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 type* is a number $n \in \mathbb{N}$. For any $n \in \mathbb{N}$ we use I_n as a canonical discrete graph with n nodes.
- An *interface* I is a discrete graph; we say that I has type $\tau I = |V_I|$. The only canonical interface of type n is I_n.
- An *open graph* is an arrow $g\colon I \to G$ where I is a discrete graph; we say that g has *type* $\tau g = \tau I$, *interface* I_g and *core* $G_g = G$. g is called *canonical* if I_g is canonical.
- We will only work with canonical interfaces and open graphs.
- An *interface morphism* f from I to J is a graph morphism $f : I \to J$.
- An open graph g *shifted back* over an interface morphism $f : J \to I_g$ is the open graph $f;g$ with type τJ.
- An *open graph morphism* from g to h with $\tau g = \tau h$ is a graph morphism $a : G_g \to G_h$ such that $h = g;a$.
- An open graph g *shifted forward* over a graph morphsm $a\colon G_g \to H$ is the open graph $g;a$ with type $\tau\mathcal{G}$ and core H.

AC: what about defining concretely $I_n = \{1, 2, \ldots, n\}$? And handling it (ambiguously) as a set?

AC: I don't like *core*. Body?

Is this acceptable?

AC: function?

A *condition tree* is a tuple $(V, E, P, \{g_v\}_{v\in V}, \{u_e\}_{e\in E})$ such that

- V is a set of nodes;
- $E \subseteq V \times V$ is a set of edges, such that one $v \in P$ (the *root*) has no incoming edge and all other elements of V have exactly one incoming edge (from its *parent*);
- for all $v \in V$, g_v is an open graph;
- for all $e = (v, w) \in E$, u_e is an open graph (thought of as an upward-pointing arrow, hence u) such that $G_{u_e} = G_{g_v}$ and $\tau u_e = \tau g_w$. In words, d_e connects the interface of g_w to the core of g_v.

AC: P is also two lines above, but not defined. Here should it be V?

What is d_e? I think that it is lighter to define an edge u_e as a graph morphism $u_e : I_{g_w} \to G_{g_v}$

For convenience we also denote $I_v = I_{g_v}$ and $G_v = G_{g_v}$. The root of a tree $\mathcal{T}$ is denoted $\top_{\mathcal{T}}$.

Satisfaction is defined for open graphs, on a node-by-node basis. (Actually, here the node is regarded as the root of its subtree.) $g \models v$ if there is an open graph morphism $a : g_v \to g$ such that for all edges $e = (v, w) \in E$, $u_e; a \not\models w$.

Given two condition trees $\mathcal{T}, \mathcal{U}$, a *condition tree morphism* is a pair $(R, \{a_p\}_{p\in R})$ such that (assuming $V_{\mathcal{T}}$ and $V_{\mathcal{U}}$ to be disjoint, and using $E = E_{\mathcal{T}} \cup E_{\mathcal{U}}$):

AC: too complex to spell out...

- $R \subseteq V_{\mathcal{T}} \times V_{\mathcal{U}} \cup V_{\mathcal{U}} \times V_{\mathcal{T}}$ is a relation between $\mathcal{T}$-nodes and $\mathcal{U}$-nodes and vice versa, such that $(\top_{\mathcal{T}}, \top_{\mathcal{U}}) \in R$.
- For all $p = (v, v') \in R$, $a_p : G_v \to G_{v'}$ is a graph morphism.
- For all $(v, v') \in R$ and all $(v, w) \in E$, one of the following holds:
  - There is a $(v', w') \in E$ such that *(i)* $(w', w) \in R$, and *(ii)* there is a $k : I_{w'} \to I_w$ such that $u_{(v',w')} = k; u_{(v,w)}; a_{(v,v')}$ and $k; g_w = g_{w'}; a_{(w',w)}$.
  - there is a $(w, x) \in E$ such that *(i)* $(x, v') \in R$, and *(ii)* $u_{v,w} = id_{G_v}$, $u_{w,x} = id_{G_w}$ and g_w is epi, with $a_{(v,v')} = g_w; a_{(x,v')}$.