Finding Double Euler Trails of Planar Graphs in Linear Time*

Zhi-Zhong Chen† Xin He‡ Chun-Hsi Huang§

Abstract

This paper answers an open question in the design of complimentary metal-oxide semiconductor (CMOS) VLSI circuits. The question asks whether a *polynomial-time* algorithm can decide if a given planar graph has a plane embedding $\mathcal{E}$ such that $\mathcal{E}$ has an Euler trail $P = e_1 e_2 \ldots e_m$ and its dual graph has an Euler trail $P^* = e_1^* e_2^* \ldots e_m^*$ where e_i^* is the dual edge of e_i for $i = 1, 2, \ldots, m$. This paper answers this question in the affirmative, by presenting a *linear-time* algorithm.

Key Words: planar graph, dual graph, Euler trail
AMSMOS subject classification: 05C45, 05C85, 05C90, 68Q35, 68R10

1 Introduction

Throughout this paper, a graph may have multiple edges but never has loops. A graph G is *planar* if it can be drawn on the plane so that the edges only intersect at their end vertices. Such a drawing is called an *embedding* of G. A *plane graph* is a planar graph with a fixed embedding. A *two-terminal graph* [1] (TTG) $G = (V, E)$ is a plane graph with a pair (s, t) of specified vertices on the outer face such that adding the edge $\{s, t\}$ to G yields a biconnected plane graph [4, 6]. We call s the *source* of G and t the *sink* of G. We also call s and t the *poles* of G and the other vertices the *nonpoles* of G. As in previous studies, we visualize G in such a way that s is at the South Pole while t is at the North Pole.

The poles of G divide the boundary of the outer face into two paths, which are called the *west side* and the *east side* of G, respectively. In the dual graph $G^* = (V^*, E^*)$ of G, there are two dual vertices s^* and t^* corresponding to the outer face of G. The dual edges incident to s^* (respectively, t^*) in G^* correspond to the edges on the west (respectively, east) side of G. Figure 1(2) shows a TTG G and its dual G^*. G^* is a TTG with source s^* and sink t^* [12].

In CMOS technology, the basic layout of a circuit C of transistors on a VLSI chip uses two rows of transistors: A row of pMOS transistors is laid out next to a row of nMOS transistors [13]. The circuit C can be represented by a pair of TTGs: The p-transistors are represented by

*A preliminary version of this work was presented at the 40th Annual Symposium on Foundations of Computer Science, Oct. 1999, New York City, New York.

†Department of Mathematical Sciences, Tokyo Denki University, Hatoyama, Saitama 350-0394, Japan. Email: chen@r.dendai.ac.jp.

‡Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA. Email: xinhe@cse.buffalo.edu. Research supported in part by NSF Grant CCR-9912418.

§Department of Computer Science and Engineering, State University of New York at Buffalo, Buffalo, NY 14260, USA. Email: ch25@cse.buffalo.edu.

[1] Previously called *planar st-graphs*, and treated as digraphs just for convenience.

G, where each edge of G represents a p-transistor. The n-transistors are represented by the dual graph G^* of G, where each edge of G^* represents an n-transistor. In Figure 1, (1) shows a circuit implementing the Boolean function $z = \bar{e} \wedge (\bar{a} \vee \bar{b}) \wedge (\bar{c} \vee \bar{d})$, and (2) shows the corresponding TTGs G and G^*.

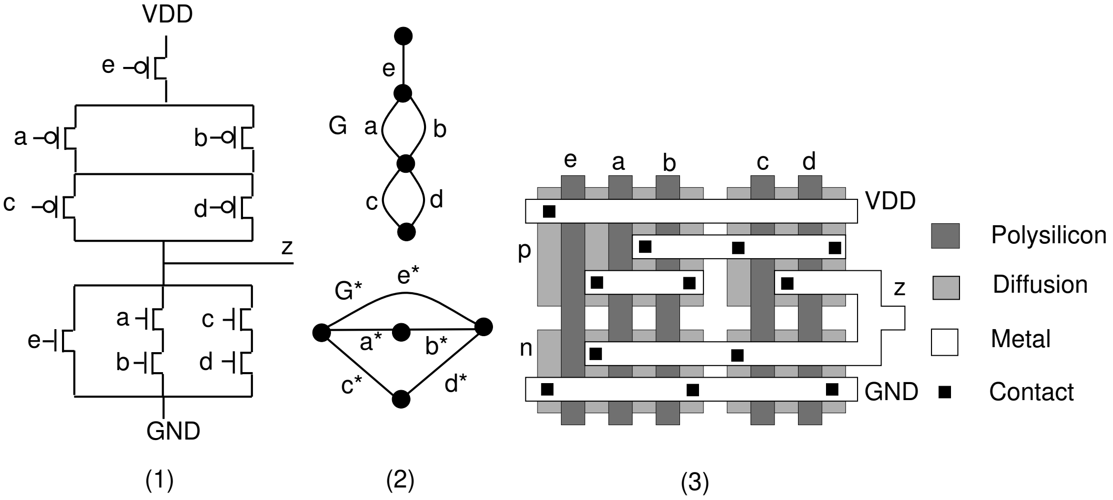

Figure 1: (1) A circuit C implementing $z = \bar{e} \wedge (\bar{a} \vee \bar{b}) \wedge (\bar{c} \vee \bar{d})$. (2) The corresponding TTGs G and G^*. (3) A layout of C.

1.1 Double Euler trails and optimal layouts

A *trail* P in G is a sequence $e_1 e_2 \dots e_k$ of distinct and oriented edges of G such that for each $i \in \{1, \dots, k-1\}$, the ending vertex of e_i and the starting vertex of e_{i+1} are the same. Possibly, P has repeated vertices. An *Euler trail* is a trail that visits each edge of G exactly once. For each edge e of G, let e^* be the dual edge of e in G^*. If the sequence $P^* = e_1^* e_2^* \dots e_k^*$ is also a trail in G^*, we call P a *double trail* [2] in G and call P^* the *dual trail* of P. If P is a double trail and traverses every edge of G exactly once, we call P a *double Euler trail* (DET) in G.

For each edge e in G, the p-transistor represented by e has the same input as the n-transistor represented by e^*; hence, in the layout, the two transistors are required to be laid out vertically aligned (to facilitate input line connection) [13]. Transistors which are physically adjacent and have common diffusion regions may be collapsed. Otherwise, a vertical gap must be inserted between the diffusion region of the two transistors. Two transistors corresponding to two consecutive edges on a double trail can be laid out physically adjacent without vertical gap [13]. Figure 1(3) shows a layout of C using this approach. Note that the sequence eab is a double trail of G. So, in the layout, no vertical gaps are needed between the transistors e, a, b. This is also true for transistors c, d because of the double trail cd. However, since the edges b and c are incident in G but not in G^*, there is a vertical gap between the transistors b and c in the layout. In Figure 2, (1) shows a logically equivalent circuit, and (2) shows the corresponding TTGs G and G^*. Since $abecd$ is a DET of G, the transistors can be laid out in this order with no vertical gap, as seen in Figure 2(3).

[2] Previously called "dual trail". We reserve the term "dual trail" exclusively to refer to a trail in G^*.

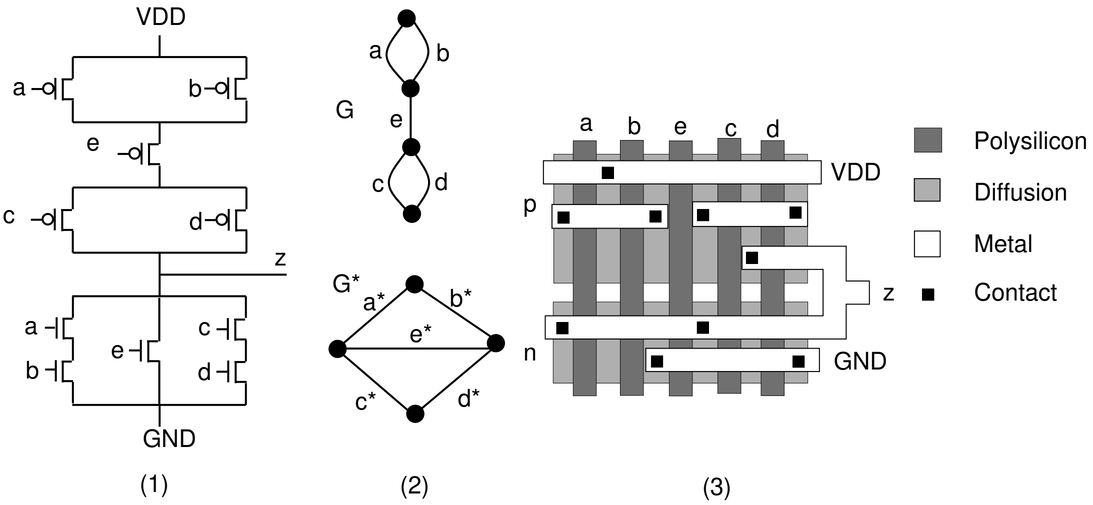

Figure 2: (1) A circuit C logically equivalent to the circuit in Fig 1 (1). (2) The corresponding TTGs G and G^*. (3) A layout of C using the DET *abecd*.

1.2 Problem definitions

The circuits in Figures 1(1) and 2(1) are logically equivalent. Thus, the TTGs G in Figures 1(2) and 2(2) should be viewed as different embeddings of each other. That is, to obtain a different embedding of a TTG, we may not only horizontally flip blocks of the TTG as one does traditionally, but also vertically flip blocks of the TTG. We postpone the precise definition of "different embeddings" to the end of Section 2. For convenience, we regard G as a different embedding of itself.

As seen from Figures 1 and 2, even if G has no DET, a different embedding of G may have one. A *valid embedding* of G is a different embedding of G that has a DET. A *double-trail cover* (DTC) $\mathcal{Q}$ of G is a set of edge-disjoint double trails in G such that each edge of G appears in a trail in $\mathcal{Q}$. The *size* of $\mathcal{Q}$ is the number of double trails in it. $\mathcal{Q}$ is a *minimum* DTC of G if the size of $\mathcal{Q}$ is minimized over all DTCs of G. A different embedding of G with a minimum DTC leads to a layout with the minimum number of vertical gaps (hence the minimum chip area). We are interested in the following problems:

DET-F Problem: Given a TTG G, decide whether G has a DET.

DET-V Problem: Given a TTG G, decide whether G has a valid embedding.

DTC-F Problem: Given a TTG G, find a minimum DTC of G.

DTC-V Problem: Given a TTG G, find a different embedding $\mathcal{G}$ of G and a minimum DTC $\mathcal{Q}$ of $\mathcal{G}$ such that the size of $\mathcal{Q}$ is minimized over all different embeddings of G.

1.3 Previous work and our result

The layout method described above was first introduced by Uehara and VanCleemout [13]. Most previous studies in this area are concerned with *series-parallel* (SP) graphs, a much restricted subclass of TTGs. This is partially because the problems above for general TTGs are more difficult to solve. A heuristic algorithm for solving the DTC-V problem for SP graphs was given in [13]. Lengauer and Müller [7] developed linear-time algorithms for finding a minimum trail cover (not DTC) for SP graphs. Nair, Buss and Reif [10] presented polynomial-time algorithms for solving the DTC-F problem and the DTC-V problem both for a restricted class of embedded SP graphs. Maziasz and Hayes [8, 9] gave a linear-time algorithm for solving the DTC-F problem for SP graphs, and a worst-case exponential-time algorithm for the DTC-V problem for SP graphs. Linear-time algorithms for solving the DET-V problem for SP graphs were given in [5]. These algorithms are based on dynamic programming and are complicated. Whether the DTC-V problem for SP graphs can be solved in polynomial time is still open.

For general TTGs, a linear-time algorithm for solving the DET-F problem was given in [1]. The DTC-F problem is NP-hard [14, 15], which implies that the DTC-V problem for TTGs is also NP-hard. Whether the DET-V problem for general TTGs can be solved in *polynomial time* is posed as an open question in [1].

Here, we answer this question in the affirmative, by presenting a *linear-time* algorithm for the DET-V problem. Our algorithm is based on careful analysis of the structures of DTCs of TTGs. In more details, we first find out the types of those DTCs that are the intersection of a DET of a TTG G and a subgraph of G. We then define two operations for composing such DTCs of edge-disjoint subgraphs into a single such DTC of a larger subgraph. We further prove that the two operations have nice properties based on which a linear-time algorithm can be designed. The main data structure we use is the *SPQR-decomposition tree* of a TTG [3].

1.4 Organization of this paper

Section 2 describes the *SPQR decomposition* of a TTG. Section 3 shows several useful properties of different embeddings of a TTG. Section 4 discusses the special case of the DET-V problem where neither the input TTG nor its dual graph has an odd-degree vertex. Section 5 gives a polynomial-time algorithm for the DET-V problem. Section 6 refines this algorithm to a linear-time algorithm.

2 SPQR decomposition

Throughout this paper, for a TTG G, G^* denotes the dual graph of G, $s(G)$ denotes the source of G, $t(G)$ denotes the sink of G, and $|G|$ denotes the total number of vertices and edges in G. For an unordered list L, we write $L = \langle a_1, \ldots, a_k \rangle$ where a_1 through a_k are the elements of L. For an ordered list L, we write $L = (a_1, \ldots, a_k)$ where $a_1, \ldots, a_k$ are the elements of L and are ordered in this order.

In the rest of this section, fix a TTG G. A *cut vertex* of G is a vertex whose removal disconnects G. G is *biconnected* if it has no cut vertex. A *biconnected component* of G is a maximal subgraph of G with no cut vertex. A *split pair* of G is either a pair of adjacent vertices, or a pair of vertices whose removal disconnects G' where G' is the graph obtained from G by adding the edge $\{s(G), t(G)\}$. A *split component* of a split pair $\langle u, v \rangle$ is either an edge $\{u, v\}$ or a maximal subgraph H of G such that H is a TTG with poles u and v and $\langle u, v \rangle$ is not a split pair of H. A split pair $\langle u, v \rangle$ of G is *maximal* if there is no other split pair $\langle w_1, w_2 \rangle$ in G such that $\langle u, v \rangle$ is contained in a split component of $\langle w_1, w_2 \rangle$.

The *decomposition tree* T of G describes a recursive decomposition of G with respect to its split pairs [2, 3]. T is a rooted ordered tree whose nodes are of four types: S, P, Q, and R. Each node μ of T has an associated TTG, called the *skeleton* of μ and denoted by $\mathrm{Sk}(\mu)$. Also, it is associated with an edge in the skeleton of the parent χ of μ, called the *virtual edge* of μ in $\mathrm{Sk}(\chi)$. T is recursively defined as follows.

Trivial case: G consists of a single edge $\{s(G), t(G)\}$. Then, T consists of a single Q-node whose skeleton is G itself.

Series case: G is not biconnected. Let $c_1, \ldots, c_{k-1}$ $(k \geq 2)$ be the cut vertices of G. Since G is a TTG, each $c_i \in \{c_1, \ldots, c_{k-1}\}$ is contained in exactly two biconnected components G_i and G_{i+1} of G such that $s(G)$ is in G_1 and $t(G)$ is in G_k. The root of T is an S-node μ. $\mathrm{Sk}(\mu)$ is a path $e_1, \ldots, e_k$, where the edge $e_i = \{c_{i-1}, c_i\}$, $c_0 = s(G)$, and $c_k = t(G)$.

Parallel case: $s(G)$ and $t(G)$ constitute a split pair of G with split components $G_1, \ldots, G_k$ $(k \geq 2)$. Then, the root of T is a P-node μ. $\mathrm{Sk}(\mu)$ consists of k parallel edges between $s(G)$ and $t(G)$, denoted by $e_1, \ldots, e_k$.

Rigid case: none of the above cases applies. Let $\langle s_1, t_1 \rangle, \ldots, \langle s_k, t_k \rangle$ $(k \geq 1)$ be the maximal split pairs of G. For $i = 1, \ldots, k$, let G_i be the union of all the split components of $\langle s_i, t_i \rangle$. The root of T is an R-node μ. $\mathrm{Sk}(\mu)$ is obtained from G by replacing each subgraph G_i with an edge $e_i = \{s_i, t_i\}$. Note that adding the edge $\{s(G), t(G)\}$ to $\mathrm{Sk}(\mu)$ yields a triconnected graph. By this, $\mathrm{Sk}(\mu)$ has no different embedding other than itself [11].

In the last three cases, μ has children $\nu_1, \ldots, \nu_k$ (in this order), such that ν_i is the root of the decomposition tree of graph G_i for all $i \in \{1, \ldots, k\}$. The virtual edge of node ν_i is the edge e_i in $\mathrm{Sk}(\mu)$. G_i is called the *pertinent graph* of ν_i and is denoted by $\mathrm{Pt}(\nu_i)$. Note that G is the pertinent graph of the root of T. Figure 3 illustrates the decomposition tree of a TTG G. The skeletons of the nodes μ and ν are also shown.

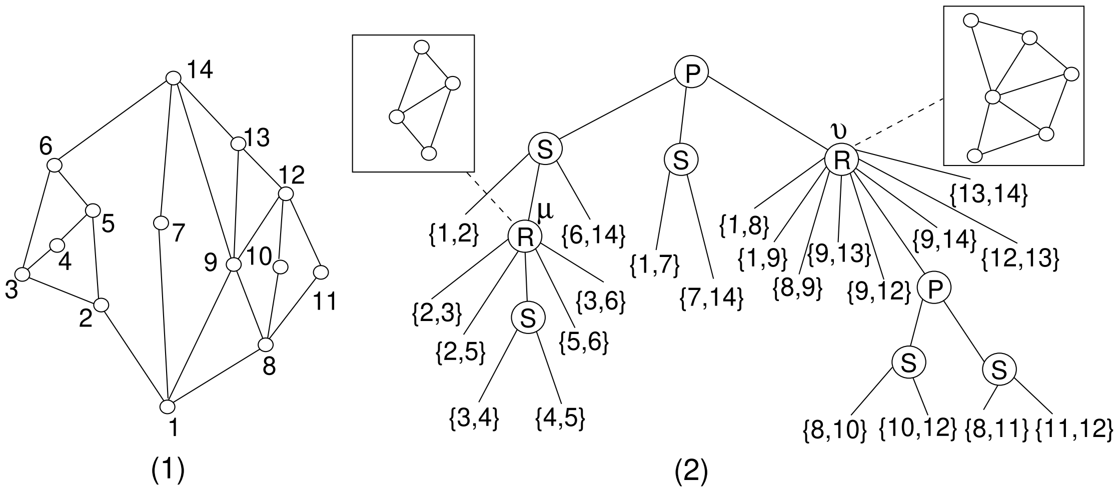

Figure 3: (1) A TTG G. (2) Its decomposition tree T.

