William Bombardelli da Silva

August 11, 2018

Abstract

0.1 Theoretical Review

In this section, we introduce the theoretical concepts used along this thesis. The definitions below are taken from the works of ...We first go on to define graphs and graph grammars and then, building upon it, we construct the so-called triple graph grammars.

0.1.1 Graph Grammars

We start presenting our notation for graphs and grammars, accompanied by examples, then we introduce the dynamic aspects of the graph grammar formalism that is, how graph grammars are to be interpreted.

Definition. A directed labeled graph G over the set of symbols Σ, $G = (V, E, \phi)$ consists of a finite set of vertices V, a set of labeled directed edges $E \subseteq V \times \Sigma \times V$ and a total vertex labeling function $\phi : V \to \Sigma$. Directed labeled graphs are often referred to simply as graphs. For a fixed graph G we refer to its components as V_G, E_G and ϕ_G. Moreover, we define the special empty graph as $\varepsilon := (\emptyset, \emptyset, \emptyset)$ and we denote the set of all graphs over Σ by $\mathcal{G}_\Sigma$.

If $\phi_G(v) = a$ we say v is labeled by a. Two vertices v and w are neighbors (also adjacent) iff there is one or more edges between them, that is, $(v, \_, w) \in E_G \vee (w, \_, v) \in E_G$. Two graphs G and H are disjoint iff $V_G \cap V_H = \emptyset$.

Definition. A morphism of graphs G and H is a total mapping $m : V_G \to V_H$.

Definition. An isomorphism of directed labeled graphs G and H is a bijective mapping $m : V_G \to V_H$ that maintains the connections between vertices and their labels, that is, $(v, l, w) \in E_G$ if, and only if, $(m(v), l, m(w)) \in E_H$ and if $m(v) = w$ then $\phi_G(v) = \phi_H(w)$. In this case, G and H are said to be isomorphic, we write $G \cong H$, and we denote the equivalence class of all graphs isomorphic to G by $[G]$. Notice that, contrary to isomorphisms, morphism do not require bijectivity nor label or edge-preserving properties.

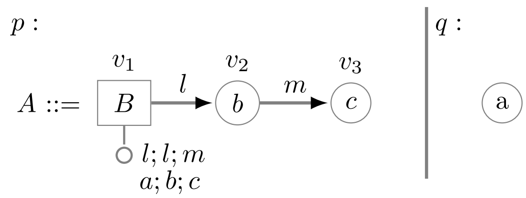
Definition. A Γ-boundary graph G is such that vertices labeled with any symbol from Γ are not neighbors. That is, the graph G is Γ-boundary iff, $\nexists (v, \_, w) \in E_G.\ \phi_G(v) \in \Gamma \wedge \phi_G(w) \in \Gamma$.

Definition. An graph grammar with neighborhood-controlled embedding (NCE graph grammar) $GG = (\Sigma, \Delta \subseteq \Sigma, S \in \Sigma, P)$ consists of a finite set of symbols Σ that is the alphabet, a subset of the alphabet $\Delta \subseteq \Sigma$ that holds the terminal symbols (we define the complementary set of non-terminal symbols as $\Gamma := \Sigma \setminus \Delta$), a special symbol of the alphabet $S \in \Sigma$ that is the start symbol, and a finite set of production rules P of the form $(A \to R, \omega)$ where $A \in \Gamma$ is the so-called left-hand side, $R \in \mathcal{G}_\Sigma$ is the right-hand side and $\omega : V_R \nrightarrow 2^{\Sigma \times \Sigma}$ is the partial embedding function from the R's vertices to pairs of edge and label symbols. NCE graph grammars are often referred to as graph grammars or simply as grammars.

Vertices from the right-hand sides of rules labeled by non-terminal (terminal) symbols are said to be non-terminal (terminal) vertices.

Definition. A boundary graph grammar with neighborhood-controlled embedding (BNCE graph grammar) GG is such that non-terminal vertices of the right-hand sides of rules are not neighbors. That is, the graph grammar GG is boundary iff all its rules' right-hand sides are Γ-boundary graphs.

