Nested Graph Conditions as String Diagrams

Filippo Bonchi[1], Andrea Corradini[1], and Arend Rensink[2]

[1] University of Pisa, Italy, filippo.bonchi@unipi.it
[2] University of Pisa, Italy, andrea.corradini@unipi.it
[3] University of Twente, Netherlands, arend.rensink@utwente.nl

1 Definitions

- An *interface* I is a discrete graph.
- An *open graph* is an arrow $g: I \to G$ where I is an interface; we say that g has *interface* $I_g = I$ and *pattern* $P_g = G$. Open graphs are used for different purposes: within a condition to associate (both upper and lower) interfaces with the patterns, and also as assignments — that is, the objects on which satisfaction is defined.
- An *interface morphism* f from I to J is a graph morphism $k : I \to J$.
- An *open graph morphism* $a: g \to h$ a pair of graph morphisms $a^I : I_g \to I_h, a^P : P_g \to P_h$ such that $a^I; h = g; a^P$.

A *condition tree* is a tuple $(V, E, \{d_v\}_{v \in V}, \{u_w\}_{(v,w) \in C})$ such that

- V is a set of vertices;
- $C \subseteq V \times V$ is a set of edges representing the parent-child relation, such that one $v \in P$ (the *root*) has no outgoing edge, and all other elements of V have exactly one outgoing edge (to their *parent*);
- for all $v \in V$, d_v is an open graph. We denote $I_v = I_{d_v}$ and $P_v = P_{d_v}$.
- for all $c = (v, w) \in C$, $u_w: I_w \to P_v$ is an open graph.

The root of a tree $\mathcal{T}$ is denoted $\top_{\mathcal{T}}$; its *root interface* is $I_{\mathcal{T}} = I_{\top_{\mathcal{T}}}$. For an arbitrary vertex $v \in V_{\mathcal{T}}$, $\hat{v}$ will denote the subtree rooted at v.

Satisfaction. Let $\mathcal{T}$ be a condition tree. Satisfaction is defined for nodes $v \in V_{\mathcal{T}}$ and assignments $g: I^v \to G$, as follows. g *satisfies* v, denoted $g \models v$, if there is a graph morphism $h: P^v \to g$ such that *(i)* $g = d_v; h$, and *(ii)* $u_w; a_P \not\models w$ for all children w of v. h is then called a *witness* of $g \models v$. g satisfies $\mathcal{T}$, denoted $g \models \mathcal{T}$, if $g \models \top_{\mathcal{T}}$.

Morphisms. Given two condition trees $\mathcal{T}$,$\mathcal{U}$ with $I_{\mathcal{T}} = I_{\mathcal{U}}$, a morphism $\mathcal{M}$ from $\mathcal{T}$ to $\mathcal{U}$ is a set of triples (v, a, w) where $(v, w) \in (V_{\mathcal{T}} \times V_{\mathcal{U}}) \cup (V_{\mathcal{U}} \times V_{\mathcal{T}})$ and $a: d_v \to d_w$ is an open graph morphism, such that $\top_{\mathcal{M}} = (\top_{\mathcal{T}}, (id, r), \top_{\mathcal{U}}) \in \mathcal{M}$ for some r.

Given a condition tree $\mathcal{T}$, the identity morphism $\mathcal{I}_{\mathcal{T}}$ is defined as

$$\mathcal{I}_{\mathcal{T}} = \{(v, id_{d_v}, v) \mid v \in V_{\mathcal{T}}\} \ .$$

Given two condition tree moprhisms $\mathcal{M}: \mathcal{T} \to \mathcal{U}, \mathcal{N}: \mathcal{U} \to \mathcal{V}$, their composition is defined as

$$\mathcal{M}; \mathcal{N} = \{(v, a; b, x) \mid \exists w.\ (v, a, w) \in \mathcal{M}, (w, b, x) \in \mathcal{N}\} \ .$$