By the definition of T, no child of an S-node is an S-node and no child of a P-node is a P-node.

Lemma 1 (see [3]) *T has $O(|G|)$ nodes. Also, the total number of edges of the skeletons stored at the nodes of T is $O(|G|)$.*

The *dual decomposition-tree* of T is the decomposition tree of the dual graph of G.

Lemma 2 *The dual decomposition-tree of T can be obtained from T by exchanging the roles of the S-nodes and the P-nodes and further changing the skeleton of each node to the dual graph of the skeleton.*

Proof: Obvious. □

A *different embedding* of G is obtained from G by a sequence of operations from the following set [7, 8, 9, 10, 13]:

- Horizontally flipping the skeleton of an R-node around its poles.
- Permuting the children of a P-node.
- Vertically flipping the skeleton of an R-node, which is equivalent to horizontally flipping the dual skeleton around its poles.
- Permuting the children of an S-node, which is equivalent to permuting the children of the dual P-node.

Note that if $\mathcal{G}$ is a different embedding of a TTG G, the circuits represented by $\mathcal{G}$ and G are logically equivalent. Also note that the commonly-known definition of "different embeddings" only allows the first two operations [2, 3]. Hence, the meaning of the phrase "different embedding" used here is different from its commonly-known meaning.

3 Properties of embeddings and double trails

Throughout this paper, a list is ordered unless stated explicitly otherwise. An unordered list is actually a multiset. So, for two unordered lists $L_1 = \langle a_1, \ldots, a_k\rangle$ and $L_2 = \langle b_1, \ldots, b_\ell\rangle$, we write $L_1 \cup L_2$ for the unordered list $\langle a_1, \ldots, a_k, b_1, \ldots, b_\ell\rangle$. For an ordered (respectively, unordered) list L and a set S, let $L - S$ denote the ordered (respectively, unordered) list obtained from L by deleting all appearances of each $x \in S$.

The following fact is well known.

Fact 1 *For a connected graph G, the following two statements hold:*

- *G has a closed Euler trail if and only if it has no odd-degree vertex; in this case G is called* Eulerian*.*
- *G has an open Euler trail if and only if it contains exactly two odd-degree vertices, which are the first and the last vertex of the trail.*

In the rest of this section, fix a TTG G. Let T be the decomposition tree of G. A *block* of G is either the pertinent graph of a node in T or the union of the pertinent graphs of some consecutive children of a P- or S-node in T. A *proper block* of G is a block B such that $G \neq B$. Let $B = \mathrm{Pt}(\mu)$ where μ is a node of T. For every child ν of μ in T, we call $\mathrm{Pt}(\nu)$ a *child block* of B.

The following properties of double trails are clear. They play very important roles in our algorithm.

Non-crossing Property: Let Q be a double trail of G. Let e_1 and e_2 be two consecutive edges in Q. Then, e_1 and e_2 appear consecutively around v in G, where v is the ending vertex of e_1 in Q.

Let B be a block of G. Let e_1 (e_2, respectively) be the edge in B incident to $s(B)$ that is on the west (east, respectively) side of B. (If there is only one edge in B incident to $s(B)$ then

$e_1 = e_2$). Let e_3 (e_4, respectively) be the edge in B incident to $t(B)$ that is on the west (east, respectively) side of B. (If there is only one edge in B incident to $t(B)$ then $e_3 = e_4$). We call e_1, e_2, e_3, e_4 the *access edges* of B. Each access edge e_i is incident to a pole of B and its dual edge e_i^* is incident to a pole of B^*; only access edges of B satisfy this property.

Access-edge Property: Let Q be any double trail of G. Let e', e'' be two consecutive edges in Q such that e' is in B and e'' is not in B. Then, e' must be an access edge of B.

Lemma 3 (see [1]) *Let $e = \{u, v\}$ be an edge in G and $e^* = \{F_1, F_2\}$ its dual edge. Let Q be a maximal double trail in G starting with e. If the direction of e in Q and the direction of e^* in the dual trail Q^* are given, then Q is uniquely determined and can be traced out in $O(|Q|)$ time.*

Proof: (Sketch) Suppose that Q traverses e from u to v while Q^* traverses e^* from F_1 to F_2. Then, the successor of e in Q must be the edge incident to v and on the face F_2 of G. Repeating this argument, we can uniquely trace out all edges of Q. □

The *degree parity* of a vertex in G is 0 if its degree is even, and is 1 otherwise.

Fact 2 *If G is Eulerian, then so is every different embedding of G.*

Proof: Let μ be a node of T, and $B = \mathrm{Pt}(\mu)$. Clearly, no vertex gets a different degree after B is horizontally flipped. Similarly, when μ is a P-node, no vertex gets a different degree after the children of μ are permuted.

Suppose that μ is an R-node. After B is vertically flipped, only the poles $s(B)$ and $t(B)$ may get a different degree in the resulting embedding. Let m_B be the number of edges in B. For each vertex v in B, let $d_B(v)$ be the degree of v in B. Thus, $2m_B = d_B(s(B)) + d_B(t(B)) + \sum_{v \neq s(B), t(B)} d_B(v)$. Since $d_B(v)$ is even for all $v \notin \{s(B), t(B)\}$, $d_B(s(B))$ and $d_B(t(B))$ must have the same parity. Thus, vertically flipping B creates no odd-degree vertex.

Next, suppose μ is an S-node. Let $B_1, \ldots, B_k$ be the child blocks of B. Since no B_i contains an odd-degree nonpole, the two poles $s(B_i)$ and $t(B_i)$ must have the same degree parity within B_i (as proved in the last paragraph). Suppose that $d_{B_1}(s(B_1))$ is even. Then, $d_{B_1}(t(B_1))$ must be even. Since $t(B_1) = s(B_2)$ and $d_B(t(B_1)) = d_{B_1}(t(B_1)) + d_{B_2}(s(B_2))$ is even, $d_{B_2}(s(B_2))$ must be even. Continuing this argument, we can see that all $s(B_i)$ and $t(B_i)$ must have even degree within B_i. By this, permuting the children of μ creates no odd-degree vertex. Similarly, if $d_{B_1}(s(B_1))$ is odd, then all $s(B_i)$ and $t(B_i)$ must have odd degree within B_i. Thus, permuting the children of μ creates no odd-degree vertex. □

For a trail P in G and a proper block B of G, the *intersection* of P and B, denoted by $P \cap B$, is the list $(Q_1, Q_2, \ldots, Q_k)$, where Q_1 is the first (maximal) sub-trail of P in B, Q_2 is the second (maximal) sub-trail of P in B, …, Q_k is the last (maximal) sub-trail of P in B. An *ordered* DTC (ODTC) of G is a list $(P_1, \ldots, P_\ell)$ such that $\{P_1, \ldots, P_\ell\}$ is a DTC of G. For an ODTC $\mathcal{P} = (P_1, \ldots, P_\ell)$ of G and a proper block B of G, the *intersection* of $\mathcal{P}$ and B, denoted by $\mathcal{P} \cap B$, is the list $(Q_{1,1}, \ldots, Q_{1,k_1}, \ldots, Q_{\ell,1}, \ldots, Q_{\ell,k_\ell})$, where $(Q_{i,1}, \ldots, Q_{i,k_i})$ is the intersection of P_i and B for all $i \in \{1, \ldots, \ell\}$. Clearly, $\mathcal{P} \cap B$ is an ODTC of B.

Let $\varphi_G : V \to \{1, 2, 3\}$ be the function that maps $s(G)$ to 1, maps $t(G)$ to 2, and maps each nonpole of G to 3. The *starting pair* (respectively, *ending pair*) of a double trail P in G is the pair $(\varphi_G(u), \varphi_{G^*}(x))$, where u is the starting (respectively, ending) vertex of P and x is the starting (respectively, ending) vertex of the dual trail of P. The *extreme quadruple* of a double trail P in G is the quadruple (p_1, p_2, p_3, p_4), where (p_1, p_2) is the starting pair of P and (p_3, p_4) is the ending pair of P. The *σ-extreme list* of an ODTC $(P_1, \ldots, P_k)$ is the list $(q_1, \ldots, q_k)$, where q_i is the extreme quadruple of P_i for all $i \in \{1, \ldots, k\}$. The *π-extreme list* of an ODTC $(P_1, \ldots, P_k)$ is the list $(p_1, q_2, \ldots, q_k)$, where p_1 is the ending pair of P_1 and q_i is the extreme quadruple of P_i for all $i \in \{2, \ldots, k\}$. The *δ-extreme list* of an ODTC $(P_1, \ldots, P_k)$ is the list $(p_1, q_2, \ldots, q_{k-1}, p_k)$, where p_1 is the ending pair of P_1, p_k is the starting pair of P_k, and q_i is the extreme quadruple of P_i for all $i \in \{2, \ldots, k-1\}$.

Lemma 4 *Let B be a block of G. Let $\mathcal{P} = (P_1, \ldots, P_\ell)$ be an ODTC of G. Let D be a TTG, and $\mathcal{Q} = (Q_1, \ldots, Q_k)$ be an ODTC of D. Let H be the TTG obtained from G by replacing B with D. Then, the following three statements hold:*

1. *Suppose that (1) the σ-extreme list of $\mathcal{Q}$ equals that of $\mathcal{P} \cap B$ and (2) no trail in $\mathcal{P}$ starts or ends with an edge of B. Then, H has an ODTC $\mathcal{P}'$ whose σ-extreme list equals that of $\mathcal{P}$.*

2. *Suppose that (1) the π-extreme list of $\mathcal{Q}$ equals that of $\mathcal{P} \cap B$ and (2) P_1 starts with an edge of B while no trail in $\mathcal{P} - \{P_1\}$ starts or ends with an edge of B. Then, H has an ODTC $\mathcal{P}'$ whose π-extreme list equals that of $\mathcal{P}$.*

3. *Suppose that (1) the δ-extreme list of $\mathcal{Q}$ equals that of $\mathcal{P} \cap B$ and (2) P_1 starts with an edge of B and P_ℓ ends with an edge of B while no trail in $\mathcal{P} - \{P_1, P_\ell\}$ starts or ends with an edge of B. Then, H has an ODTC $\mathcal{P}'$ whose δ-extreme list equals that of $\mathcal{P}$.*

Proof: For each $P_i \in \mathcal{P}$, if $P_i \cap B$ is empty, then let $P_i' = P_i$; otherwise, let P_i' be the trail obtained from P_i by replacing each $S_j \in P_i \cap B$ with Q_j, where S_j is the j-th trail in $\mathcal{P} \cap B$. Then, $P' = (P_1', \ldots, P_\ell')$ is the required ODTC of H. $\square$

4 The case where G and G^* are Eulerian

Throughout this section, fix an Eulerian TTG G whose dual graph is also Eulerian. Let T be the decomposition tree of G. By Fact 2, each different embedding $\mathcal{G}$ of G and its dual $\mathcal{G}^*$ are still Eulerian. So, each DET P of a valid embedding of G must start and end at the same vertex, and so does the dual trail of P.

4.1 Types of DTCs of proper blocks

Suppose that G has a valid embedding $\mathcal{G}$ and P is a DET of $\mathcal{G}$. Fix a proper block B of $\mathcal{G}$. Since P starts and ends at the same vertex and so does the dual trail of P, we may assume that the starting edge of P is not in B. Then, each double trail in $P \cap B$ must start and end at poles of B. Let $P \cap B = (P_1, \ldots, P_k)$. For each $i \in \{1, \ldots, k\}$, let P_i^* be the dual trail of P_i.

Lemma 5 1. *If $k \geq 2$, then each $P_i \in \{P_1, \ldots, P_k\}$ uses two distinct access edges of B. Consequently, $k \leq 2$.*

2. *If P_i starts and ends at the same pole of B, then P_i^* must start and end at different poles of B^*.*

3. *If $k = 2$ and P_1 starts and ends at different poles of B, then P_1^* must start and end at the same pole of B^*.*

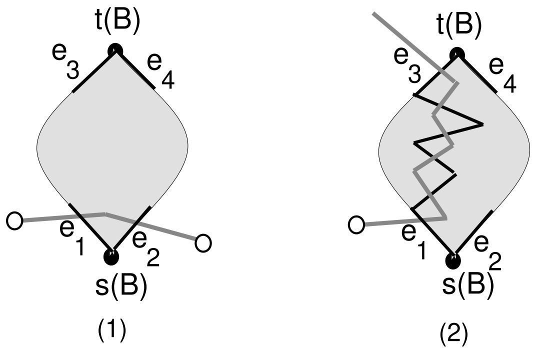

Figure 4: The proof of Lemma 5.

Proof: Let e_1 and e_2 be the two access edges that are incident to $s(B)$, and on the west side and the east side of B respectively. Let e_3 and e_4 be the two access edges that are incident to $t(B)$, and on the west side and the east side of B respectively. We prove the three statements separately as follows.

Statement 1. Suppose $k \geq 2$. Consider an arbitrary $P_i \in \{P_1, \ldots, P_k\}$. Let e (respectively, e') be the starting (respectively, ending) edge of P_i. Since the starting edge of P is not in B, $e \in \{e_1, \ldots, e_4\}$ by the Access-edge property. Even if e' is the ending edge of P, since P is an Euler trail, e' must be incident to the starting edge of P which is not in B by our assumption. So, $e' \in \{e_1, \ldots, e_4\}$. Thus, we are done if $e \neq e'$. Towards a contradiction, assume $e = e'$. Then, $e = \{s(B), t(B)\}$ is the unique edge of P_i. In turn, $\{s(B^*), t(B^*)\}$ is the unique edge of P_i^*. It follows that B consists of a single edge, contradicting the assumption that $k \geq 2$. Therefore, $e \neq e'$ and P_i uses two distinct access edges of B. Since B has at most four access edges (namely e_1 through e_4), $k \leq 2$.

Statement 2. Without loss of generality, suppose that P_i starts and ends at $s(B)$. Then, $e_1 \neq e_2$. By the Access-edge property, we can assume that P_i starts with e_1 and ends with e_2. Hence, P_i^* starts at the west side, and ends at the east side of B. (See Figure 4(1).)

Statement 3. Suppose that $k = 2$ and P_1 starts and ends at different poles of B. Then, by Statement 1, e_1 through e_4 are distinct edges. Without loss of generality, assume that P_1 starts with e_1. Towards a contradiction, suppose that P_1 ends with e_4. Then, P_2 starts with e_2 and ends with e_3. Since B is a plane graph, P_1 and P_2 must cross each other somewhere inside B. However, this is impossible by the Non-crossing property. Hence, P_1 ends with e_3; in turn P_1^* starts and ends at the west side of B. (See Figure 4(2).) □

Based on Lemma 5, $P \cap B$ can only be of the following five possible types:

- *Type* σ_1: $k = 1$; P_1 starts at one pole and ends at the other pole, and so does P_1^*.
- *Type* σ_2: $k = 1$; P_1 starts at one pole and ends at the same pole while P_1^* starts at one pole and ends at the other pole.
- *Type* σ_3: $k = 1$; P_1 starts at one pole and ends at the other pole while P_1^* starts at one pole and ends at the same pole.
- *Type* σ_4: $k = 2$; P_1 starts at one pole u and ends at u while P_2 starts at the other pole v and ends at v; P_1^* starts at one pole and ends at the other pole, and so does P_2^*.

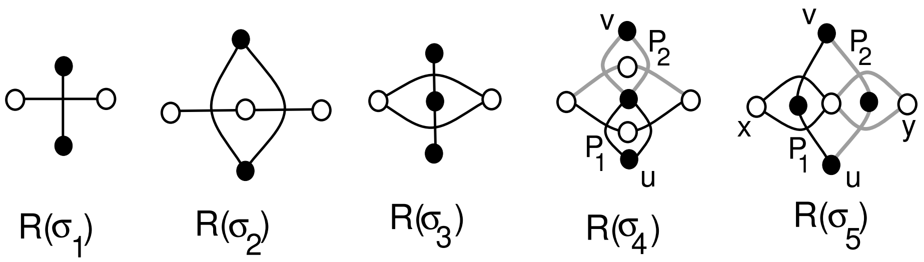

Figure 5: Representative graphs for σ-types.

- *Type* σ_5: $k = 2$; P_1 starts at one pole and ends at the other pole, and so does P_2; P_1^* starts at one pole x and ends at x while P_2^* starts at the other pole y and ends at y.

Define $\Sigma = \{\sigma_1, \ldots, \sigma_5\}$. Each σ_i is called a *σ-type.* For a TTG H, let $\mathcal{K}_\sigma(H)$ denote the set of all $\sigma_i \in \Sigma$ such that a different embedding of H has a Type-σ_i ODTC. We call each member of $\mathcal{K}_\sigma(H)$ a *σ-type of* H. For each $\sigma_i \in \Sigma$, a *representative graph* $R(\sigma_i)$ is shown in Figure 5 in which the solid lines denote the edges in the trail P_1 and the dual trail P_1^*, while the light lines denote the edges in the trail P_2 and the dual trail P_2^*. Note that σ_i is the unique σ-type of $R(\sigma_i)$ for all $i \in \{1, \ldots, 5\}$.

4.2 Composing DTCs

Let H_1 and H_2 be two vertex-disjoint TTGs. The *series composition* of H_1 and H_2, denoted by $H_1 \star H_2$, is the TTG obtained from H_1 and H_2 by identifying $t(H_1)$ with $s(H_2)$, $s(H_1^*)$ with $s(H_2^*)$, and $t(H_1^*)$ with $t(H_2^*)$. The *parallel composition* of H_1 and H_2, denoted by $H_1 \diamond H_2$, is the TTG obtained from H_1 and H_2 by identifying $s(H_1)$ with $s(H_2)$, $t(H_1)$ with $t(H_2)$, and $t(H_1^*)$ with $s(H_2^*)$. For an integer $k \geq 3$ and k vertex-disjoint TTGs $H_1, \ldots, H_k$, let $H_1 \star H_2 \star \cdots \star H_k = (\cdots((H_1 \star H_2) \star H_3) \star \cdots H_{k-1}) \star H_k$ and $H_1 \diamond H_2 \diamond \cdots \diamond H_k = (\cdots((H_1 \diamond H_2) \diamond H_3) \diamond \cdots H_{k-1}) \diamond H_k$. We call $H_1 \star \cdots \star H_k$ (respectively, $H_1 \diamond \cdots \diamond H_k$) the *series* (respectively, *parallel*) *composition* of H_1 through H_k.

Define two binary operators $\star$ and $\diamond$ over Σ as in Tables 1 and 2; each empty entry in Table 1 (respectively, 2) indicates that the $\star$ (respectively, $\diamond$) operator is undefined for that entry.