In the following, we present our concrete syntax inspired by the well-known backus-naur form to denote BNCE graph grammar rules. Let $GG = (\{A, a, b, c\}, \{a, b, c\}, A, \{p, q\})$ be a graph grammar with production rules $p = (A \to G, \omega)$ and $q = (A \to H, \zeta)$ where $G = (\{v_1, v_2, v_3\}, \{(v_1, l, v_2), (v_2, m, v_3)\}, \{v_1 \mapsto B, v_2 \mapsto b, v_3 \mapsto c\})$, $\omega = \{v_1 \mapsto \{(l, a), (l, b), (m, c)\}\}$, and $H = (\{u_1\}, \emptyset, \{u_1 \mapsto a\})$ and $\zeta = \emptyset$, we denote p and q together as



Notice that, we use squares for non-terminal vertices, circles for terminal vertices, position the respective label inside the shape and the (possibly omitted) identifier over it. Over each edge is positioned its respective label. To depict the embedding function, we place below the respective vertex a small circle labeled with the image pairs of the embedding function for this node aligned vertically and separated by semi-colons.

With these syntactic notions of the formalism presented, we introduce below its semantics by means of the concepts of derivation step, derivation and language.

Definition. Let $GG = (\Sigma, \Delta, S, P)$ be a graph grammar and G and H be two graphs over Σ disjoint from any right-hand side from P, G concretely derives in one step into H with rule r and vertex v, we write $G \overset{r,v}{\Rrightarrow}_{GG} H$ and call it a concrete derivation step, if, and only if, the following holds:

$$\begin{aligned}
r &= (A \to R, \omega) \in P \text{ and } A = \phi_G(v) \text{ and} \\
V_H &= (V_G \setminus \{v\}) \cup V_R \text{ and} \\
E_H &= (E_G \setminus \{(w, l, t) \in E_G \mid v = w \vee v = t\}) \\
&\quad \cup E_R \\
&\quad \cup \{(w, l, t) \mid (w, l, v) \in E_G \wedge (l, \phi_G(w)) \in \omega(t)\} \\
&\quad \cup \{(t, l, w) \mid (v, l, w) \in E_G \wedge (l, \phi_G(w)) \in \omega(t)\} \text{ and} \\
\phi_H &= (\phi_G \setminus \{(v, x) \mid x \in V_G\}) \cup \phi_R
\end{aligned}$$

Notice that, without loss of generalization, we set $\omega(t) = \emptyset$ for all vertices t without an image defined in ω.

If G concretely derives in one step into any graph H' isomorphic to H, we say it derives in one step into H' and write $G \overset{r,v}{\Rightarrow}_{GG} H'$.

When GG, r or v are clear in the context or irrelevant we might omit them and simply write $G \Rrightarrow H$ or $G \Rightarrow H$. Moreover, we denote the reflexive transitive closure of $\Rightarrow$ by $\Rightarrow^*$ and, for $G \Rightarrow^* H'$, we say G derives in one or more steps into H', or simply G derives into H'.

Definition. A derivation D in GG is a sequence of derivation steps and is written as

$$D = (G_0 \overset{r_0,v_0}{\Rightarrow} G_1 \overset{r_1,v_1}{\Rightarrow} G_2 \overset{r_2,v_2}{\Rightarrow} \dots \overset{r_{n-1},v_{n-1}}{\Rightarrow} G_n)$$

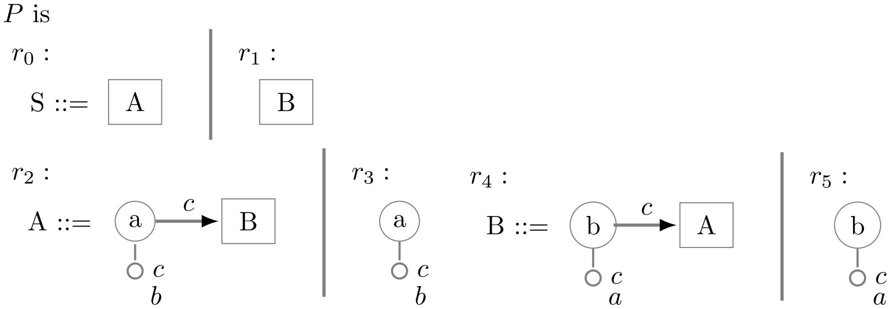
Definition. The language $L(GG)$ generated by the grammar GG is the set of all graphs containing only terminal vertices derived from the initial graph $Z = (\{v_s\}, \emptyset, \{v_s \mapsto S\})$, that is

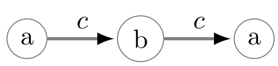
$$L(GG) = \{H \mid \phi_H(V_H) \subseteq \Delta \wedge Z \Rightarrow^* H\}$$

Notice that for every graph $G \in L(GG)$, there is at least one finite derivation $(Z \overset{r_0,v_0}{\Rightarrow} \dots \overset{r_{n-1},v_{n-1}}{\Rightarrow} G)$, but it is not guaranteed that this derivation be unique. In the case that there are more than one derivation for a G, we say that the grammar GG is ambiguous.

