A Categorical Model of Computation for Graph Transformation

True Concurrency and Logic

Reiko Heckel, University of Leicester

IT University of Copenhagen, 1 December 2006

Logic from 1st Principles

Reasoning on the behaviour of DPO graph transformations

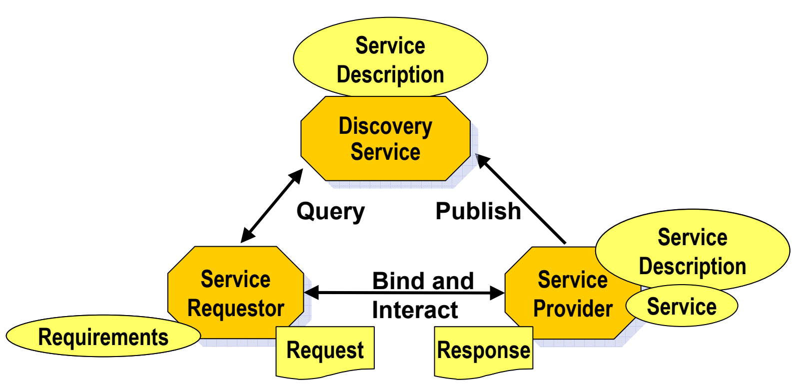
- in the 'graphs as objects' view
- at the level of abstraction of concurrent derivations (up to shift-equivalence)
- combining temporal and spatial aspects
- supporting reasoning on graph programs, their preconditions, actions, and effects on the state
- retaining the graphical (visual) appearance of GT

Approach: derive logic from the inherent structure of graphs and graph transformations

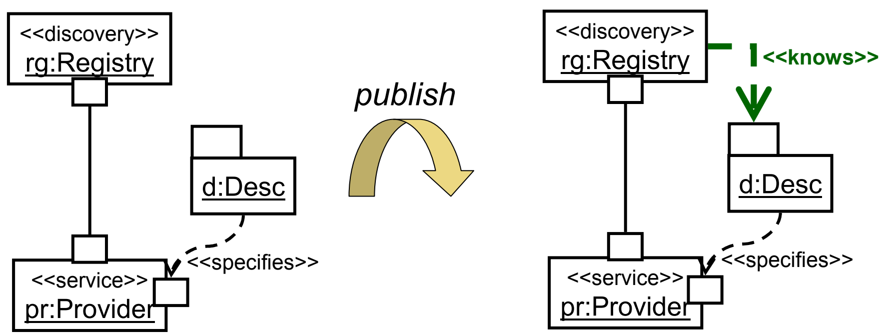
Outline

- an example
- an algebraic model of concurrent derivations
- an arrow logic with a spatial twist
- some open questions

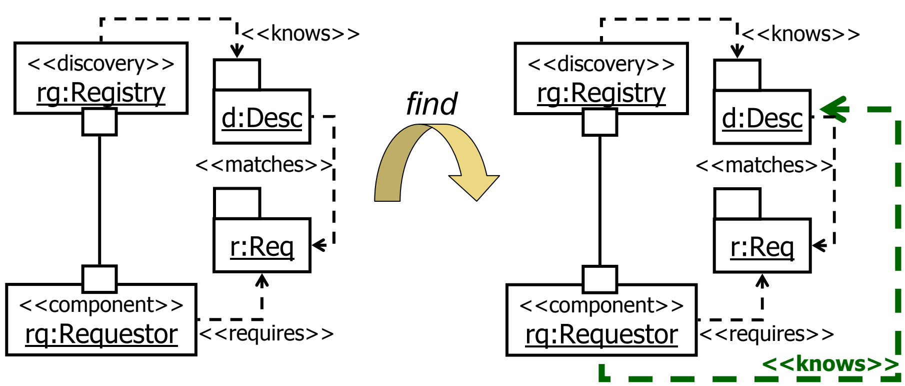
Example: Rules of Service-Oriented Architecture



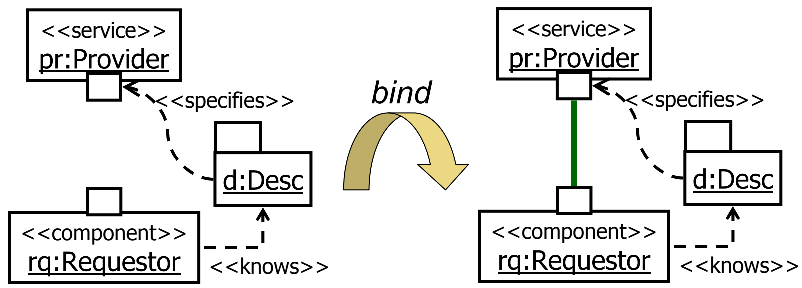
Rules of SOA: Publishing a service description