	σ_1	σ_2	σ_3	σ_4	σ_5
σ_1	σ_3		σ_1		
σ_2		σ_4			σ_2
σ_3	σ_1		σ_3		
σ_4					σ_4
σ_5		σ_2		σ_4	σ_5

Table 1: Operator $\star$ over Σ

Lemma 6 *Both $\star$ and $\diamond$ are commutative and associative operators over Σ.*

Proof: One can use Tables 1 and 2 to verify. □

	σ_1	σ_2	σ_3	σ_4	σ_5
σ_1	σ_2	σ_1			
σ_2	σ_1	σ_2			
σ_3			σ_5	σ_3	
σ_4			σ_3	σ_4	σ_5
σ_5				σ_5	

Table 2: Operator $\diamond$ over Σ

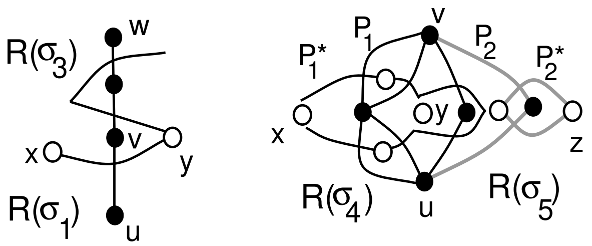

Figure 6: The proof of Lemma 7

.

Lemma 7 *Let H be a TTG. Then, the following two statements hold:*

1. *Suppose $H = H_1 \star H_2$ where $\mathcal{K}_\sigma(H_1) = \{\sigma_i\}$ and $\mathcal{K}_\sigma(H_2) = \{\sigma_j\}$. Then, $\mathcal{K}_\sigma(H) = \{\sigma_i \star \sigma_j\}$ if $\sigma_i \star \sigma_j$ is defined, while $\mathcal{K}_\sigma(H) = \emptyset$ otherwise.*

2. *Suppose $H = H_1 \diamond H_2$ where $\mathcal{K}_\sigma(H_1) = \{\sigma_i\}$ and $\mathcal{K}_\sigma(H_2) = \{\sigma_j\}$. Then, $\mathcal{K}_\sigma(H) = \{\sigma_i \diamond \sigma_j\}$ if $\sigma_i \diamond \sigma_j$ is defined, while $\mathcal{K}_\sigma(H) = \emptyset$ otherwise.*

Proof: By Statement 1 in Lemma 4, it suffices to verify that for every $\sigma_i, \sigma_j \in \Sigma$, $\mathcal{K}_\sigma(R(\sigma_i) \star R(\sigma_j)) = \{\sigma_i \star \sigma_j\}$ and $\mathcal{K}_\sigma(R(\sigma_i) \diamond R(\sigma_j)) = \{\sigma_i \diamond \sigma_j\}$. For the verification, it is helpful to note that for each $\sigma_i \in \Sigma$, $R(\sigma_i)$ remains unchanged after a horizontal or vertical flipping. We only show one example for each of the $\star$ and the $\diamond$ operators. The other cases can be verified similarly.

Consider the graph $R(\sigma_1)$ with poles u and v, and the graph $R(\sigma_3)$ with poles v and w. Let x and y be the west side and the east side of their series composition $H' = R(\sigma_1) \star R(\sigma_3)$. (See Figure 6(1).) The only DTC of H' consists of a single double trail P_1 which starts at u, passing through v and ends at w, while the dual trail P_1^* starts at x, travels to y, to x, and to y again. This is a Type-σ_1 DTC. So $\mathcal{K}_\sigma(R(\sigma_1) \star R(\sigma_3)) = \{\sigma_1\} = \{\sigma_1 \star \sigma_3\}$ as to be shown.

Consider the graph $R(\sigma_4)$ with poles u, v, the wast side x and the east side y; and the graph $R(\sigma_5)$ with poles u, v, the west side y and the east side z. (See Figure 6(2).) Let $H'' = R(\sigma_4) \diamond R(\sigma_5)$. The only DTC of H'' consists of two double trails P_1 and P_2 as shown in Figure 6(2). This is a Type-σ_4 DTC. So $\mathcal{K}_\sigma(R(\sigma_4) \diamond R(\sigma_5)) = \{\sigma_5\} = \{\sigma_4 \diamond \sigma_5\}$. $\square$

Let μ be a node in T and $B = \mathrm{Pt}(\mu)$. For convenience, we also use $\mathcal{K}_\sigma(\mu)$ to denote $\mathcal{K}_\sigma(B)$.

Lemma 8 *Let μ be a non-root node of T and $B = \mathrm{Pt}(\mu)$. Then, $|\mathcal{K}_\sigma(B)| \leq 1$. Moreover, $\mathcal{K}_\sigma(\mathrm{Pt}(\nu))$ for all descendants ν (including μ itself) of μ in T can be computed in $O(|B|)$ time.*

Proof: By induction. In case μ is a leaf node of T, the lemma trivially holds because $\mathcal{K}_\sigma(B) = \{\sigma_1\}$. Suppose that μ is a non-leaf node of T. Let $\nu_1, \ldots, \nu_k$ be the children of μ in T. Let $B_i = \mathrm{Pt}(\nu_i)$ for all $i \in \{1, \ldots, k\}$. By the inductive hypothesis, $|\mathcal{K}_\sigma(B_i)| \leq 1$. If $\mathcal{K}_\sigma(B_i) = \emptyset$ for some $i \in \{1, \ldots, k\}$, then clearly $\mathcal{K}_\sigma(B) = \emptyset$. So, suppose that each $\mathcal{K}_\sigma(B_i)$ consists of a unique $\sigma_{j_i} \in \Sigma$. Consider the following three cases:

Case 1: μ is an R-node. Then, $\mathrm{Sk}(\mu)$ has no different embedding other than itself. Let H be the TTG obtained from $\mathrm{Sk}(\mu)$ by replacing the virtual edge of ν_i in $\mathrm{Sk}(\mu)$ with $R(\sigma_{j_i})$ for all $i \in \{1, \ldots, k\}$. Since $\mathcal{K}_\sigma(B_i) = \mathcal{K}_\sigma(R(\sigma_{j_i}))$ for all $i \in \{1, \ldots, k\}$, by repeatedly applying Statement 1 in Lemma 4, we have $\mathcal{K}_\sigma(\mu) = \mathcal{K}_\sigma(H)$. $\mathcal{K}_\sigma(H)$ can be computed as follows.

Let P_1 be the maximal double trail in H which starts at $s(H)$ and whose dual trail P_1^* starts at $s(H^*)$. Let M be the set of all edges of H not traversed by P_1. In case $M = \emptyset$, we check if $\{P_1\}$ is a DTC of H of a σ-type; if it is, then $\mathcal{K}_\sigma(\mu)$ consists of this σ-type, while $\mathcal{K}_\sigma(\mu) = \emptyset$ otherwise. Suppose $M \neq \emptyset$. If M contains no access edge of H, then $\mathcal{K}_\sigma(\mu) = \emptyset$. Otherwise, let $e \in M$ be an access edge of H. We find the maximal double trail P_2 in H such that e and e^* are respectively the starting edge of P_2 and its dual trail P_2^*. If $\{P_1, P_2\}$ is a DTC of H of a σ-type, then $\mathcal{K}_\sigma(\mu)$ consists of this σ-type, while $\mathcal{K}_\sigma(\mu) = \emptyset$ otherwise.

Case 2: μ is an S-node. Then, each different embedding $\mathcal{B}$ of B can be written as $\mathcal{B}_{1'} \star \cdots \star \mathcal{B}_{k'}$, where $(1', 2', \ldots, k')$ is a permutation of $(1, 2, \ldots, k)$ and $\mathcal{B}_{i'}$ is a different embedding of $B_{i'}$ for all $i \in \{1, \ldots, k\}$. By Lemma 7, if $(\cdots((\sigma_{j_{1'}} \star \sigma_{j_{2'}}) \star \sigma_{j_{3'}}) \star \cdots) \star \sigma_{j_{k'}}$ is undefined, then $\mathcal{K}_\sigma(\mathcal{B}) = \emptyset$; otherwise, by Lemma 6, $\mathcal{K}_\sigma(\mathcal{B}) = \{(\cdots((\sigma_{j_1} \star \sigma_{j_2}) \star \sigma_{j_3}) \star \cdots) \star \sigma_{j_k}\}$. Thus, $|\mathcal{K}_\sigma(B)| \leq 1$.

Case 3: μ is a P-node. Similar to Case 2. □

4.3 The algorithm

We process the nodes of T in post-order. When processing a non-root node μ of T, we compute $\mathcal{K}_\sigma(\mu)$ as in the proof of Lemma 8. Let μ be the root of T. Let $\nu_1, \ldots, \nu_k$ be the children of μ in T. For each $i \in \{1, \ldots, k\}$, let $B_i = \mathrm{Pt}(\nu_i)$. If $\mathcal{K}_\sigma(B_i) = \emptyset$ for some $i \in \{1, \ldots, k\}$, then G has no valid embedding and we stop immediately. So, suppose that $\mathcal{K}_\sigma(B_1) = \{\sigma_{j_1}\}, \ldots, \mathcal{K}_\sigma(B_k) = \{\sigma_{j_k}\}$. We distinguish three cases as follows.

- *Case 1:* μ is an R-node. Then, $\mathrm{Sk}(\mu)$ has no different embedding other than itself. As in Case 1 in the proof of Lemma 8, we construct a TTG H from $\mathrm{Sk}(\mu)$. By the same argument as in Case 1 in the proof of Lemma 8, G has a DET if and only if H has a DET.

  To test if H has a DET, we pick an arbitrary edge e in H and fix a direction as the starting point. The dual edge e^* can be directed in two ways. Once the direction of e^* is fixed, we can uniquely trace the dual trail by Lemma 3.

  More specifically, let e be the edge incident to $s(B)$ whose dual edge e^* is incident to $s(H^*)$. Let x be the endpoint of e^* other than $s(H^*)$. Let P (respectively, Q) be the maximal double trail in H which starts at $s(H)$ and whose dual trail starts at $s(H^*)$ (respectively, x). By Lemma 3, P and Q can be found in $O(|\mathrm{Sk}(\mu)|)$ time. By Statement 1 in Lemma 4, if either P or Q is a DET in H, then G has a DET. If neither P nor Q is a DET in H, then H has no DET, and hence G has no DET either.

- *Case 2:* μ is an S-node. First we prove the following claim:

  Claim: For a TTG $H = H_1 \star H_2$ with $|\mathcal{K}_\sigma(H_1)| \leq 1$ and $|\mathcal{K}_\sigma(H_2)| \leq 1$, H has a valid embedding if and only if $\mathcal{K}_\sigma(H_1) = \mathcal{K}_\sigma(H_2) = \{\sigma_2\}$.

  To see the "if" part of the claim, it suffices to observe that a series composition of two copies of $R(\sigma_2)$ has a valid embedding. To see the "only if" part, suppose that H has a DET P. Then, P is obtained by concatenating the double trail(s) in the DTC of H_1 and the double trail(s) in the DTC of H_2. This can be done only in the following two possible ways (a) and (b):

  (a) The DTC of H_1 consists of a single double trail P_1 starting and ending at the same pole of H_1, and the DTC of H_2 consists of a single double trail P_2 starting and ending at the same pole of H_2. That is $\mathcal{K}_\sigma(H_1) = \mathcal{K}_\sigma(H_2) = \{\sigma_2\}$. In this case, if H is obtained by identifying the common pole where both P_1 and P_2 start and end, we indeed get a DET of H.

  (b) The DTC of H_1 consists of a single double trail P_1 starting and ending at the same pole of H_1, and the DTC of H_2 consists of two double trails P_2 and P_3, each starting and ending at the different poles of H_2. (In this case, by concatenating P_2, P_1, P_3 in this order, we get a closed Euler trail P of H). That is $\mathcal{K}_\sigma(H_1) = \{\sigma_2\}$ and $\mathcal{K}_\sigma(H_2) = \{\sigma_5\}$. In this case, however, the dual trail P^* of P starts at one pole of the dual graph H^*, and ends at the other pole of H^*. (This can be verified by using the representative graphs in Figure 5). Hence, P^* is not a closed Euler trail of H^*. So, this case is impossible. This completes the proof of the claim.

  Observe that for each nonempty proper subset $\{1', \ldots, h'\}$ of $\{1, \ldots, k\}$, $B_{1'} \star \cdots \star B_{h'}$ has at most one σ-type. This follows from Lemmas 6 and 7. By the claim and this observation, G has a valid embedding if and only if $\{1, \ldots, k\}$ can be partitioned into two sets $\{1', \ldots, h'\}$, $\{1'', \ldots, g''\}$ such that σ_2 is the σ-type of both $B_{1'} \star \cdots \star B_{h'}$ and $B_{1''} \star \cdots \star B_{g''}$. By Table 1, such a partition is possible if and only if (1) σ_2 is the σ-type of exactly two of $B_1, \ldots, B_k$ and (2) σ_5 is the σ-type of all the rest.

- *Case 3:* μ is a P-node. It suffices to modify the argument in Case 2 by replacing each symbol "$\star$" with "$\diamond$", "σ_2" with "σ_3", and "σ_5" with "σ_4".

By Lemmas 1 and 8, the time needed to process the entire tree T is $O(|G|)$. Thus, we have:

Theorem 1 *Given an Eulerian TTG G whose dual graph is also Eulerian, it takes $O(|G|)$ time to decide if G has a valid embedding.*

5 An $O(|G|^3)$-time algorithm

Throughout the rest of this paper, let G be a TTG, and T its decomposition tree. The *reverse* of a trail $e_1 e_2 \ldots e_k$ in G is the trail $e_k e_{k-1} \ldots e_1$. Section 5.1 investigates how a DET of G can intersect a block of G. Section 5.2 (respectively, 5.3) shows how a DET P can intersect the series (respectively, parallel) composition of two or more blocks of G, given the way that P intersects the blocks. Section 5.4 describes the algorithm for deciding the existence of a DET.

5.1 Types of DTCs of proper blocks

Suppose that G has a valid embedding $\mathcal{G}$ and P is a DET of $\mathcal{G}$. Let e_s and e_t be the starting and the ending edge of P, respectively. Fix a proper block B of $\mathcal{G}$. Let $P \cap B = (P_1, \ldots, P_k)$. For each $i \in \{1, \ldots, k\}$, let P_i^* be the dual trail of P_i.

Case 1: neither e_s nor e_t is in B. Then, as discussed in Section 4.1, $P \cap B$ can only be of a σ-type.

Case 2: exactly one of e_s and e_t is in B. If e_s is in B, let $Q_i = P_i$ for all $i \in \{1, \ldots, k\}$; otherwise, let Q_i be the reverse of P_{k-i+1} for all $i \in \{1, \ldots, k\}$. Let Q_i^* be the dual trail of Q_i for each $i \in \{1, \ldots, k\}$.

Without loss of generality, suppose that Q_1 starts with e_s and ends with an access edge of B that is incident to a pole u of B. Similarly to the proof of Statement 1 in Lemma 5, we can show that each $Q_i \in \{Q_2, \ldots, Q_k\}$ must use exactly two access edges of B. Since B has at most 4 access edges, k must be either 1 or 2. If $k = 2$, the trail Q_2 or its reverse must either starts with an access edge incident to u and ends with an access edge incident to the other pole v of B; or starts and ends with the two access edges incident to v. Based on these observations, the ODTC $(Q_1, \ldots, Q_k)$ can only be of the following possible types:

- *Type* π_1: $k = 1$; Q_1 ends at a pole; Q_1^* ends at a pole.
- *Type* π_2: $k = 2$; Q_1 ends at a pole, and Q_2 starts at a pole and ends at the other pole; Q_1^* ends at a pole, and Q_2^* starts at a pole and ends at the other pole.
- *Type* π_3: $k = 2$; Q_1 ends at a pole, and Q_2 starts at a pole and ends at the other pole; Q_1^* ends at a pole x, and Q_2^* starts at the other pole y and ends at y.
- *Type* π_4: $k = 2$; Q_1 ends at a pole u, and Q_2 starts at the other pole v and ends at v; Q_1^* ends at a pole, and Q_2^* starts at a pole and ends at the other pole.

Note that for each type π_i, the trail Q_1 is required to end at a pole, and starts with the edge e_s. However, the edge e_s might be either incident to a pole, or not incident to a pole.

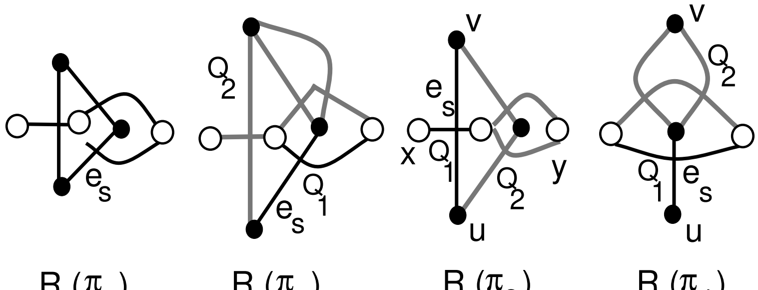

Figure 7: Representative graphs for π-types.

Define $\Pi = \{\pi_1, \ldots, \pi_4\}$. We call the elements of Π *π-types*. For a TTG H, let $\mathcal{K}_\pi(H)$ denote the set of all $\pi_i \in \Pi$ such that a different embedding of H has a Type-π_i DTC. We call each member of $\mathcal{K}_\pi(H)$ a *π-type of H*. For each $\pi_i \in \Pi$, a *representative graph* $R(\pi_i)$ is shown in Figure 7. Note that for all $i \in \{1, \ldots, 4\}$, π_i is the unique π-type of $R(\pi_i)$.

Case 3: both e_s and e_t are in B. Then, $k > 1$ or else B could not have been a proper block of G. Without loss of generality, suppose that P_1 starts with e_s and P_k ends with e_t. Each of P_1 and P_k uses at least one access edge of B. Similarly to the proof of Statement 1 in Lemma 5, we can show that each $P_i \in \{P_2, \ldots, P_{k-1}\}$ must use exactly two access edges of B. Thus, k must be either 2 or 3. By considering which access edges are used by which P_i, it can be seen that $P \cap B$ can only be one of the following possible types:

- *Type* δ_1: $k = 2$; P_1 ends at a pole, and P_2 starts at the same pole; P_1^* ends at a pole, and P_2^* starts at the other pole.
- *Type* δ_2: $k = 2$; P_1 ends at a pole, and P_2 starts at the other pole; P_1^* ends at a pole, and P_2^* starts at the same pole.
- *Type* δ_3: $k = 2$; P_1 ends at a pole, and P_2 starts at the other pole; P_1^* ends at a pole, and P_2^* starts at the other pole.
- *Type* δ_4: $k = 3$; P_1 ends at a pole u, P_2 starts at the other pole v and ends at v, and P_3 starts at u; P_1^* ends at a pole x, P_2^* starts at x and ends at the other pole y, and P_3^* starts at y.
- *Type* δ_5: $k = 3$; P_1 ends at a pole u, P_2 starts at u and ends at the other pole v, and P_3 starts at v; P_1^* ends at a pole x, P_2^* starts at the other pole y and ends at y, and P_3^* starts at x.
- *Type* δ_6: $k = 3$; P_1 ends at a pole u, P_2 starts at a pole and ends at the other pole, and P_3 starts at the pole $v \neq u$; P_1^* ends at a pole x, P_2^* starts at a pole and ends at the other pole, and P_3^* starts at the pole $y \neq x$.