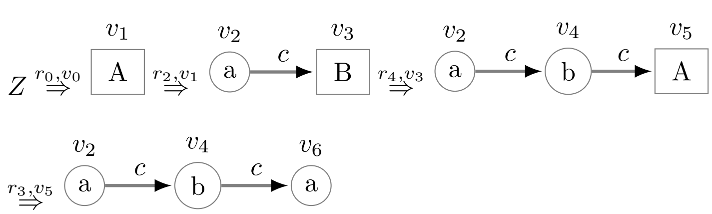
Below we give one example of a grammar whose language consists of all chains of one or more vertices with interleaved vertices labeled with a and b.

Example. Chains of a's and b's. $GG = (\{S, A, B, a, b, c\}, \{a, b, c\}, S, P)$, where P is



The graph $G =$ 

belongs to $L(GG)$ because it contains only terminal vertices and Z derives into it using the following derivation:



0.1.2 Triple Graph Grammars

Building upon the concepts of graphs and graph grammars, we present, in the following, our understanding over triple graphs and triple graph grammars (TGGs), supported by the TGG specification from ().

Definition. A directed labeled triple graph $TG = G_s \overset{m_s}{\leftarrow} G_c \overset{m_t}{\rightarrow} G_t$ over Σ consists of three disjoint directed labeled graphs over Σ (see 0.1.1), respectively, the source graph G_s, the correspondence graph G_c and the target graph G_t, together with two injective morphisms (see 0.1.1) $m_s : V_{G_c} \rightarrow V_{G_s}$ and $m_t : V_{G_c} \rightarrow G_{G_t}$. Directed labeled triple graphs are often referred to simply as triple graphs and we might omit the morphisms' names in the notation. Moreover, we denote the set of all triple graphs over Σ as $\mathcal{TG}_\Sigma$. We might refer to all vertices of TG by $V_{TG} := V_s \cup V_c \cup V_t$, all edges by $E_{TG} := E_s \cup E_c \cup E_t$ and the complete labeling function by $\phi_{TG} := \phi_{G_s} \cup \phi_{G_c} \cup \phi_{G_t}$.

Definition. A Γ-boundary triple graph $TG = G_s \leftarrow G_c \rightarrow G_t$ is such that G_s, G_c and G_t are Γ-boundary graphs.

Below we start introducing the standard definition of TGG of the current research's literature. As the reader should notice, this definition of TGG does not fit our needs optimally, because it defines a context-sensitive-like graph grammar whilst we wish a context-free-like graph grammar to use together with the NCE graph grammar formalism. Hence, after presenting the conventional TGG definition, we refine it to create a NCE TGG, that fits our context best.

Definition. A triple graph grammar $TGG = (\Sigma, \Delta \subseteq \Sigma, S \in \Sigma, P)$ consists of, analogously to graph grammars (see 0.1.1), an alphabet Σ, a set of terminal symbols Δ (also define $\Gamma := \Sigma \setminus \Delta$), a start symbol S and a set of production rules P of the form $L \rightarrow R$ with $L = L_s \leftarrow L_c \rightarrow L_t$ and $R = R_s \leftarrow R_c \rightarrow R_t$ and $L \subseteq R$.

Definition. A triple graph grammar with neighborhood-controlled embedding (NCE TGG) $TGG = (\Sigma, \Delta \subseteq \Sigma, S \in \Sigma, P)$ consists of, an alphabet Σ, a set of terminal symbols Δ (also define $\Gamma := \Sigma \setminus \Delta$), a start symbol S and a set of production rules P of the form $((A, B, C) \rightarrow R_s \leftarrow R_c \rightarrow R_t, \omega_s, \omega_t)$ with $A, B, C \in \Gamma$ being the left-hand side, $(R_s \leftarrow R_c \rightarrow R_t) \in \mathcal{TG}_\Sigma$ the right-hand side and $\omega_s : V_{R_s} \nrightarrow 2^{\Sigma \times \Sigma}$ and $\omega_t : V_{R_t} \nrightarrow 2^{\Sigma \times \Sigma}$ the partial embedding functions from the right-hand side's vertices to pairs of edge and label symbols. We might refer to the complete embedding function by $\omega := \omega_s \cup \omega_t$.