Proposition 1. *Condition trees and their morphisms form a category.*

Source saturation. Given a morphism $\mathcal{M}$, an element $(v, a, w) \in \mathcal{M}$ is called *source-saturated* if for every child x of v, one of the following holds:

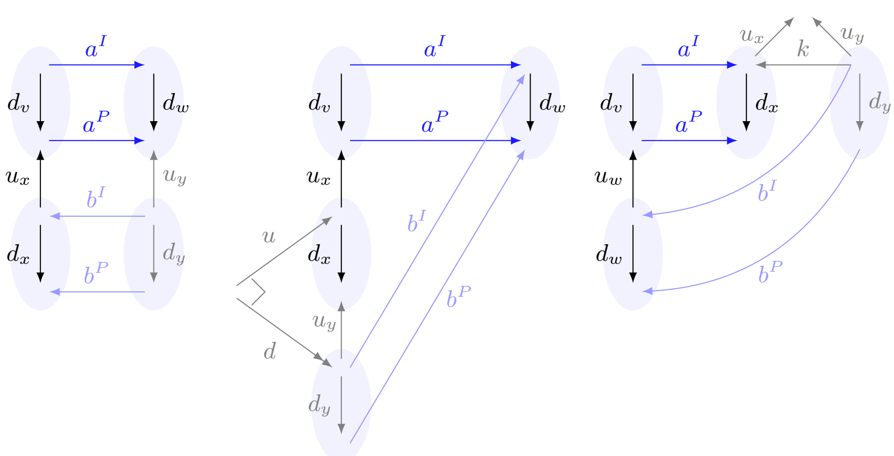
(a) There is a child y of w and a source-saturated $(y, b, x) \in \mathcal{M}$ such that $b^I; u_x; a^P = u_y$.
(b) There is a child y of x and a source-saturated $(y, b, w) \in \mathcal{M}$ such that, constructing the pullback (u, d) of (d_x, u_y), d is epi and $u; u_x; a^P = d; d_y; b^P$.
(c) There is a sibling y of w, a source-saturated $(y, b, x) \in \mathcal{M}$ and an interface morphism $k: I_y \to I_w$ such that $u_y = k; u_w$ and $b^I; u_x; a^P = k; d_w$.

AR: alternatively, if all elements of $\mathcal{M}$ are source-saturated

We call $\mathcal{M}$ source-saturated if $\top_{\mathcal{M}}$ is source-saturated.

Vacuously, the condition of source saturation is fulfilled if v does not have children, hence in that case all $(v, x, a) \in \mathcal{M}$ are source-saturated; this will form the base case of our induction proofs.

The three cases of source saturation are visualised in fig. 1.



(a) Standard (b) Child rule (c) Sibling rule

Fig. 1: The three options for source-saturation

Lemma 1 (source saturation reflects satisfaction). *Let $\mathcal{M}: \mathcal{T} \to \mathcal{U}$ be a condition tree morphism. For all source-saturated $(v, a, w) \in \mathcal{M}$, if $g \models w$ with witness h, then $a^I; g \models v$ with witness $a^P; h$. It follows that, if $\mathcal{M}$ is source-saturated, $g \models \mathcal{U}$ implies $g \models \mathcal{T}$.*

Target saturation. Given a morphism $\mathcal{M}$, an element $(v, a, w) \in \mathcal{M}$ is called *target-saturated* if one of the following holds:

(a) There is a child x of w with $d_x = u_x = d_v$, such that for all children y of x there is a target-saturated (y, c, w)

Lemma 2 (target saturation preserves satisfaction). *Let $\mathcal{M}\colon \mathcal{T} \to \mathcal{U}$ be a condition tree morphism. For all target-saturated $(v, a, w) \in \mathcal{M}$, if $a ; g \models v$ with witness h, then $g \models w$ with witness h' such that $h = a^P ; h'$. It follows that, if $\mathcal{M}$ is target-saturated, $g \models \mathcal{T}$ implies $g \models \mathcal{U}$.*