Note that for each type δ_i, the trail P_1 is required to end at a pole, and starts with the edge e_s. However, e_s might be either incident to a pole, or not incident to a pole. The trail P_2 in the types $\delta_1, \delta_2, \delta_3$ (the trail P_3 in the types $\delta_4, \delta_5, \delta_6$, respectively) is required to starts at a pole and ends with the edge e_t. However, e_t might be either incident to a pole, or not incident to a pole.

Define $\Delta = \{\delta_1, \ldots, \delta_6\}$. We note that when $B = G$, $P \cap B$ must be of the following type:

- *Type* δ_0: $k = 1$.

We call the elements of $\Delta \cup \{\delta_0\}$ *δ-types.* For a TTG H, let $\mathcal{K}_\delta(H)$ denote the set of all $\delta_i \in \Delta \cup \{\delta_0\}$ such that a different embedding of H has a Type-δ_i DTC. We call each member of $\mathcal{K}_\delta(H)$ a *δ-type of H*. For each $\delta_i \in \Delta$, a *representative graph* $R(\delta_i)$ is shown in Figure 8. Note that for all $i \in \{1, \ldots, 6\}$, δ_i is the unique δ-type of $R(\delta_i)$.

Define $\Gamma = \Sigma \cup \Pi \cup \Delta$. Let H be a representative graph in Figures 7 and 8. If $\mathcal{Q}_1$ and $\mathcal{Q}_2$ are two DTCs of H such that $\mathcal{Q}_2$ can be obtained from $\mathcal{Q}_1$ by reversing the direction of exactly one double trail on which neither e_s nor e_t appears, then we view $\mathcal{Q}_1$ and $\mathcal{Q}_2$ as the same DTC. A DTC $\mathcal{Q}$ of H is *valid* if $\mathcal{Q}$ is of a σ-, π-, or δ-type. From the figures, we can see that H has a unique valid DTC. Thus, if H is a graph in Figure 7, then we think that e_s and its dual edge have been oriented by the valid DTC of H. Similarly, if H is a graph in Figure 8, then we think that e_s and e_t and their dual edges have been oriented by the valid DTC of H.

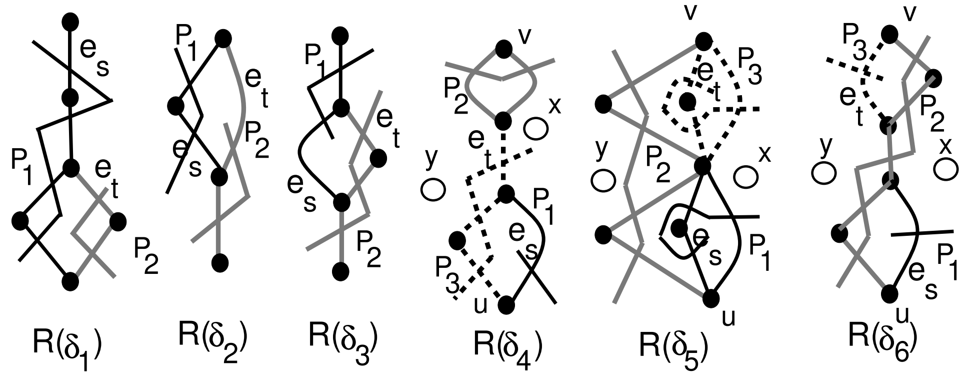

Figure 8: Representative graphs for δ-types.

5.2 Series composition of DTCs

This subsection shows how to compute the π- or δ-types of a series composition of two or more TTGs $H_1, \ldots, H_k$, when the σ-, π-, or δ-types of H_1 through H_k are given. This is needed by our algorithm to process the S-nodes of T. Let $\Gamma_1 = \Sigma_1 \cup \Pi_1 \cup \Delta_1$ and $\Gamma_2 = \Sigma_2 \cup \Pi_2 \cup \Delta_2$ be two subsets of Γ, where for $i = 1, 2$, $\Sigma_i \subseteq \Sigma$, $\Pi_i \subseteq \Pi$, and $\Delta_i \subseteq \Delta$. Define:

$$\begin{aligned}
\Gamma_1 \star \Gamma_2 \quad = \quad & (\cup_{\sigma_i \in \Sigma_1, \sigma_j \in \Sigma_2} \mathcal{K}_\sigma(R(\sigma_i) \star R(\sigma_j))) \cup (\cup_{\sigma_i \in \Sigma_1, \pi_j \in \Pi_2} \mathcal{K}_\pi(R(\sigma_i) \star R(\pi_j))) \cup \\
& (\cup_{\pi_i \in \Pi_1, \sigma_j \in \Sigma_2} \mathcal{K}_\pi(R(\pi_i) \star R(\sigma_j))) \cup (\cup_{\sigma_i \in \Sigma_1, \delta_j \in \Delta_2} \mathcal{K}_\delta(R(\sigma_i) \star R(\delta_j))) \cup \\
& (\cup_{\delta_i \in \Delta_1, \sigma_j \in \Sigma_2} \mathcal{K}_\delta(R(\delta_i) \star R(\sigma_j))) \cup (\cup_{\pi_i \in \Pi_1, \pi_j \in \Pi_2} \mathcal{K}_\delta(R(\pi_i) \star R(\pi_j)))
\end{aligned}$$

If $\Gamma_1 = \emptyset$ or $\Gamma_2 = \emptyset$, then $\Gamma_1 \star \Gamma_2 = \emptyset$.

Recall that, by the definition, $\mathcal{K}_\sigma(H)$ is the set of $\sigma_i \in \Sigma$ such that a different embedding of H has a type-σ_i ODTC. In the case $H = H_1 \star H_2$, if we switch H_1 and H_2, the graph $H_2 \star H_1$ is a different embedding of H, and hence the σ-types of $H_1 \star H_2$ and $H_2 \star H_1$ both belong to $\mathcal{K}_\sigma(H)$. This remark is also true for $\mathcal{K}_\pi(H)$ and $\mathcal{K}_\delta(H)$. Thus, the operator $\star$ defined on Γ_1 and Γ_2 is trivially commutative by its definition. For later reference, we state this in the following lemma.

Lemma 9 *If Γ_1 and Γ_2 are two subsets of Γ, then $\Gamma_1 \star \Gamma_2 = \Gamma_2 \star \Gamma_1$.*

For each $\sigma_i \in \Sigma$ and each $\delta_j \in \Delta$, the (i, j)-entry of Table 3 contains all $\delta_k \in \{\sigma_i\} \star \{\delta_j\}$. For each $\pi_i \in \Pi$ and each $\sigma_j \in \Sigma$, the (i, j)-entry of Table 4 contains all $\pi_k \in \{\pi_i\} \star \{\sigma_j\}$. For each $\pi_i \in \Pi$ and each $\pi_j \in \Pi$, the (i, j)-entry of Table 5 contains all $\delta_k \in \{\pi_i\} \star \{\pi_j\}$. These tables can be verified by inspecting the representative graphs in Figures 5, 7, and 8. As examples, we verify one entry from each of Tables 3, 4 and 5. They are shown in Figure 9. In each case, two representative graphs are composed in series by identifying the common pole v. The type of the resulting graph is shown in the figure.

Suppose that $H = H_1 \star H_2 \star \ldots \star H_k$ and that $\mathcal{K}_\sigma(H_i)$, $\mathcal{K}_\pi(H_i)$ and $\mathcal{K}_\delta(H_i)$ have already been computed for all $i \in \{1, \ldots, k\}$. Our goal is to calculate $\mathcal{K}_\sigma(H)$, $\mathcal{K}_\pi(H)$ and $\mathcal{K}_\delta(H)$ for the series composition H. $\mathcal{K}_\sigma(H)$ can be computed as in Section 4.2. By definition, $\mathcal{K}_\pi(H)$ contains all the π-types of all different embeddings of H. Essentially, this means that the computation of $\mathcal{K}_\pi(H)$ requires the evaluation of all expressions consisting of one operand from some $\mathcal{K}_\pi(H_i)$

	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6
σ_1		δ_3	δ_2			
σ_2	$\delta_0,\ \delta_4$				δ_1	δ_1
σ_3		δ_2	δ_3			
σ_4					δ_4	δ_4
σ_5	δ_1			δ_4	δ_5	δ_6

Table 3: $\{\sigma_i\} \star \{\delta_j\}$.

	σ_1	σ_2	σ_3	σ_4	σ_5
π_1	π_1	π_4	π_1		
π_2	π_3	π_1	π_2	π_4	π_2
π_3	π_2	π_1	π_3	π_4	π_3
π_4	π_4		π_4		π_4

Table 4: $\{\pi_i\} \star \{\sigma_j\}$.

	π_1	π_2	π_3	π_4
π_1	$\delta_0,\ \delta_2,\ \delta_3$	δ_1	δ_1	
π_2	δ_1	$\delta_3,\ \delta_5$	$\delta_2,\ \delta_6$	$\delta_2,\ \delta_3,\ \delta_4$
π_3	δ_1	$\delta_2,\ \delta_6$	$\delta_3,\ \delta_5$	$\delta_2,\ \delta_3,\ \delta_4$
π_4		$\delta_2,\ \delta_3,\ \delta_4$	$\delta_2,\ \delta_3,\ \delta_4$	

Table 5: $\{\pi_i\} \star \{\pi_j\}$.

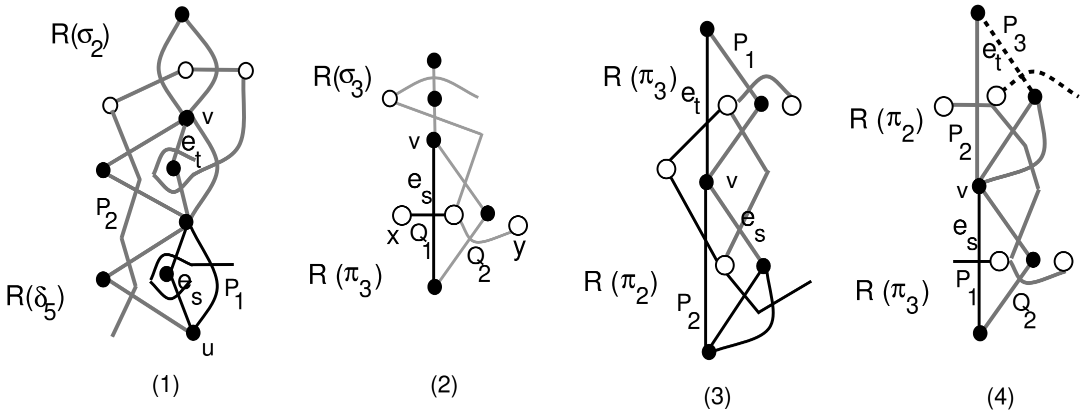

Figure 9: (1) $\sigma_2 \star \delta_5 = \delta_1$, (2) $\pi_3 \star \sigma_3 = \pi_3$, (3) $\pi_2 \star \pi_3 = \delta_2$, (4) $\pi_2 \star \pi_3 = \delta_6$.

and one operand from each of $\mathcal{K}_\sigma(H_1), \ldots, \mathcal{K}_\sigma(H_{i-1}), \mathcal{K}_\sigma(H_{i+1}), \ldots, \mathcal{K}_\sigma(H_k)$ connected by the operator $\star$. Although the operator $\star$ is commutative, it is not associative. This makes the calculation of $\mathcal{K}_\pi(H)$ very complex. The calculation of $\mathcal{K}_\delta(H)$ is even more complex. The rest of this subsection discusses how to perform these calculations.

A $\star$-*expression* E, its *value*, and its *operand multiset* $\mathrm{Op}(E)$ are defined inductively as follows. Each $\gamma \in \Gamma$ is a $\star$-expression, its value is $\{\gamma\}$, and its operand multiset is $\langle\gamma\rangle$. Each (possibly empty) $\Gamma_1 \subseteq \Gamma$ is a $\star$-expression, its value is Γ_1, and its operand multiset is $\langle\Gamma_1\rangle$. If E_1 and E_2 are $\star$-expressions, then $(E_1 \star E_2)$ is a $\star$-expression, its operand multiset is $\mathrm{Op}(E_1) \cup \mathrm{Op}(E_2)$, its value is $(\Gamma_1 \star \Gamma_2)$ where Γ_1 and Γ_2 are the value of E_1 and E_2, respectively.

We identify a $\star$-expression with its value. For a list $(X_1, \ldots, X_k)$ of elements or subsets of Γ, let $X_1 \star X_2 \star \cdots \star X_k$ denote the $\star$-expression $(\cdots((X_1 \star X_2) \star X_3) \star \cdots X_k)$. For a subset Γ_1 of Γ and a list $L = (\sigma_{i_1}, \ldots, \sigma_{i_k})$ of σ-types, let $\Gamma_1 \star [L]_\star$ denote the $\star$-expression $\Gamma_1 \star \sigma_{i_1} \star \cdots \star \sigma_{i_k}$ if $k \geq 1$; and denote Γ_1 otherwise. For a $\sigma_i \in \Sigma$ and a nonnegative integer k, let σ_i^k denote the list of k σ_i's.

The *parse tree* T_E of a $\star$-expression E is defined inductively as follows. If $|\mathrm{Op}(E)| = 1$, then T_E consists of a single node labeled with the unique element of $\mathrm{Op}(E)$. If $E = (E_1 \star E_2)$, then T_E is obtained from the parse tree T_{E_1} of E_1 and the parse tree T_{E_2} of E_2 by introducing a new node r and letting the roots of T_{E_1} and T_{E_2} be the children of r.

For convenience, each node v in a parse tree T_E is viewed as an ancestor of v itself in T_E.

Lemma 10 *Suppose $\Delta_1 \subseteq \Delta$. Let $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Then, the following two statements hold:*

1. *For every $\star$-expression E with $\mathrm{Op}(E) = \langle\sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1\rangle$, if σ_2 is not in L, then $E \subseteq \Delta_1 \star [L]_\star$; otherwise, $E \subseteq \Delta_1 \star [L - \{\sigma_5\}]_\star$.*

2. *Let U be the union of the values of all $\star$-expressions E with $\mathrm{Op}(E) = \langle\sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1\rangle$. If σ_2 is not in L, then $U = \Delta_1 \star [L]_\star$; otherwise, $U = \Delta_1 \star [L - \{\sigma_5\}]_\star$.*

Proof: We prove the two statements separately as follows.

Statement 1. First, we claim that if σ_i and σ_j are (possibly the same) elements of Σ with $\{\sigma_i, \sigma_j\} \neq \{\sigma_2, \sigma_5\}$, then $\Delta_1 \star (\sigma_i \star \sigma_j) \subseteq \Delta_1 \star \sigma_i \star \sigma_j = \Delta_1 \star \sigma_j \star \sigma_i$. One may verify this claim using Tables 1 and 3, under the assumption that $|\Delta_1| = 1$. This claim then holds for every $\Delta_1 \subseteq \Delta$.

Second, we claim that $(\Delta_1 \star \sigma_5 \star \sigma_2) \cup (\Delta_1 \star \sigma_2 \star \sigma_5) \subseteq \Delta_1 \star (\sigma_5 \star \sigma_2) = \Delta_1 \star \sigma_2$. One may verify this claim using Tables 1 and 3, under the assumption that $|\Delta_1| = 1$. This claim then holds for every $\Delta_1 \subseteq \Delta$.

Third, we claim that for every nonempty list L of σ-types and every permutation L' of L, if σ_2 is not in L, then $\Delta_1 \star [L']_\star \subseteq \Delta_1 \star [L]_\star$; otherwise, $\Delta_1 \star [L']_\star \subseteq \Delta_1 \star [L - \{\sigma_5\}]_\star$. This is shown by induction on the number q of elements of L as follows. The claim trivially holds when $q = 1$. Let $q \geq 2$ and $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Let $L' = (\sigma_{j_1}, \ldots, \sigma_{j_q})$ be a permutation of L. If σ_5 or σ_2 does not occur in L, then by the first claim, $\Delta_1 \star [L']_\star \subseteq \Delta_1 \star [L]_\star$. So, suppose that both σ_5 and σ_2 occur in L. Consider a sublist $(\sigma_{j_k}, \sigma_{j_{k+1}}, \ldots, \sigma_{j_\ell})$ of L' such that $\{\sigma_{j_k}, \sigma_{j_\ell}\} = \{\sigma_5, \sigma_2\}$ and $\{\sigma_5, \sigma_2\} \cap \{\sigma_{j_{k+1}}, \ldots, \sigma_{j_{\ell-1}}\} = \emptyset$. By the first claim, $\Delta_1 \star [L']_\star = \Delta_1 \star \sigma_{j_1} \star \cdots \star \sigma_{j_k} \star \sigma_{j_\ell} \star \sigma_{j_{k+1}} \star \cdots \star \sigma_{j_{\ell-1}} \star \sigma_{j_{\ell+1}} \star \cdots \star \sigma_{j_q}$. This together with the second claim implies that $\Delta_1 \star [L']_\star \subseteq \Delta_1 \star \sigma_{j_1} \star \cdots \star \sigma_{j_{k-1}} \star \sigma_2 \star \sigma_{j_{k+1}} \star \cdots \star \sigma_{j_{\ell-1}} \star \sigma_{j_{\ell+1}} \star \cdots \star \sigma_{j_q}$. Now, the third claim follows from the inductive hypothesis.

