', \text{reg}(R), \text{proj}(R)]\)

pushouts 5, 6, \([K', \text{proj}(L), D, G], [K', \text{proj}(R), D, H]\)
Limit examples

trivial underlying graph: a set of isolated nodes, connected by the network

trivial network:
- converting SPO matches into DPO ones:
  - let $loc$ defined as an injection on nodes, such that for each pair of distinct $n_1, n_2$, $loc(n_1)$ and $loc(n_2)$ are \(\leq\)-incomparable, and edges are mapped to prime joins
  - then for each SPO match, its closed concrete region is the smallest DPO one
  - in general, they are only the smallest wrt the topology
Conclusion

- possibility to extend notion of distributed graph transformation, allowing for more levels
- SPO with a weak notion of DPO-style compositionality
- local dangling condition dropped to allow for simpler rules
- network architecture, global dangling condition
- algebraic characterisation
- work to be done on the modelling as well as on the formalisation