Rules of SOA: Discovering a service descriptions



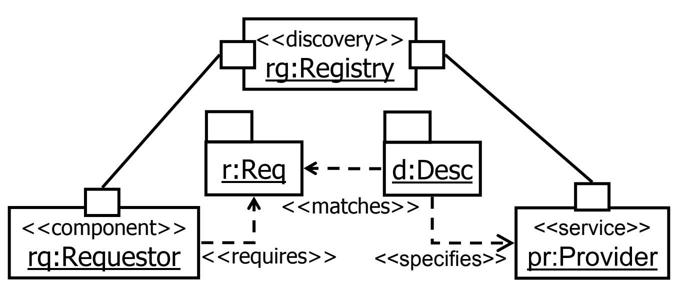
Rules of SOA: Binding to a service



Want to say things like

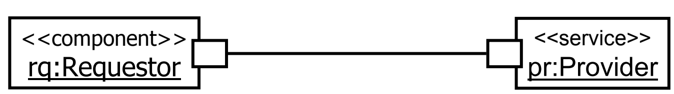
- In a graph containing P, publish, find, and bind are applicable.

P:



- In the resulting graph there is a connection between rq and pr.

Q:



Outline

- ✓ an example
- ✖ an algebraic model of concurrent derivations
- ✖ an arrow logic with a spatial twist
- ✖ some open questions

Structured Categories of Computations

Characteristics

- ✖ parametric in the alg. structure of states
- ✖ universal construction of cats of computations
- ✖ true concurrency for free

Four Steps

- ✖ states
- ✖ rules
- ✖ transitions
- ✖ computations

$$\textit{Programs} \underset{U}{\overset{TS}{\rightleftarrows}} \textit{STrans} \underset{V}{\overset{CM}{\rightleftarrows}} \textit{Models}$$

Examples

PT Nets: monoidal cats
Term Rewriting: cartesian 2-cats
Graph Transformation: double cats with finite horizontal colimits

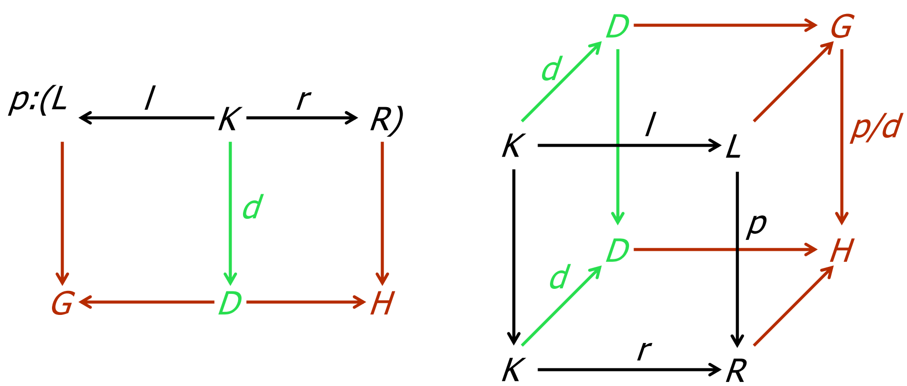
Place-Transition Petri Nets

	concrete	abstract (algebraic)
states	multisets of tokens	free commutative monoid P^{CM}
rules	transitions	edges of heterogeneous graph over nodes in P^{CM}
transitions	parallel steps	free monoidal reflexive graph
computations	net processes	free monoidal category

DPO Graph Transformation

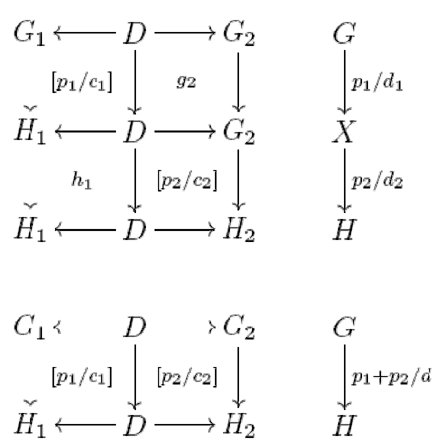
	concrete	abstract (categorical)
states	graphs	free cat with finite colimits
rules	DPO rules	basic double cells
transitions	parallel DPO steps	free horizontal category with finite colimits
computations	net processes	free double category with horizontal finite colimits

DPO Step in FHC Double Category



Theorem

- Let C be the set of "rule squares" for a GTS *G*.
- Then, there exists a pseudo-free FHC double category over C whose category of vertical arrows are the shift-equivalence classes of derivations in *G*.