Let L and E be as described in the lemma. Let T_E be the parse tree of E, and v be the leaf node labeled with Δ_1. Let $v, u_1, \ldots, u_m$ be the path from v to the root of T_E. For each $i \in \{1, \ldots, m\}$, let E_i be the $\star$-expression whose parse tree is the subtree of T_E rooted at the child of u_i that is not an ancestor of v. Then, $E = \Delta_1 \star E_1 \star \cdots \star E_m$. Define $E'_1 = \sigma_{j_1} \star \cdots \star \sigma_{j_a}$, where $\langle \sigma_{j_1}, \ldots, \sigma_{j_a} \rangle$ equals $\mathrm{Op}(E_1)$ if σ_2 is not in $\mathrm{Op}(E_1)$, while equals $\mathrm{Op}(E_1) - \{\sigma_5\}$ otherwise. Define $E'_2 = \sigma_{k_1} \star \cdots \star \sigma_{k_b}$, $\ldots$, $E'_m = \sigma_{\ell_1} \star \cdots \star \sigma_{\ell_c}$ similarly. Since operator $\star$ is commutative and associative over Σ (by Lemma 6) and $\sigma_5 \star \sigma_2 = \sigma_2$, we have $E_i = E'_i$ for all $i \in \{1, \ldots, m\}$. Thus, $E = \Delta_1 \star E'_1 \star \cdots \star E'_m$. By Table 1, if two elements σ_g and σ_h of Σ satisfy $\sigma_g \star \sigma_h = \sigma_2$, then $\sigma_2 \in \{\sigma_g, \sigma_h\}$; similarly, if two elements σ_g and σ_h of Σ satisfy $\sigma_g \star \sigma_h = \sigma_5$, then $\sigma_g = \sigma_h = \sigma_5$. By this fact, for every $i \in \{1, \ldots, m\}$, if σ_2 (respectively, σ_5) is not in $\mathrm{Op}(E'_i)$, then σ_2 (respectively, σ_5) does not result when we evaluate E'_i from left to right. Thus, by the first claim, $E \subseteq \Delta_1 \star \sigma_{j_1} \star \cdots \star \sigma_{j_a} \star \sigma_{k_1} \star \cdots \star \sigma_{k_b} \star \cdots \star \sigma_{\ell_1} \star \cdots \star \sigma_{\ell_c}$. Now, by the third claim, Statement 1 holds.

Statement 2. First suppose that σ_2 is not in L. Then, by Statement 1, $U \subseteq \Delta_1 \star [L]_\star$. On the other hand, $\Delta_1 \star [L]_\star \subseteq U$ because $\Delta_1 \star [L]_\star$ is a $\star$-expression whose operand multiset is $\langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1 \rangle$. So, $U = \Delta_1 \star [L]_\star$.

Next, suppose that σ_2 is in L. Then, by Statement 1, $U \subseteq \Delta_1 \star [L - \{\sigma_5\}]_\star$. On the other hand, we can modify the $\star$-expression $\Delta_1 \star [L - \{\sigma_5\}]_\star$ by replacing one appearance of σ_2 with $(\sigma_2 \star \sigma_5 \star \cdots \star \sigma_5)$ in which σ_5 appears the same times as in L. The resulting $\star$-expression has the same value as $\Delta_1 \star [L - \{\sigma_5\}]_\star$ since $\sigma_2 \star \sigma_5 = \sigma_2$, and is a $\star$-expression whose operand multiset is $\langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1 \rangle$. So, $\Delta_1 \star [L - \{\sigma_5\}]_\star \subseteq U$. This completes the proof of Statement 2. □

Lemma 10 still holds after replacing each appearance of "Δ" in it with "Π", as stated in the following.

Lemma 11 *Suppose $\Pi_1 \subseteq \Pi$. Let $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Then, the following two statements hold:*

1. *For every $\star$-expression E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1 \rangle$, if σ_2 is not in L, then $E \subseteq \Pi_1 \star [L]_\star$; otherwise, $E \subseteq \Pi_1 \star [L - \{\sigma_5\}]_\star$.*

2. *Let U be the union of the values of all $\star$-expressions E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1 \rangle$. If σ_2 is not in L, then $U = \Pi_1 \star [L]_\star$; otherwise, $U = \Pi_1 \star [L - \{\sigma_5\}]_\star$.*

Proof: It suffices to modify the proof of Lemma 10 by replacing each appearance of "Δ" with "Π", and "Table 3" with "Table 4". □

Lemma 12 *Suppose that* $\Pi_1 \subseteq \Pi$ *and* $\Pi_2 \subseteq \Pi$*. Then, for every* $\sigma \in \{\sigma_1, \sigma_3, \sigma_4\}$*,* $\Pi_1 \star \Pi_2 \star \sigma = \Pi_2 \star \Pi_1 \star \sigma \subseteq \Pi_1 \star \sigma \star \Pi_2 = \Pi_1 \star (\sigma \star \Pi_2) = \Pi_1 \star (\Pi_2 \star \sigma)$*.*

Proof: One may verify the lemma using Tables 3, 4 and 5, under the assumption that $|\Pi_1| = |\Pi_2| = 1$. This then implies the lemma for every $\Pi_1 \subseteq \Pi$ and every $\Pi_2 \subseteq \Pi$. $\square$

Lemma 13 *Suppose that* $\Pi_1 \subseteq \Pi$ *and* $\Pi_2 \subseteq \Pi$*. Let* $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ *be a list of* σ*-types. Let* q_2 *and* q_5 *be the number of appearances of* σ_2 *and* σ_5 *in* L*, respectively. Let* $L' = L - \{\sigma_2, \sigma_5\}$*. Then, the following two statements hold:*

1. *For every* $\star$*-expression* E *with* $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1, \Pi_2 \rangle$*, if* $q_2 \geq 7$*, then* $E = \emptyset$*; otherwise,* $E \subseteq \cup_{(a,b,c,d,h,k)} (\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$*, where* (a,b,c,d,h,k) *ranges over all sextuples of nonnegative integers such that (1)* $a+c+h = q_2$*, (2)* $b \leq 1$*,* $d \leq 1$*,* $k \leq 1$*, and* $b+d+k \leq q_5$*, (3) if* $a \geq 1$ *then* $b = 0$*, (4) if* $c \geq 1$ *then* $d = 0$*, (5) if* $h \geq 1$ *then* $k = 0$*, and (6) if* $q_2 = 0$ *then* $b+d+k \geq \min\{1, q_5\}$*.*

2. *Let* U *be the union of the values of all* $\star$*-expressions* E *with* $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1, \Pi_2 \rangle$*. If* $q_2 \geq 7$*, then* $U = \emptyset$*; otherwise,* $U = \cup_{(a,b,c,d,h,k)} (\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$*, where* (a,b,c,d,h,k) *ranges over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1.*

Proof: We prove the two statements separately as follows.

Statement 1. Let T_E be the parse tree of E, v_1 be the leaf node of T_E labeled with Π_1, v_2 be the leaf node of T_E labeled with Π_2, and w be the lowest common ancestor of v_1 and v_2 in T_E. Let $v_1, u_1, \ldots, u_\ell, w$ be the path from v_1 to w in T_E. Similarly, let $v_2, u_{\ell+1}, \ldots, u_p, w$ be the path from v_2 to w in T_E. Let $w, u_{p+1}, \ldots, u_r$ be the path from w to the root of T_E. For each $i \in \{1, \ldots, r\}$, let E_i be the $\star$-expression whose parse tree is the subtree of T_E rooted at the child of u_i that is neither an ancestor of v_1 nor an ancestor of v_2. Then, $E = (\Pi_1 \star E_1 \star \cdots \star E_\ell) \star (\Pi_2 \star E_{\ell+1} \star \cdots \star E_p) \star E_{p+1} \star \cdots \star E_r$. Suppose that $\cup_{1 \leq x \leq \ell} \mathrm{Op}(E_x) = \langle \sigma_{j_1}, \ldots, \sigma_{j_f} \rangle$, $\cup_{\ell+1 \leq x \leq p} \mathrm{Op}(E_x) = \langle \sigma_{j_{f+1}}, \ldots, \sigma_{j_g} \rangle$, and $\cup_{p+1 \leq x \leq r} \mathrm{Op}(E_x) = \langle \sigma_{j_{g+1}}, \ldots, \sigma_{j_q} \rangle$. Let L_1 be the list $(\sigma_{j_1}, \ldots, \sigma_{j_f})$, L_2 be the list $(\sigma_{j_{f+1}}, \ldots, \sigma_{j_g})$, and L_3 be the list $(\sigma_{j_{g+1}}, \ldots, \sigma_{j_q})$. If σ_2 is not in L_1, then let $L'_1 = L_1$; otherwise, let $L'_1 = L_1 - \{\sigma_5\}$. Similarly, L'_2 and L'_3 are defined from L_2 and L_3, respectively. Then, by Statement 1 in Lemma 10 and Statement 1 in Lemma 11, $E \subseteq (\Pi_1 \star [L'_1]_\star) \star (\Pi_2 \star [L'_2]_\star) \star [L'_3]_\star = (\Pi_1 \star [L'_1]_\star) \star (\Pi_2 \star [L'_2]_\star) \star [L'_3 - \{\sigma_2, \sigma_5\}]_\star \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$. In turn, by Lemma 12, $E \subseteq (\Pi_1 \star [L'_1]_\star \star [L'_3 - \{\sigma_2, \sigma_5\}]_\star) \star (\Pi_2 \star [L'_2]_\star) \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$. So, by Statement 1 in Lemma 11, $E \subseteq (\Pi_1 \star [L'_1]_\star \star [L'_3 - \{\sigma_2, \sigma_5\}]_\star) \star (\Pi_2 \star [L'_2 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star \star [L'_2 - \{\sigma_2, \sigma_5\}]_\star) \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$. In turn, by Lemma 12, $E \subseteq (\Pi_1 \star [L'_1]_\star \star [L'_3 - \{\sigma_2, \sigma_5\}]_\star \star [L'_2 - \{\sigma_2, \sigma_5\}]_\star) \star (\Pi_2 \star [L'_2 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star) \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$. Moreover, by Statement 1 in Lemma 11, $E \subseteq (\Pi_1 \star [L'_1 - \{\sigma_2, \sigma_5\}]_\star \star [L'_3 - \{\sigma_2, \sigma_5\}]_\star \star [L'_2 - \{\sigma_2, \sigma_5\}]_\star \star [L'_1 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star) \star (\Pi_2 \star [L'_2 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star) \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$. Note that $\mathrm{Op}([L']_\star) = \mathrm{Op}([L'_1 - \{\sigma_2, \sigma_5\}]_\star) \cup \mathrm{Op}([L'_3 - \{\sigma_2, \sigma_5\}]_\star) \cup \mathrm{Op}([L'_2 - \{\sigma_2, \sigma_5\}]_\star)$. Thus, by Statement 1 in Lemma 11, $E \subseteq (\Pi_1 \star [L']_\star \star [L'_1 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star) \star (\Pi_2 \star [L'_2 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star) \star [L'_3 - \{\sigma_1, \sigma_3, \sigma_4\}]_\star$.

Let a (respectively, c and h) be the number of appearances of σ_2 in L'_1 (respectively, in L'_2 and L'_3). Let b (respectively, d and k) be the smaller of 1 and the number of appearances of σ_5 in L'_1 (respectively, L'_2 and L'_3). By the definition of L'_1, L'_2, L'_3, the sextuple (a,b,c,d,h,k) satisfies the conditions (1) through (6) in the lemma.

Note that for every $\pi \in \Pi$ and every $\delta \in \Delta$, $\pi \star \sigma_2 \star \sigma_2 \star \sigma_2 = \emptyset$, $\delta \star \sigma_2 \star \sigma_2 \star \sigma_2 = \emptyset$. If $q_2 \geq 7$, then at least one of a, c, h is ≥ 3 and hence $E = \emptyset$. Otherwise, since $\pi \star \sigma_5 \star \sigma_5 = \pi \star \sigma_5$ and $\delta \star \sigma_5 \star \sigma_5 =$

$\delta\star\sigma_5$ for every $\pi \in \Pi$ and every $\delta \in \Delta$, $E \subseteq (\Pi_1\star[L']_\star\star[\sigma_2^a]_\star\star[\sigma_5^b]_\star)\star(\Pi_2\star[\sigma_2^c]_\star\star[\sigma_5^d]_\star)\star[\sigma_2^h]_\star\star[\sigma_5^k]_\star$. This completes the proof of Statement 1.

Statement 2. If $q_2 \geq 7$, then $U = \emptyset$ by Statement 1. Next, suppose that $q_2 \leq 6$. Then, by Statement 1, $U \subseteq \cup_{(a,b,c,d,h,k)}(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$, where (a,b,c,d,h,k) ranges over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1. On the other hand, if $q_2 = q_5 = 0$, then for each sextuple (a,b,c,d,h,k) satisfying the conditions (1) through (6) in Statement 1, $(\Pi_1\star[L']_\star\star[\sigma_2^a]_\star\star[\sigma_5^b]_\star)\star(\Pi_2\star[\sigma_2^c]_\star\star[\sigma_5^d]_\star)\star[\sigma_2^h]_\star\star[\sigma_5^k]_\star$ is a $\star$-expression whose operand multiset is $\langle\sigma_{i_1},\ldots,\sigma_{i_q},\Pi_1,\Pi_2\rangle$; and so $(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star \subseteq U$. If $q_2 \geq 1$ or $q_5 \geq 1$, then for each sextuple (a,b,c,d,h,k) satisfying the conditions (1) through (6) in Statement 1, we can modify the $\star$-expression $(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$ either (i) by replacing one appearance of σ_2 with $(\sigma_2\star\sigma_5\star\cdots\star\sigma_5)$ in which σ_5 appears $q_5-(b+d+k)$ times, or (ii) by replacing one appearance of σ_5 with $(\sigma_5\star\cdots\star\sigma_5)$ in which σ_5 appears $q_5-(b+d+k)+1$ times. The resulting $\star$-expression has the same value as $(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$ since $\sigma_2 \star \sigma_5 = \sigma_2$ and $\sigma_5 \star \sigma_5 = \sigma_5$; and is a $\star$-expression whose operand multiset is $\langle\sigma_{i_1},\ldots,\sigma_{i_q},\Pi_1,\Pi_2\rangle$. So, $(\Pi_1\star[L']_\star\star[\sigma_2^a]_\star\star[\sigma_5^b]_\star)\star(\Pi_2\star[\sigma_2^c]_\star\star[\sigma_5^d]_\star)\star[\sigma_2^h]_\star\star[\sigma_5^k]_\star \subseteq U$. Consequently, $\cup_{(a,b,c,d,h,k)}(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star \subseteq U$, where (a,b,c,d,h,k) ranges over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1. This completes the proof of Statement 2. $\square$

5.3 Parallel composition of DTCs

Let $\Gamma_1 = \Sigma_1 \cup \Pi_1 \cup \Delta_1$ and $\Gamma_2 = \Sigma_2 \cup \Pi_2 \cup \Delta_2$ be two subsets of Γ, where for $i = 1,2$, $\Sigma_i \subseteq \Sigma$, $\Pi_i \subseteq \Pi$, and $\Delta_i \subseteq \Delta$. Define:

$$\begin{aligned}\Gamma_1 \diamond \Gamma_2 \quad = \quad & (\cup_{\sigma_i\in\Sigma_1,\sigma_j\in\Sigma_2}\mathcal{K}_\sigma(R(\sigma_i)\diamond R(\sigma_j))) \cup (\cup_{\sigma_i\in\Sigma_1,\pi_j\in\Pi_2}\mathcal{K}_\pi(R(\sigma_i)\diamond R(\pi_j))) \cup \\ & (\cup_{\pi_i\in\Pi_1,\sigma_j\in\Sigma_2}\mathcal{K}_\pi(R(\pi_i)\diamond R(\sigma_j))) \cup (\cup_{\sigma_i\in\Sigma_1,\delta_j\in\Delta_2}\mathcal{K}_\delta(R(\sigma_i)\diamond R(\delta_j))) \cup \\ & (\cup_{\delta_i\in\Delta_1,\sigma_j\in\Sigma_2}\mathcal{K}_\delta(R(\delta_i)\diamond R(\sigma_j))) \cup (\cup_{\pi_i\in\Pi_1,\pi_j\in\Pi_2}\mathcal{K}_\delta(R(\pi_i)\diamond R(\pi_j)))\end{aligned}$$

If $\Gamma_1 = \emptyset$ or $\Gamma_2 = \emptyset$, then $\Gamma_1 \diamond \Gamma_2 = \emptyset$.

As in the paragraph right before Lemma 9, we can argue that the operator $\diamond$ is commutative by its definition. For later reference, we state this in the following lemma.

Lemma 14 If Γ_1 and Γ_2 are two subsets of Γ, then $\Gamma_1 \diamond \Gamma_2 = \Gamma_2 \diamond \Gamma_1$.

For each $\sigma_i \in \Sigma$ and each $\delta_j \in \Delta$, the (i,j)-entry of Table 6 contains all $\delta_k \in \{\sigma_i\} \diamond \{\delta_j\}$. For each $\pi_i \in \Pi$ and each $\sigma_j \in \Sigma$, the (i,j)-entry of Table 7 contains all $\pi_k \in \{\pi_i\} \diamond \{\sigma_j\}$. For each $\pi_i \in \Pi$ and each $\pi_j \in \Pi$, the (i,j)-entry of Table 8 contains all $\delta_k \in \{\pi_i\} \diamond \{\pi_j\}$. These tables can be verified by inspecting the graphs in Figures 5, 7, and 8. As examples, we verify one entry from each of the Tables 6, 7 and 8. They are shown in Figure 10. In each case, two representative graphs are composed in parallel by identifying their common poles. The type of the resulting graph is shown in the figure.

By modifying the four paragraphs right before Lemma 10 by replacing each letter "$\star$" with "$\diamond$", we can define a $\diamond$-*expression* E, its *value*, etc. Moreover, we can prove the following four lemmas, just via modifying the proofs of Lemmas 10 through 13 by replacing each appearance of "σ_2", "σ_5", "$\star$", "Table 3", "Table 4", and "Table 5" in them with "σ_3", "σ_4", "$\diamond$", "Table 6", "Table 7", and "Table 8", respectively. We state these corresponding lemmas as follows.

	δ_1	δ_2	δ_3	δ_4	δ_5	δ_6
σ_1	δ_3		δ_1			
σ_2	δ_1		δ_3			
σ_3		$\delta_0,\, \delta_5$		δ_2		δ_2
σ_4		δ_2		δ_4	δ_5	δ_6
σ_5				δ_5		δ_5

Table 6: $\{\sigma_i\} \diamond \{\delta_j\}$.

	σ_1	σ_2	σ_3	σ_4	σ_5
π_1	π_1	π_1	π_3		
π_2	π_4	π_2	π_1	π_2	π_3
π_3	π_3	π_3		π_3	
π_4	π_2	π_4	π_1	π_4	π_3

Table 7: $\{\pi_i\} \diamond \{\sigma_j\}$.