Definition. A boundary triple graph grammar with neighborhood-controlled embedding (BNCE TGG) TGG is such that non-terminal vertices of the right-hand sides of rules are not neighbors. That is, the triple graph grammar TGG is boundary iff all its rules' right-hand sides are Γ-boundary triple graphs.

In the following, the semantics for NCE TGG is presented analogously to the semantics for NCE graph grammars.

Definition. Let $TGG = (\Sigma, \Delta, S, P)$ be a NCE TGG and G and H be two triple graphs over Σ disjoint from any right-hand side from P, G concretely derives in one step into H with rule r and distinct vertices v_s, v_c, v_t, we write $G \overset{r,v_s,v_c,v_t}{\Rrightarrow}_{TGG} H$ if, and only if, the following holds:

$$
\begin{aligned}
r &= ((A,B,C) \to R, \omega_s, \omega_t) \in P \text{ and} \\
A &= \phi_{G_s}(v_s) \text{ and } B = \phi_{G_c}(v_c) \text{ and } C = \phi_{G_t}(v_t) \\
V_H &= (V_G \setminus \{v_s, v_c, v_t\}) \cup V_R \text{ and} \\
E_H &= (E_G \setminus \{(w,l,t) \in E_G \mid w \in \{v_s,v_c,v_t\} \vee t \in \{v_s,v_c,v_t\}\}) \\
&\quad \cup E_R \\
&\quad \cup \{(w,l,t) \mid (w,l,v) \in E_G \wedge (l, \phi_G(w)) \in \omega(t)\} \\
&\quad \cup \{(t,l,w) \mid (v,l,w) \in E_G \wedge (l, \phi_G(w)) \in \omega(t)\} \text{ and} \\
\phi_H &= (\phi_G \setminus \{(v_s,x),(v_c,x),(v_t,x) \mid x \in V_G\}) \cup \phi_R
\end{aligned}
$$

Notice that, without loss of generalization, we set $\omega(t) = \emptyset$ for all vertices t without an image defined in ω.

Analogously to graph grammars, if $G \overset{r,v_s,v_c,v_t}{\Rrightarrow}_{TGG} H$ and $H' \in [H]$, then $G \overset{r,v_s,v_c,v_t}{\Rightarrow}_{TGG} H'$, moreover the reflexive transitive closure of $\Rightarrow$ is denoted by $\Rightarrow^*$ and we call these relations by the same names as before, namely, derivation in one step and derivation. We might also omit identifiers.

Definition. A derivation D in TGG is a sequence of derivation steps

$$D = (G_0 \overset{r_0,s_0,c_0,t_0}{\Rightarrow} G_1 \overset{r_1,s_1,c_1,t_1}{\Rightarrow} G_2 \overset{r_2,s_2,c_2,t_2}{\Rightarrow} \dots \overset{r_{n-1},s_{n-1},c_{n-1},t_{n-1}}{\Rightarrow} G_n)$$

Definition. The language $L(TGG)$ generated by the triple grammar TGG is the set of all triple graphs containing only terminal vertices derived from the initial triple graph $Z = Z_s \leftarrow Z_c \rightarrow Z_t$ where $Z_s = (\{s_0\}, \emptyset, \{s_0 \mapsto S\})$, $Z_c = (\{c_0\}, \emptyset, \{c_0 \mapsto S\})$ and $Z_t = (\{t_0\}, \emptyset, \{t_0 \mapsto S\})$, that is

$$L(TGG) = \{H \mid \phi_H(V_H) \subseteq \Delta \wedge Z \Rightarrow^* H\}$$

Our concrete syntax for NCE TGG is similar to the one for NCE graph grammars and is presented below by means of an example. The only difference is at the left-hand sides, which are now depicted by three symbols and the dashed lines at the right-hand sides that depict the morphisms between the correspondence graph and the source and target graphs.

0.2 Parsing of Graphs with BNCE Graph Grammars

0.3 Transformation of Graphs with BNCE Triple Graph Grammars