Outline

- ✓ an example
- ✓ an algebraic model of concurrent derivations
- ✖ an arrow logic with a spatial twist
- ✖ some open questions

Spatial - Temporal Logic

spatial structure of states

- q: X → Q - graph patterns as state formulae S
- exists x . S - hiding
- S + T - gluing

rules

- p - application

temporal structure of computations

- pre F - precondition
- post F - postcondition
- idle S - id computation on state
- F ; G - sequential composition

Temporal Structure of Computations

(c, a) |= pre F , if there is c: g ➔ h
with (g, a) |= F

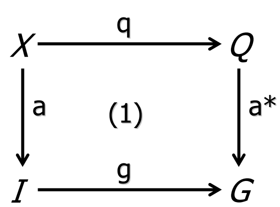
(c, a) |= post F, if there is c: s ➔ t
with (h, a) |= F

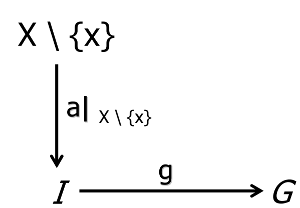
(c, a) |= idle S , if there is a state g
with (g, a) |= S and c = 1_g

(c, a) |= F_1 ; F_2 , if c = c_1 . c_2
with (c_1, a) |= F_1
and (c_2, a) |= F_2

Spatial Structure of States

(g, a) |= q: X → Q , if ex. a* s.t. (1) commutes

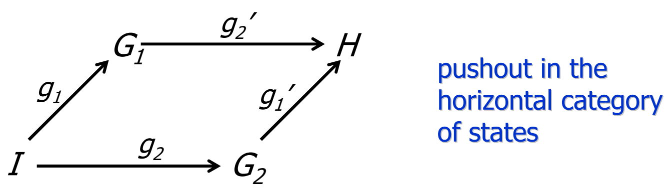




(g, a) |= exists x . S , if (g, a_x) |= S

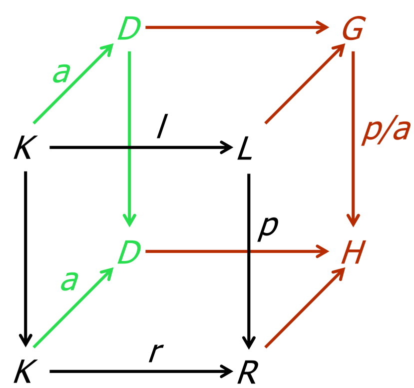
Spatial Structure of States

$(g, a) \models S_1 + S_2$, if $g = g_2' \circ g_1$ such that
$(g_1, a) \models S_1$
$(g_2, a) \models S_2$



Rules

$(c, a) \models p$, if c is an application of p at a



Spatial - Temporal Logic

spatial structure of computations

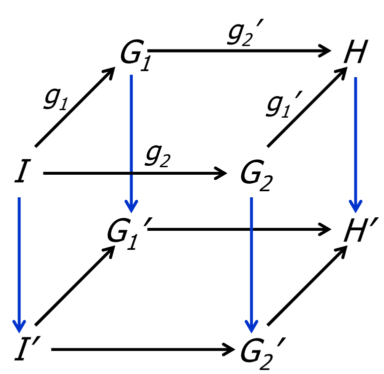
- exists x . F - hiding
- F + G - gluing

propositional logic

- F implies G - implication
- ...

Spatial Structure of Computations

(g, a) |= $F_1 + F_2$, if $c = c_2' \circ c_1$ such that
$(c_1 , a) \models F_1$
$(c_2 , a) \models F_2$



pushout in the horizontal category of computations

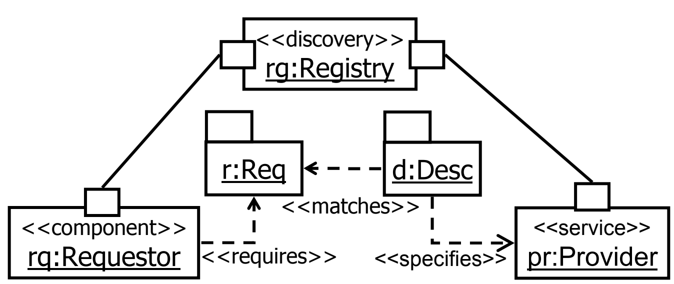
Outline

- ✓ an example
- ✓ an algebraic model of concurrent derivations
- ✓ an arrow logic with a spatial twist
- ✖ some open questions

Want to say things like

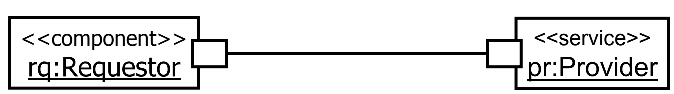
- ✖ In a graph containing P, publish, find, and bind are applicable.
- ✖ P implies pre(publish ; find; bind)

P:



- ✖ In the resulting graph there is a connection between rq and pr.
- ✖ post(publish ; find; bind) implies Q

Q:



Some Open Questions

Is this logic

- at the right level of abstraction ?
  - are two derivations indistinguishable iff they are shift-equivalent
- supporting reasoning ?
  - rules like

$$\frac{c \models p + q}{c \models p \,;\, q} \qquad \frac{c \models p + q}{c \models q \,;\, p}$$

    follow directly from the axioms of the model
  - completeness should follow from the free construction

What about 'graphs as arrows' view?