	π_1	π_2	π_3	π_4
π_1	$\delta_0,\, \delta_1,\, \delta_3$	δ_2		δ_2
π_2	δ_2	$\delta_3,\, \delta_4$	$\delta_1,\, \delta_3,\, \delta_5$	$\delta_1,\, \delta_6$
π_3		$\delta_1,\, \delta_3,\, \delta_5$		$\delta_1,\, \delta_3,\, \delta_5$
π_4	δ_2	$\delta_1,\, \delta_6$	$\delta_1,\, \delta_3,\, \delta_5$	$\delta_3,\, \delta_4$

Table 8: $\{\pi_i\} \diamond \{\pi_j\}$.

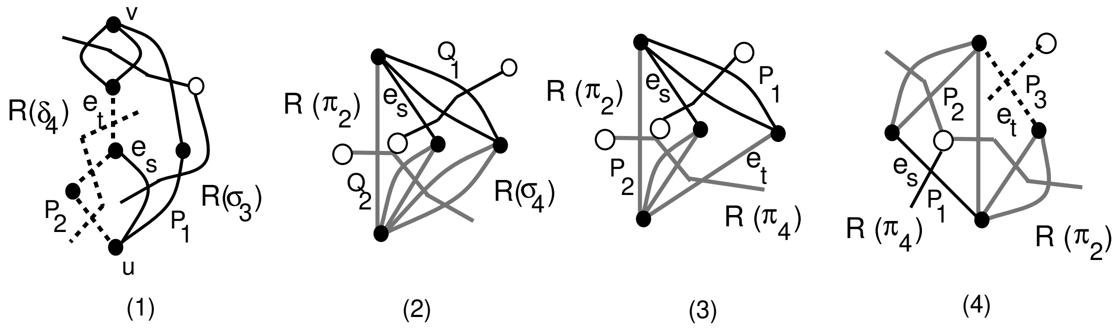

Figure 10: (1) $\sigma_3 \diamond \delta_4 = \delta_2$, (2) $\pi_2 \diamond \sigma_4 = \pi_2$, (3) $\pi_2 \diamond \pi_4 = \delta_1$, (4) $\pi_2 \diamond \pi_4 = \delta_6$.

Lemma 15 *Suppose $\Delta_1 \subseteq \Delta$. Let $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Then, the following two statements hold:*

1. *For every $\diamond$-expression E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1 \rangle$, if σ_3 is not in L, then $E \subseteq \Delta_1 \diamond [L]_\diamond$; otherwise, $E \subseteq \Delta_1 \diamond [L - \{\sigma_4\}]_\diamond$.*
2. *Let U be the union of the values of all $\diamond$-expressions E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Delta_1 \rangle$. If σ_3 is not in L, then $U = \Delta_1 \diamond [L]_\diamond$; otherwise, $U = \Delta_1 \diamond [L - \{\sigma_4\}]_\diamond$.*

Lemma 16 *Suppose $\Pi_1 \subseteq \Pi$. Let $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Then, the following two statements hold:*

1. *For every $\diamond$-expression E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1 \rangle$, if σ_3 is not in L, then $E \subseteq \Pi_1 \diamond [L]_\diamond$; otherwise, $E \subseteq \Pi_1 \diamond [L - \{\sigma_4\}]_\diamond$.*
2. *Let U be the union of the values of all $\diamond$-expressions E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1 \rangle$. If σ_3 is not in L, then $U = \Pi_1 \diamond [L]_\diamond$; otherwise, $U = \Pi_1 \diamond [L - \{\sigma_4\}]_\diamond$.*

Lemma 17 *Suppose that $\Pi_1 \subseteq \Pi$ and $\Pi_2 \subseteq \Pi$. Then, for every $\sigma \in \{\sigma_1, \sigma_2, \sigma_5\}$, $\Pi_1 \diamond \Pi_2 \diamond \sigma = \Pi_2 \diamond \Pi_1 \diamond \sigma \subseteq \Pi_1 \diamond \sigma \diamond \Pi_2 = \Pi_1 \diamond (\sigma \diamond \Pi_2) = \Pi_1 \diamond (\Pi_2 \diamond \sigma)$.*

Lemma 18 *Suppose that $\Pi_1 \subseteq \Pi$ and $\Pi_2 \subseteq \Pi$. Let $L = (\sigma_{i_1}, \ldots, \sigma_{i_q})$ be a list of σ-types. Let q_3 and q_4 be the number of appearances of σ_3 and σ_4 in L, respectively. Let $L' = L - \{\sigma_3, \sigma_4\}$. Then, the following two statements hold:*

1. *For every $\diamond$-expression E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1, \Pi_2 \rangle$, if $q_3 \geq 7$, then $E = \emptyset$; otherwise, $E \subseteq \cup_{(a,b,c,d,h,k)} (\Pi_1 \diamond [L']_\diamond \diamond [\sigma_3^a]_\diamond \diamond [\sigma_4^b]_\diamond) \diamond (\Pi_2 \diamond [\sigma_3^c]_\diamond \diamond [\sigma_4^d]_\diamond) \diamond [\sigma_3^h]_\diamond \diamond [\sigma_4^k]_\diamond$, where (a, b, c, d, h, k) ranges over all sextuples of nonnegative integers such that (1) $a + c + h = q_3$, (2) $b \leq 1$, $d \leq 1$, $k \leq 1$, and $b + d + k \leq q_4$, (3) if $a \geq 1$ then $b = 0$, (4) if $c \geq 1$ then $d = 0$, (5) if $h \geq 1$ then $k = 0$, and (6) if $q_3 = 0$ then $b + d + k \geq \min\{1, q_4\}$.*
2. *Let U be the union of the values of all $\diamond$-expressions E with $\mathrm{Op}(E) = \langle \sigma_{i_1}, \ldots, \sigma_{i_q}, \Pi_1, \Pi_2 \rangle$. If $q_3 \geq 7$, then $U = \emptyset$; otherwise, $U = \cup_{(a,b,c,d,h,k)} (\Pi_1 \diamond [L']_\diamond \diamond [\sigma_3^a]_\diamond \diamond [\sigma_4^b]_\diamond) \diamond (\Pi_2 \diamond [\sigma_3^c]_\diamond \diamond [\sigma_4^d]_\diamond) \diamond [\sigma_3^h]_\diamond \diamond [\sigma_4^k]_\diamond$, where (a, b, c, d, h, k) ranges over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1.*

5.4 An $O(|G|^3)$-time algorithm

Let e_s and e_t be a pair of distinct edges of G. We want to decide whether G has a valid embedding that has a DET starting with e_s and ending with e_t. To do this, we process the nodes of T in post-order. For each node μ of T such that neither e_s nor e_t is in $\mathrm{Pt}(\mu)$, we compute $\mathcal{K}_\sigma(\mathrm{Pt}(\mu))$ as in Section 4.3; we may assume that $|\mathcal{K}_\sigma(\mathrm{Pt}(\mu))| = 1$ since otherwise no valid embedding of G has a DET starting with e_s and ending with e_t. For each Q-node μ such that $\mathrm{Pt}(\mu)$ consists of e_s only or e_t only, we set $\mathcal{K}_\pi(\mathrm{Pt}(\mu)) = \{\pi_1\}$.

Consider the processing of a non-Q-node μ such that e_s or e_t is in $\mathrm{Pt}(\mu)$. Let $\nu_1, \ldots, \nu_r$ be the children of μ in T. Let $B = \mathrm{Pt}(\mu)$. For each $i \in \{1, \ldots, r\}$, let $B_i = \mathrm{Pt}(\nu_i)$, and e_i be the virtual edge of ν_i in $\mathrm{Sk}(\mu)$.

- *Case 1:* B contains both e_s and e_t but no child of B does. Without loss of generality, we may assume that B_1 contains e_s and B_2 contains e_t. Then, none of $B_3, \ldots, B_r$ contains e_s or e_t. Let $\Pi_1 = \mathcal{K}_\pi(B_1)$ and $\Pi_2 = \mathcal{K}_\pi(B_2)$. For each $j \in \{3, \ldots, r\}$, let $\mathcal{K}_\sigma(B_j) = \{\sigma_{\ell_j}\}$. Let L be the list $(\sigma_{\ell_3}, \ldots, \sigma_{\ell_r})$.
  - *Case 1.1:* μ is an R-node. Then, $\mathrm{Sk}(\mu)$ has no different embedding other than itself. Based on Statements 1 and 2 in Lemma 4, we compute $\mathcal{K}_\delta(B)$ as follows. We initialize $\mathcal{K}_\delta(B) = \emptyset$. After this, for each quadruple $D = (\pi_{i_1}, \pi_{i_2}, G_1, G_2)$ such that $\pi_{i_1} \in \mathcal{K}_\pi(B_1)$, $\pi_{i_2} \in \mathcal{K}_\pi(B_2)$, and G_1 (respectively, G_2) is obtained from $R(\pi_{i_1})$ (respectively, $R(\pi_{i_2})$) by horizontally and/or vertically flipping it, we first construct a TTG H_D from $\mathrm{Sk}(\mu)$ by replacing e_1 with G_1, e_2 with G_2, and e_j with $R(\sigma_{\ell_j})$ for all $j \in \{3, \ldots, r\}$. We then obtain the maximal double trail P_1 in H_D such that (1) e_s and its dual edge e_s^* are respectively the starting edge of P_1 and its dual trail P_1^* and (2) the direction of e_s in P_1 and that of e_s^* in P_1^* are the same as oriented by the unique valid DTC of G_1. Similarly, we obtain the maximal double trail P_2 in H_D such that (1) e_t and its dual edge e_t^* are respectively the starting edge of P_2 and its dual trail P_2^* and (2) the direction of e_t in P_2 and that of e_t^* in P_2^* are the same as oriented by the unique valid DTC of G_2. Let P_2' be the reverse of P_2. Let M be the set of all edges of H_D traversed by neither P_1 nor P_2. Suppose $M = \emptyset$. If P_1 and P_2' are distinct, we check if P_1 and P_2' form a DTC of H_D of a δ-type. If P_1 and P_2' are the same, we check if P_1 forms a DTC of H_D of a δ-type. If it is, we add this δ-type to $\mathcal{K}_\delta(B)$.

    Suppose that $M \neq \emptyset$ and P_1 and P_2 are edge-disjoint. We try to find an access edge $e \in M$ such that e is incident to a pole u of H_D and the dual edge e^* of e is incident to a pole x of H_D^*. If such e exists, then we obtain the maximal double trail P_3 in H_D such that (1) e and e^* are respectively the starting edge of P_3 and its dual trail P_3^* and (2) u and x are respectively the starting vertex of P_3 and P_3^*. Note that the other endpoint of e (respectively, e^*) is not a pole of H_D (respectively, H_D^*). If $\{P_1, P_2', P_3\}$ is a DTC of H_D of a δ-type, we add this δ-type to $\mathcal{K}_\delta(B)$. Similarly, if $\{P_1, P_2', P_3'\}$ is a DTC of H_D of a δ-type, we add this δ-type to $\mathcal{K}_\delta(B)$, where P_3' is the reverse of P_3.
  - *Case 1.2:* μ is an S-node. Let q_2 and q_5 be the number of appearances of σ_2 and σ_5 in L, respectively. Let $L' = L - \{\sigma_2, \sigma_5\}$. By Statement 2 in Lemma 13 and Statements 1 and 2 in Lemma 4, if $q_2 \geq 7$, then $\mathcal{K}_\delta(B) = \emptyset$; otherwise, $\mathcal{K}_\delta(B) = \cup_{(a,b,c,d,h,k)}(\Pi_1 \star [L']_\star \star [\sigma_2^a]_\star \star [\sigma_5^b]_\star) \star (\Pi_2 \star [\sigma_2^c]_\star \star [\sigma_5^d]_\star) \star [\sigma_2^h]_\star \star [\sigma_5^k]_\star$, where (a, b, c, d, h, k) ranges

over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1 in Lemma 13,

  - *Case 1.3:* μ is a P-node. Let q_3 and q_4 be the number of appearances of σ_3 and σ_4 in L, respectively. Let $L' = L - \{\sigma_3, \sigma_4\}$. By Statement 2 in Lemma 18 and Statements 1 and 2 in Lemma 4, if $q_3 \geq 7$, then $\mathcal{K}_\delta(B) = \emptyset$; otherwise, $\mathcal{K}_\delta(B) = \cup_{(a,b,c,d,h,k)}(\Pi_1 \diamond [L']_\diamond \diamond [\sigma_3^a]_\diamond \diamond [\sigma_4^b]_\diamond) \diamond (\Pi_2 \diamond [\sigma_3^c]_\diamond \diamond [\sigma_4^d]_\diamond) \diamond [\sigma_3^h]_\diamond \diamond [\sigma_4^k]_\diamond$, where (a, b, c, d, h, k) ranges over all sextuples of nonnegative integers that satisfy the conditions (1) through (6) in Statement 1 in Lemma 18.

- *Case 2:* B contains both e_s and e_t and so does one child block of B. Without loss of generality, we can assume that B_1 contains both e_s and e_t. Then, none of $B_2, \ldots, B_r$ contains e_s or e_t. Let $\Delta_1 = \mathcal{K}_\delta(B_1) - \{\delta_0\}$. For each $j \in \{2, \ldots, r\}$, let $\mathcal{K}_\sigma(B_j) = \{\sigma_{\ell_j}\}$. Let L be the list $(\sigma_{\ell_2}, \ldots, \sigma_{\ell_r})$.

  - *Case 2.1:* μ is an R-node. Then, based on Statements 1 and 3 in Lemma 4, we compute $\mathcal{K}_\delta(B)$ as follows. We initialize $\mathcal{K}_\delta(B) = \emptyset$. After this, for each pair $D = (\delta_i, G_1)$ such that $\delta_i \in \Delta_1$ and G_1 is obtained from $R(\delta_i)$ by horizontally and/or vertically flipping it, we first construct a TTG H_D from $\mathrm{Sk}(\mu)$ by replacing e_1 with G_1, and e_j with $R(\sigma_{\ell_j})$ for all $j \in \{2, \ldots, r\}$. Similarly to Case 1.1, we then check if H_D has a δ-type DTC; if it has, we add the δ-type of the DTC into $\mathcal{K}_\delta(B)$.
  - *Case 2.2:* μ is an S-node. Then, by Statement 2 in Lemma 10 and Statements 1 and 3 in Lemma 4, if σ_2 is not in L, then $\mathcal{K}_\delta(B) = \Delta_1 \star [L]_\star$; otherwise, $\mathcal{K}_\delta(B) = \Delta_1 \star [L - \{\sigma_5\}]_\star$.
  - *Case 2.3:* μ is a P-node. Then, by Statement 2 in Lemma 15 and Statements 1 and 3 in Lemma 4, if σ_3 is not in L, then $\mathcal{K}_\delta(B) = \Delta_1 \diamond [L]_\diamond$; otherwise, $\mathcal{K}_\delta(B) = \Delta_1 \diamond [L - \{\sigma_4\}]_\diamond$.

- *Case 3:* B contains one of e_s and e_t. We assume that B contains e_s; the other case is similar. Without loss of generality, we can assume that B_1 contains e_s. Then, none of $B_2, \ldots, B_r$ contains e_s or e_t. Let $\Pi_1 = \mathcal{K}_\pi(B_1)$. For each $j \in \{2, \ldots, r\}$, let $\mathcal{K}_\sigma(B_j) = \{\sigma_{\ell_j}\}$. Let L be the list $(\sigma_{\ell_2}, \ldots, \sigma_{\ell_r})$.

  - *Case 3.1:* μ is an R-node. Then, based on Statements 1 and 2 in Lemma 4, we compute $\mathcal{K}_\pi(B)$ as follows. We initialize $\mathcal{K}_\pi(B) = \emptyset$. After this, for each pair $D = (\pi_i, G_1)$ such that $\pi_i \in \Pi_1$ and G_1 is obtained from $R(\pi_i)$ by horizontally and/or vertically flipping it, we first construct a TTG H_D from $\mathrm{Sk}(\mu)$ by replacing e_1 with G_1, and e_j with $R(\sigma_{\ell_j})$ for all $j \in \{2, \ldots, r\}$. Similarly to Case 1.1, we then check if H_D has a π-type DTC; if it has, we add the π-type of the DTC into $\mathcal{K}_\pi(B)$.
  - *Case 3.2:* μ is an S-node. Then, by Statement 2 in Lemma 11 and Statements 1 and 2 in Lemma 4, if σ_2 is not in L, then $\mathcal{K}_\pi(B) = \Pi_1 \star [L]_\star$; otherwise, $\mathcal{K}_\pi(B) = \Pi_1 \star [L - \{\sigma_5\}]_\star$.
  - *Case 3.3:* μ is a P-node. Then, by Statement 2 in Lemma 16 and Statements 1 and 2 in Lemma 4, if σ_3 is not in L, then $\mathcal{K}_\pi(B) = \Pi_1 \diamond [L]_\diamond$; otherwise, $\mathcal{K}_\pi(B) = \Pi_1 \diamond [L - \{\sigma_4\}]_\diamond$.

By the above discussions, it takes $O(|\mathrm{Sk}(\mu)|)$ time to process μ. Thus, by Lemma 1, the time needed to compute $\mathcal{K}_\delta(G)$ is $O(|G|)$. Note that G has a DET starting with e_s and ending with e_t if and only if $\delta_0 \in \mathcal{K}_\delta(G)$. Thus, by trying all pairs (e_s, e_t) of edges of G, we can decide in $O(|G|^3)$ time whether G has a DET.

Theorem 2 *Given a TTG G, it takes $O(|G|^3)$ time to decide if G has a valid embedding.*

6 An $O(|G|)$-time algorithm

This section refines the algorithm in Section 5.4. The basic idea is that we do not specify the starting and the ending edges in advance. Rather, we try to discover them when processing T in post-order. Throughout this section, we assume that G^* is not Eulerian; the case where G is not Eulerian is similar. Let T^* be the dual decomposition-tree of T.

First, we need some definitions and lemmas. Recall that the degree parity of a vertex in G is 0 if its degree is even, and is 1 otherwise. The *pole-parity multiset* M_G of G consists of the degree parities of the poles of G. The *pole-status* of G is the pair (M_G, M_{G^*}) where M_G and M_{G^*} are the pole-parity multiset of G and G^*, respectively.

G is *E-good* if it is Eulerian. G is *O-good* if its odd-degree vertices are just its poles. G is *good* if it is either E- or O-good. G is *bad* if it has at least one odd-degree nonpole.

A node μ of T is *E-good* (respectively, *O-good*, *good*, or *bad*) if all different embeddings of $\mathrm{Pt}(\mu)$ are E-good (respectively, O-good, good, or bad).

Lemma 19 *Let μ be a node of T. Let $B = \mathrm{Pt}(\mu)$. Then, exactly one of the following (1) through (3) holds: (1) μ is E-good; (2) μ is O-good; (3) μ is bad. Moreover, for all descendants ν (including μ itself) of μ in T, it takes $O(|B|)$ total time to (a) decide whether ν is E-good, O-good, or bad, and (b) compute the pole-parity multiset of $\mathrm{Pt}(\nu)$ when ν is good.*

Proof: By induction. In case μ is a leaf node of T, B is a single edge; hence, μ is O-good.

Next, suppose that μ is a non-leaf node. Let $\nu_1, \ldots, \nu_q$ be the children of μ in T. For each $i \in \{1, \ldots, q\}$, let $B_i = \mathrm{Pt}(\nu_i)$ and e_i be the virtual edge of ν_i in $\mathrm{Sk}(\mu)$. If some child of μ is bad, then μ must be bad. So, assume that no child of μ is bad. By the inductive hypothesis, each child of μ is either E- or O-good. Let $\mathcal{B}$ be an (arbitrary) different embedding of B. For each $i \in \{1, \ldots, q\}$, let $\mathcal{B}_i$ be the child block of $\mathcal{B}$ that is a different embedding of B_i. For the purpose of deciding whether $\mathcal{B}$ is E-good, O-good, or bad, we may assume that $\mathcal{B}_i = B_i$ for all $i \in \{1, \ldots, q\}$, because $\mathcal{B}_i$ is E-good (respectively, O-good) if and only if so is B_i.

Case 1: μ is a P-node. Then, $\mathcal{B}$ can be obtained from B by permuting the children of μ. Permuting the children of μ does not change the degree parity of any vertex in B. Thus, if an odd (respectively, even) number of children of μ is O-good, μ is O-good (respectively, E-good).

Case 2: μ is an S-node. Then, $\mathcal{B}$ can be obtained from B by permuting the children of μ. When every child of μ is E-good, permuting the children of μ does not change the degree parity of any vertex in B, and hence $\mathcal{B}$ is E-good and so is μ. Similarly, when every child of μ is O-good, μ is O-good. So, suppose that μ has both E-good children and O-good children. Then, some E-good $\mathcal{B}_i$ must be next to an O-good $\mathcal{B}_j$ in $\mathcal{B}$. The common pole of $\mathcal{B}_i$ and $\mathcal{B}_j$ in $\mathcal{B}$ has odd degree. Thus, $\mathcal{B}$ is bad and so is μ.

Case 3: μ is an R-node. Then, $\mathrm{Sk}(\mu)$ has no different embedding other than itself. $\mathcal{B}$ is obtained from $\mathrm{Sk}(\mu)$ by replacing e_i with $\mathcal{B}_i$ for all $i \in \{1, \ldots, q\}$. Note that the edges of $\mathrm{Sk}(\mu)$ are just $e_1, \ldots, e_q$. For all $i \in \{1, \ldots, q\}$, we mark e_i if ν_i is O-good. If some nonpole in $\mathrm{Sk}(\mu)$ is incident to an odd number of marked edges, then μ is bad. Otherwise, if a pole of $\mathrm{Sk}(\mu)$ is incident to an odd number of marked edges, then μ is O-good; otherwise, μ is E-good.

Next we describe the implementation of this procedure. Let T_μ be the subtree of T rooted at the node μ. We perform a post-order computation on T_μ. Each leaf node ν is an O-good node, and both poles of $\mathrm{Pt}(\nu)$ have odd parity. For each internal node ν in T_μ, whether ν is

E-good, O-good or bad and the pole-parity multiset of $\mathrm{Pt}(\nu)$ can be determined as in the three cases above, in at most $O(|\mathrm{Sk}(\nu))|$ time. Thus, the total time needed is $O(|B|)$ by Lemma 1. □

By Lemma 2, each node of T is also a node of the dual decomposition-tree T^* of T. A node μ of T^* is *E-good* (respectively, *O-good*, *good*, or *bad*) if all different embeddings of the dual graph of $\mathrm{Pt}(\mu)$ are E-good (respectively, O-good, good, or bad). Lemma 19 still holds after changing T to T^*, $\mathrm{Pt}(\mu)$ to its dual graph, and $\mathrm{Pt}(\nu)$ to its dual graph.

Let μ be a node of T and $B = \mathrm{Pt}(\mu)$. μ is *doubly good* if for every different embedding $\mathcal{B}$ of B, neither $\mathcal{B}$ nor $\mathcal{B}^*$ has an odd-degree nonpole. μ is *doubly bad* if for every different embedding $\mathcal{B}$ of B, $\mathcal{B}$ or $\mathcal{B}^*$ has at least one odd-degree nonpole. By Lemma 19, μ is either doubly good or doubly bad. μ is *critical* if (1) all children of μ are doubly good and (2) μ is doubly bad or is the root of T.

Lemma 20 *If G has a valid embedding, then either one or two nodes of T are critical.*

Proof: Note that each leaf node of T is doubly good. Thus, at least one node of T is critical. If the root of T is critical, then all nodes of T except the root must be doubly good and the root is the unique critical node. Otherwise, let $\mu_1, \ldots, \mu_k$ be the critical nodes in T and $B_i = \mathrm{Pt}(\mu_i)$. By the definition of critical nodes, none of $\mu_1, \ldots, \mu_k$ can be an ancestor of another. Thus, no two of $B_1, \ldots, B_k$ share an edge. On the other hand, for each $i \in \{1, \ldots, k\}$, either B_i or B_i^* has an odd-degree nonpole; any DET of G must start or end with an edge in B_i. Therefore, if $k \geq 3$, G cannot have a DET. □

Lemma 21 *Let μ be a doubly-bad node of T, and $B = \mathrm{Pt}(\mu)$. Then, all different embeddings $\mathcal{B}$ of B such that $\mathcal{B}$ has a π-type DTC have the same pole-status p, which can be computed as follows:*

1. *If μ is an E-good node of T, then $p = (\langle 0, 0\rangle, \langle 0, 1\rangle)$.*
2. *If μ is an O-good node of T, then $p = (\langle 1, 1\rangle, \langle 0, 1\rangle)$.*
3. *If μ is an E-good node of T^*, then $p = (\langle 0, 1\rangle, \langle 0, 0\rangle)$.*
4. *If μ is an O-good node of T^*, then $p = (\langle 0, 1\rangle, \langle 1, 1\rangle)$.*
5. *If μ is a bad node of both T and T^*, then $p = (\langle 0, 1\rangle, \langle 0, 1\rangle)$.*

Proof: Suppose that a different embedding $\mathcal{B}$ of B has a π-type DTC. Then, both $\mathcal{B}$ and $\mathcal{B}^*$ have at most one odd-degree nonpole (see Figure 7). Indeed, at least one of $\mathcal{B}$ and $\mathcal{B}^*$ has exactly one odd-degree nonpole, because μ is doubly bad. If μ is a bad node of T, then the pole-parity multiset of $\mathcal{B}$ is $\langle 0, 1\rangle$. On the other hand, by Lemma 19, if μ is an E-good (respectively, O-good) node of T, then the pole-parity multiset of $\mathcal{B}$ is $\langle 0, 0\rangle$ (respectively, $\langle 1, 1\rangle$). Similarly, by checking whether μ is a good node of T^*, we can decide the pole-parity multiset of $\mathcal{B}^*$. □

Let $B = \mathrm{Pt}(\mu)$ where μ is a non-root node of T. A π- or δ-type γ of B is *useful* if some valid embedding $\mathcal{G}$ of G has a DET P such that $P \cap \mathcal{B}$ is of Type γ, where $\mathcal{B}$ is the block of $\mathcal{G}$ that is a different embedding of B.

Even for a doubly-good node μ, the block $B = \mathrm{Pt}(\mu)$ might contain the starting edge e_s and/or the ending edge e_t of a DET of a valid embedding of G (when e_s and/or e_t is incident to a pole of B). Thus, a σ-type of B might be actually a π- or δ-type. In the following, we investigate which σ-type can be mapped to which π- or δ-type. The mapping from σ-types to π-types is specified by the following function $f : \{1, \ldots, 5\} \to \{1, \ldots, 4\}$ where $f(1) = f(2) = f(3) = 1$, $f(4) = 4$, and $f(5) = 3$.

Lemma 22 *Let μ be a doubly-good non-root node of T. Let $B = \mathrm{Pt}(\mu)$. Suppose that π_j is a useful π-type of B. Then, $\mathcal{K}_\sigma(B) = \{\sigma_i\}$ for some $\sigma_i \in \Sigma$, and $\pi_j = \pi_{f(i)}$.*

Proof: Suppose that a DET P of a valid embedding $\mathcal{G}$ of G starts with an edge e_s of $\mathcal{B}$ but ends with an edge not in $\mathcal{B}$, where $\mathcal{B}$ is the block of $\mathcal{G}$ that is a different embedding of B. Let P^* be the dual trail of P. Since μ is doubly good, neither $\mathcal{B}$ nor $\mathcal{B}^*$ has an odd-degree nonpole. Moreover, since G^* is not Eulerian, P^* must start at an odd-degree vertex of $\mathcal{B}^*$ by Facts 1 and 2. Thus, P^* starts at a pole of $\mathcal{B}^*$. On the other hand, P starts at a pole of $\mathcal{B}$. This is because: (1) If G is Eulerian, P starts and ends at the same vertex of $\mathcal{B}$. Since e_s is in $\mathcal{B}$ and the ending edge e_t is not in $\mathcal{B}$, their common end vertex must be a pole of $\mathcal{B}$; and (2) if G is not Eulerian, P starts at an odd-degree vertex of $\mathcal{B}$ by Facts 1 and 2. By these facts, $P \cap \mathcal{B}$ is of a σ-type; this type is unique by Lemma 8. Let σ_i be this type.

To determine the π-type of $P \cap \mathcal{B}$, we first note that if $P \cap \mathcal{B}$ is of Type σ_1, σ_2 or σ_3, then $P \cap \mathcal{B}$ consits of a single trail and hence can only be of Type π_1. Thus, $\pi_1 = \pi_{f(i)}$ for $i = 1, 2, 3$. Now, suppose that $P \cap \mathcal{B}$ is of Type σ_4. Then, $P \cap \mathcal{B}$ consists of two double trails P_1 and P_2 where P_1 starts with e_s. Moreover, P_1 starts and ends at the same pole of $\mathcal{B}$, and so does P_2; the dual trail P_1^* of P_1 starts and ends at different poles of $\mathcal{B}^*$, and so does the dual trail P_2^* of P_2. Thus, $P \cap \mathcal{B}$ is of Type π_4. On the other hand, $P \cap \mathcal{B}$ is not of Type π_1 because $P \cap \mathcal{B}$ consists of two double trails; $P \cap \mathcal{B}$ is not of Type π_2 or π_3 either because P_2 starts and ends at the same pole of $\mathcal{B}$. Therefore, $\pi_4 = \pi_{f(4)}$. Similarly, we can verify that $\pi_3 = \pi_{f(5)}$. □

The mapping from σ-types to δ-types is specified by the following function $g : \{1, 2, 3, 5\} \to \{1, \ldots, 4\}$ where $g(1) = 2$, $g(2) = 4$, $g(3) = 3$, and $g(5) = 1$.

Lemma 23 *Suppose that G is Eulerian. Let μ be a doubly-good non-root node of T. Let $B = \mathrm{Pt}(\mu)$. Suppose that δ_j is a useful δ-type of B. Then, the following two statements hold:*

1. *If μ is not an S-node, then $\mathcal{K}_\sigma(B) = \{\sigma_5\}$ and $\delta_j = \delta_{g(5)}$.*
2. *If μ is an S-node, then $\mathcal{K}_\sigma(B) = \{\sigma_i\}$ for some $\sigma_i \in \{\sigma_1, \sigma_2, \sigma_3\}$, and $\delta_j = \delta_{g(i)}$.*

Proof: Suppose that a DET P of a valid embedding $\mathcal{G}$ of G starts with an edge e_s of $\mathcal{B}$ and ends with an edge e_t of $\mathcal{B}$, where $\mathcal{B}$ is the block of $\mathcal{G}$ that is a different embedding of B. Let P^* be the dual trail of P. Since G is Eulerian but G^* is not, P starts and ends at the same vertex of $\mathcal{B}$ (say, v) while P^* starts and ends at different vertices of $\mathcal{B}^*$, by Facts 1 and 2. Both e_s and e_t are incident to v.

Case 1: v is a pole of $\mathcal{B}$. Then, since μ is doubly good, P^* starts at one pole of $\mathcal{B}^*$ and ends at the other pole. Thus, $P \cap \mathcal{B}$ is of a σ-type. Moreover, since $\mathcal{B} \neq \mathcal{G}$, $P \cap \mathcal{B}$ contains more than one trail (otherwise, the single trail in $P \cap \mathcal{B}$ would be the entire DET P). Hence, $P \cap \mathcal{B}$ is of Type σ_4 or σ_5. Indeed, it is not of Type σ_4, because P starts and ends at the same vertex. Thus, $P \cap \mathcal{B}$ is of Type σ_5 and hence consists of two double trails P_1 and P_2 where P_1 starts with e_s and P_2 ends with e_t. To determine the δ-type of $P \cap \mathcal{B}$, we note that P_1 starts at v and ends at the other pole u of $\mathcal{B}$ while P_2 starts at u and ends at v. Moreover, since $P \cap \mathcal{B} = (P_1, P_2)$ is of Type σ_5, the dual trail P_1^* of P_1 starts and ends at the same pole x of $\mathcal{B}^*$ while the dual trail P_2^* of P_2 starts and ends at the other pole y of $\mathcal{B}^*$. Thus, $P \cap \mathcal{B}$ is of Type δ_1. On the other hand, $P \cap \mathcal{B}$ is not of Type δ_4, δ_5 or δ_6 because $P \cap \mathcal{B}$ consists of only two double trails. Moreover, $P \cap \mathcal{B}$ is not of Type δ_2 because the ending pole of P_1^* is different from the starting pole of P_2^*. Similarly, $P \cap \mathcal{B}$ is not of Type δ_3 because the ending pole of P_1 is the same as the starting pole of P_2. Therefore, $\delta_1 = \delta_{g(5)}$.

Case 2: v is not a pole of $\mathcal{B}$. Then, since P^* starts at one pole of $\mathcal{B}^*$ and ends at the other pole, one of e_s and e_t is on the west side of $\mathcal{B}$ and the other is on the east side. Thus, v is a cut vertex of $\mathcal{B}$ and μ is an S-node. Let $\mathcal{B}_1$ (respectively, $\mathcal{B}_2$) be the block of $\mathcal{B}$ whose source (respectively, sink) is that of $\mathcal{B}$ and whose sink (respectively, source) is v. Then, $\mathcal{B} = \mathcal{B}_1 \star \mathcal{B}_2$. Since $\mathcal{B}^*$ is good, neither $\mathcal{B}_1^*$ nor $\mathcal{B}_2^*$ has an odd-degree nonpole. We assume that e_s is on the west side of $\mathcal{B}_1$; the other cases are similar.

Case 2.1: $\mathcal{B}_1$ contains e_t. Then, e_t is on the east side of $\mathcal{B}_1$. Moreover, P^* starts at the source of $\mathcal{B}_1^*$ and ends at the sink of $\mathcal{B}_1^*$, by Facts 1 and 2. As in Case 1, we can prove that $P \cap \mathcal{B}_1$ is of Type σ_5 and hence consists of two double trails P_1 and P_3 where P_1 starts with e_s and P_3 ends with e_t. On the other hand, $P \cap \mathcal{B}_2$ must start at the sink of $\mathcal{B}_2$ and can only leave $\mathcal{B}_2$ at its sink; so, $P \cap \mathcal{B}_2$ consists of a single trail P_2 and is of Type σ_2. Consequently, $P \cap \mathcal{B}$ is of Type $\sigma_5 \star \sigma_2 = \sigma_2$ by Table 1. To determine the δ-type of $P \cap \mathcal{B}$, we note that P_1 starts at v and ends at the other pole u of $\mathcal{B}_1$ while P_3 starts at u and ends at v. Moreover, P_2 starts at the pole $w \neq v$ of $\mathcal{B}_2$ and ends at w. Since u, v, w are three distinct vertices of $\mathcal{B}$, no two of the trails P_1 through P_3 can be concatenated into a single trail in $P \cap \mathcal{B}$. So, $P \cap \mathcal{B} = (P_1, P_2, P_3)$ because P_1 starts with e_s and P_3 ends with e_t. In turn, $P \cap \mathcal{B}$ is of Type δ_4. On the other hand, $P \cap \mathcal{B}$ is not of Type δ_1, δ_2 or δ_3 because $P \cap \mathcal{B}$ consists of three trails; $P \cap \mathcal{B}$ is not of Type δ_5 or δ_6 because P_2 starts and ends at the same vertex of $\mathcal{B}$. Therefore, $\delta_4 = \delta_{g(2)}$.

Case 2.2: $\mathcal{B}_1$ does not contain e_t. Then, e_t is on the east side of $\mathcal{B}_2$. Moreover, by Facts 1 and 2, P^* starts at the source of $\mathcal{B}_1^*$ and ends at the sink of $\mathcal{B}_2^*$. By this and the fact that P starts and ends at the common pole v of $\mathcal{B}_1$ and $\mathcal{B}_2$, $P \cap \mathcal{B}_1$ must start at v and cannot re-enter $\mathcal{B}_1$ after leaving it, while $P \cap \mathcal{B}_2$ must start at the sink of $\mathcal{B}_2$ and cannot leave $\mathcal{B}_2$ after entering it. Consequently, $P \cap \mathcal{B}_1$ consists of a single double trail P_1, and $P \cap \mathcal{B}_2$ consists of a single double trail P_2 too. So, both $P \cap \mathcal{B}_1$ and $P \cap \mathcal{B}_2$ are of Type π_1. In turn, $P \cap \mathcal{B} = (P_1, P_2)$ is of Type δ_2 or δ_3 by Table 5; $P \cap \mathcal{B}$ cannot be of Type δ_0 or else $\mathcal{B}$ could not have been a proper block of G. Moreover, each of P_1 and P_2 starts and ends at different vertices; so each of $P \cap \mathcal{B}_1$ and $P \cap \mathcal{B}_2$ is of Type σ_1 or σ_3.

Suppose that one of $P \cap \mathcal{B}_1$ and $P \cap \mathcal{B}_2$ is of Type σ_1 and the other is of Type σ_3. We assume that $P \cap \mathcal{B}_1$ is of Type σ_1 and $P \cap \mathcal{B}_2$ is of Type σ_3; the other case is similar. Then, the unique σ-type of $\mathcal{B}$ is $\sigma_1 \star \sigma_3 = \sigma_1$ by Table 1. Note that the starting vertex of P^* (which is also the starting vertex of the dual trail P_1^* of P_1) and the ending vertex of P^* (which is also the ending vertex of the dual trail P_2^* of P_2) are different vertices in G^*. Since $P \cap \mathcal{B}_1 = (P_1)$ is of Type σ_1, the starting and the ending vertices of P_1^* must be different. On the other hand, since $P \cap \mathcal{B}_2 = (P_2)$ is of Type σ_3, the starting and the ending vertices of P_2^* must be the same. These requirements together imply that the ending vertex of P_1^* and the starting vertex of P_2^* must be the same pole of $\mathcal{B}^*$. Therefore, the ODTC $P \cap \mathcal{B} = (P_1, P_2)$ is not of Type δ_3 but is of Type δ_2. So, $\delta_2 = \delta_{g(1)}$.

Suppose that both of $P \cap \mathcal{B}_1$ and $P \cap \mathcal{B}_2$ are of Type σ_1 or both of them are of Type σ_3. Then, the unique σ-type of $\mathcal{B}$ is σ_3 by Table 1. Similarly to the above case where $P \cap \mathcal{B}_1$ and $P \cap \mathcal{B}_2$ are of different σ-types, we can prove that the ending vertex of the dual trail P_1^* of P_1 and the starting vertex of the dual trail P_2^* of P_2 must be different poles of $\mathcal{B}^*$. Hence, the

ODTC $P \cap \mathcal{B} = (P_1, P_2)$ is not of Type δ_2 but is of Type δ_3. So, $\delta_3 = \delta_{g(3)}$. $\square$

Lemma 24 *Suppose that G is not Eulerian. Let μ be a doubly-good non-root node of T. Let $B = \mathrm{Pt}(\mu)$. Suppose that δ_j is a useful δ-type of B. Then, $\mathcal{K}_\sigma(B)$ equals $\{\sigma_4\}$ or $\{\sigma_5\}$ and $\delta_j = \delta_3$.*

Proof: Suppose that a DET P of a valid embedding $\mathcal{G}$ of G starts with an edge e_s of $\mathcal{B}$ and ends with an edge e_t of $\mathcal{B}$, where $\mathcal{B}$ is the block of $\mathcal{G}$ that is a different embedding of B. Let P^* be the dual trail of P. Since $\mathcal{B}$ is a proper block of $\mathcal{G}$, $P \cap \mathcal{B}$ consists of at least two trails P_1 and P_2 where P_1 starts with e_s and P_2 ends with e_t. Since μ is doubly good, neither $\mathcal{B}$ nor $\mathcal{B}^*$ has an odd-degree nonpole. By Facts 1 and 2, P starts at a pole of $\mathcal{B}$ and ends at the other pole of $\mathcal{B}$, and P^* starts at a pole of $\mathcal{B}^*$ and ends at the other pole of $\mathcal{B}^*$. So, both e_s and e_t are access edges of $\mathcal{B}$. In turn, each of P_1 and P_2 uses exactly two access edges of $\mathcal{B}$. Since $\mathcal{B}$ has at most four access edges, P_1 and P_2 are the only two trails in $P \cap \mathcal{B}$. Therefore, $P \cap \mathcal{B} = (P_1, P_2)$ is of Type σ_4 or σ_5. Let P_1^* and P_2^* be the dual trail of P_1 and P_2, respectively.

Case 1: $P \cap \mathcal{B}$ is of Type σ_4. Then, P_1 starts at a pole u of $\mathcal{B}$ and ends at u while P_1^* starts at a pole x of $\mathcal{B}^*$ and ends at the other pole y of $\mathcal{B}^*$. Because neither $\mathcal{G}$ nor $\mathcal{G}^*$ is Eulerian, P_2 starts at the other pole v of $\mathcal{B}$ and ends at v while P_2^* starts at x and ends at y. Thus, $P \cap \mathcal{B}$ is of Type δ_3.

Case 2: $P \cap \mathcal{B}$ is of Type σ_5. Then, P_1 starts at a pole u of $\mathcal{B}$ and ends at the other pole v of $\mathcal{B}$ while P_1^* starts at a pole x of $\mathcal{B}^*$ and ends at x. Because neither $\mathcal{G}$ nor $\mathcal{G}^*$ is Eulerian, P_2 starts at u and ends at v while P_2^* starts at the other pole y of $\mathcal{B}^*$ and ends at y. Thus, $P \cap \mathcal{B}$ is of Type δ_3. $\square$

Let μ be a node of T, and ν be a child of μ in T. A σ-, π-, or δ-type γ_1 of $\mathrm{Pt}(\mu)$ is *consistent* with a σ-, π-, or δ-type γ_2 of $\mathrm{Pt}(\nu)$, if a different embedding H of $\mathrm{Pt}(\mu)$ has a Type-γ_1 DTC $\mathcal{Q}$ such that $\mathcal{Q} \cap B$ is a Type-γ_2 DTC of B, where B is the block of H that is a different embedding of $\mathrm{Pt}(\nu)$.

To describe the improved algorithm concisely, we abuse the notation to use $\mathcal{K}_\pi(B)$ (respectively, $\mathcal{K}_\delta(B)$) to mean its arbitrary subset that contains all useful π-types (respectively, δ-types) of a block B of G.

The algorithm works as follows. First, for all nodes μ of T, we determine if μ is a good node of T, and compute the pole-parity multiset of $\mathrm{Pt}(\mu)$ if so; we also determine if μ is a good node of T^*, and compute the pole-parity multiset of the dual graph of $\mathrm{Pt}(\mu)$ if so. By Lemma 19, this takes $O(|G|)$ time in total. Then, we find the critical nodes of T based on Lemma 19. If T has more than two critical nodes, then we stop immediately since G has no valid embedding by Lemma 20. So, suppose that T has one or two critical nodes. For all doubly-good non-root nodes μ of T, we compute $\mathcal{K}_\sigma(\mu)$ as in Section 4.3. After this, we distinguish two cases as follows.

- *Case 1*: T has two critical nodes, say μ and μ'. Let $B = \mathrm{Pt}(\mu)$ and $B' = \mathrm{Pt}(\mu')$. Then, μ is not an ancestor of μ' and vice versa. Thus, both μ and μ' are doubly bad, and every DET of a valid embedding of G starts with an edge of one of B and B' and ends with an edge of the other. We need to compute $\mathcal{K}_\pi(B)$ and $\mathcal{K}_\pi(B')$. After knowing $\mathcal{K}_\pi(B)$ and $\mathcal{K}_\pi(B')$, we are done by calling the algorithm in Section 5.4 to compute $\mathcal{K}_\delta(G)$. We describe the computation of $\mathcal{K}_\pi(B)$ below. $\mathcal{K}_\pi(B')$ is computed similarly. Let $\nu_1, \ldots, \nu_q$ be the children of μ in T. For each $i \in \{1, \ldots, q\}$, let e_i be the virtual edge of ν_i in $\mathrm{Sk}(\mu)$. If $|\mathcal{K}_\sigma(\nu_i)| = 0$ for some $i \in \{1, \ldots, q\}$, then $\mathcal{K}_\pi(B) = \emptyset$ by Lemma 22. So, suppose $\mathcal{K}_\sigma(\nu_i)$ consists of a unique $\sigma_{k_i} \in \Sigma$ for all $i \in \{1, \ldots, q\}$.

- *Case 1.1*: μ is an R-node. Let $H = \mathrm{Sk}(\mu)$. By an *extreme vertex* of H, we mean a pole of H or a nonpole v of H such that for an odd number of edges e_i incident to v in H, ν_i is an O-good node of T. Similarly, by an *extreme face* of H, we mean the west or east side of H or an inner face F of H such that for an odd number of edges e_i on the boundary of F, ν_i is an O-good node of T^*. If H has more than three extreme vertices, then it has at least two odd-degree nonpoles x and y. In every DET P of G, each of x and y must be either the starting vertex or the ending vertex. However, we only allow one of the starting and the ending vertices of P to be in H. Hence, G has no DET in this case. Similarly, if H has more than three extreme faces, then G has no valid embedding. Let W be the set of all children ν_i of μ such that e_i is incident to an extreme vertex of H and is also on an extreme face of H. Since μ is an R-node, for each (extreme) vertex v of H and each (extreme) face F of H, at most two edges incident to v are on F; so, a very rough estimate shows $|W| \leq 18$. Moreover, every DET of a valid embedding of G either starts or ends with an edge of $\mathrm{Pt}(\nu_i)$ for some $\nu_i \in W$. Thus, for each $\nu_i \in W$, we set $\mathcal{K}_\pi(\mathrm{Pt}(\nu_i)) = \{\pi_{f(k_i)}\}$, and then compute the set Π_i of all π-types of B consistent with Type $\pi_{f(k_i)}$ of $\mathrm{Pt}(\nu_i)$ as in Case 3.1 in Section 5.4. By Lemma 22 and Statements 1 and 2 in Lemma 4, $\cup_{\nu_i \in W} \Pi_i$ is $\mathcal{K}_\pi(B)$.
- *Case 1.2*: μ is an S-node (respectively, P-node). For each $j \in \{1, \ldots, 5\}$, let $I_j = \{i \in \{1, \ldots, q\} \mid \sigma_{k_i} = \sigma_j\}$. For each $j \in \{1, \ldots, 5\}$ such that $I_j \neq \emptyset$, we select an arbitrary $i \in I_j$, set $\mathcal{K}_\pi(\mathrm{Pt}(\nu_i)) = \{\pi_{f(k_i)}\}$, and compute the set Π_j of all π-types of B consistent with Type $\pi_{f(k_i)}$ of $\mathrm{Pt}(\nu_i)$ as in Case 3.2 (respectively, 3.3) in Section 5.4. By Lemma 22 and Statements 1 and 2 in Lemma 4, $\cup_{1 \leq j \leq 5, I_j \neq \emptyset} \Pi_j$ is $\mathcal{K}_\pi(B)$.

- *Case 2*: T has exactly one critical node, say μ. Let $B = \mathrm{Pt}(\mu)$. If μ is the root of T, we need to compute $\mathcal{K}_\delta(B)$; otherwise, we need to compute $\mathcal{K}_\pi(B)$ and $\mathcal{K}_\delta(B)$. The former case is easier, and we here only consider the latter case. So, we assume that μ is not the root. Then, μ is doubly bad. Let $\nu_1, \ldots, \nu_q$ be the children of μ in T. For each $i \in \{1, \ldots, q\}$, let e_i be the virtual edge of ν_i in $\mathrm{Sk}(\mu)$. If $\mathcal{K}_\sigma(\nu_i) = \emptyset$ for some $i \in \{1, \ldots, q\}$, then $\mathcal{K}_\pi(B) = \mathcal{K}_\delta(B) = \emptyset$ by Lemmas 22, 23, and 24. So, suppose $\mathcal{K}_\sigma(\nu_i) = \{\sigma_{k_i}\}$ for all $i \in \{1, \ldots, q\}$. For each $j \in \{1, \ldots, 5\}$, let $I_j = \{i \in \{1, \ldots, q\} \mid \sigma_{k_i} = \sigma_j\}$. For each $j \in \{1, 2, 3\}$, let $I'_j = \{i \in I_j \mid \nu_i \text{ is an S-node}\}$. Let $I'_5 = \{i \in I_5 \mid \nu_i \text{ is not an S-node}\}$.

  - *Case 2.1*: μ is an R-node. $\mathcal{K}_\pi(B)$ can be computed as in Case 1.1. We next explain the computation of $\mathcal{K}_\delta(B)$. Let $H = \mathrm{Sk}(\mu)$. As in Case 1.1, we define *extreme vertices* and *extreme faces* of H. If H has more than four extreme vertices, then it has at least three odd-degree nonpoles and each of them must be the starting or the ending vertex of any DET of G. Hence, G has no DET in this case. Similarly, if H has more than four extreme faces, then G has no valid embedding. If G is not Eulerian, we define a set W of children of μ as in Case 1.1; otherwise, let W be the set of all children ν_i of μ such that (1) e_i is on an extreme face of H and (2) at least one endpoint of e_i is on two extreme faces of H. Since μ is an R-node, for each pair of (extreme) faces of H, at most two vertices can appear on both of them; so, a very rough estimate shows $|W| \leq 96$. Moreover, every DET of a valid embedding of G that starts and ends both with edges of B must start with an edge of $\mathrm{Pt}(\nu_i)$ and end with an edge of $\mathrm{Pt}(\nu_j)$ for some $\nu_i, \nu_j \in W$ (possibly $\nu_i = \nu_j$). Thus, for each $\nu_i \in W$ and each $\nu_j \in W$ with $\nu_i \neq \nu_j$, we set $\mathcal{K}_\pi(\mathrm{Pt}(\nu_i)) = \{\pi_{f(k_i)}\}$ and $\mathcal{K}_\pi(\mathrm{Pt}(\nu_j)) = \{\pi_{f(k_j)}\}$, and further compute the set $\Delta_{i,j}$ of all δ-types of B consistent with both Type $\pi_{f(k_i)}$ of $\mathrm{Pt}(\nu_i)$ and Type $\pi_{f(k_j)}$ of $\mathrm{Pt}(\nu_j)$, as in Case 1.1 in Section 5.4. Moreover, for each $\nu_i \in W$, we compute a set Δ_i as follows. In case G is not Eulerian, if $\sigma_{k_i} \in \{\sigma_4, \sigma_5\}$, then we set $\mathcal{K}_\delta(\mathrm{Pt}(\nu_i)) = \{\delta_3\}$, and set Δ_i to be the set of all δ-types of B consistent with Type δ_3 of $\mathrm{Pt}(\nu_i)$, as in Case 2.1 in Section 5.4; otherwise, we set $\Delta_i = \emptyset$. In case G is Eulerian, if either (1) ν_i is not an S-node and $\sigma_{k_i} = \sigma_5$ or (2) ν_i is an S-node and $\sigma_{k_i} \in \{\sigma_1, \sigma_2, \sigma_3\}$, then we set $\mathcal{K}_\delta(\mathrm{Pt}(\nu_i)) = \{\delta_{g(k_i)}\}$, and set Δ_i to be the set of all δ-types of B consistent with Type $\delta_{g(k_i)}$ of $\mathrm{Pt}(\nu_i)$, as in Case 2.1 in Section 5.4; otherwise, we set $\Delta_i = \emptyset$. By Lemmas 4, 22, 23, and 24, the union of the computed sets $\Delta_{i,j}$ and Δ_i is $\mathcal{K}_\delta(B)$.

- *Case 2.2*: μ is an S-node (respectively, P-node). $\mathcal{K}_\pi(B)$ can be computed as in Case 1.2. We next explain the computation of $\mathcal{K}_\delta(B)$. For each $j \in \{1, \ldots, 5\}$ with $|I_j| \geq 2$, we arbitrarily select two distinct $i, i' \in I_j$, set $\mathcal{K}_\pi(\mathrm{Pt}(\nu_i)) = \{\pi_{f(k_i)}\}$ and $\mathcal{K}_\pi(\mathrm{Pt}(\nu_{i'})) = \{\pi_{f(k_{i'})}\}$, and compute the set Δ_j of all δ-types of B consistent with both Type $\pi_{f(k_i)}$ of $\mathrm{Pt}(\nu_i)$ and Type $\pi_{f(k_{i'})}$ of $\mathrm{Pt}(\nu_{i'})$, as in Case 1.2 (respectively, 1.3) in Section 5.4. Also, for each pair of distinct $j, j' \in \{1, \ldots, 5\}$ with $I_j \neq \emptyset$ and $I_{j'} \neq \emptyset$, we arbitrarily select one $i \in I_j$ and one $i' \in I_{j'}$, set $\mathcal{K}_\pi(\mathrm{Pt}(\nu_i)) = \{\pi_{f(k_i)}\}$ and $\mathcal{K}_\pi(\mathrm{Pt}(\nu_{i'})) = \{\pi_{f(k_{i'})}\}$ and compute the set $\Delta_{j,j'}$ of all δ-types of B consistent with both Type $\pi_{f(k_i)}$ of $\mathrm{Pt}(\nu_i)$ and Type $\pi_{f(k_{i'})}$ of $\mathrm{Pt}(\nu_{i'})$, as in Case 1.2 (respectively, 1.3) in Section 5.4. Furthermore, in case G is Eulerian, for each $j \in \{1, 2, 3, 5\}$ with $I'_j \neq \emptyset$, we arbitrarily select one $i \in I'_j$, set $\mathcal{K}_\delta(\mathrm{Pt}(\nu_i)) = \{\delta_{g(k_i)}\}$, and compute the set Δ'_j of all δ-types of B consistent with Type $\delta_{g(k_i)}$ of $\mathrm{Pt}(\nu_i)$, as in Case 2.2 (respectively, 2.3) in Section 5.4. On the other hand, in case G is not Eulerian, for each $j \in \{4, 5\}$ with $I_j \neq \emptyset$, we arbitrarily select one $i \in I_j$, set $\mathcal{K}_\delta(\mathrm{Pt}(\nu_i)) = \{\delta_3\}$, and compute the set Δ'_j of all δ-types of B consistent with Type δ_3 of $\mathrm{Pt}(\nu_i)$, as in Case 2.2 (respectively, 2.3) in Section 5.4. By Lemmas 4, 22, and 23, the union of the computed sets Δ_j, $\Delta_{j,j'}$, and Δ'_j is $\mathcal{K}_\delta(B)$.

After processing the unique critical node μ, we know $\mathcal{K}_\pi(B)$ and $\mathcal{K}_\delta(B)$. Note that μ is doubly bad and $\mathcal{K}_\sigma(B) = \emptyset$. We then proceed to processing the parent χ of μ. Let $B_\chi = \mathrm{Pt}(\chi)$. Let $\mu_1, \ldots, \mu_p$ be the children of χ where $\mu_1 = \mu$. For each $i \in \{1, \ldots, p\}$, let e'_i be the virtual edge of μ_i in $\mathrm{Sk}(\chi)$. Since B or B^* has an odd-degree nonpole, each DET of a valid embedding of G must start and/or end with an edge of B. So, for each $i \in \{2, \ldots, p\}$, if $\mathcal{K}_\sigma(\mathrm{Pt}(\mu_i)) = \emptyset$ then $\mathcal{K}_\pi(\mathrm{Pt}(\mu_i)) = \emptyset$ too and hence G has no valid embedding. Thus, we may assume each $\mathcal{K}_\sigma(\mathrm{Pt}(\mu_i))$ consists of a unique $\sigma_{\ell_i} \in \Sigma$. $\mathcal{K}_\pi(B_\chi)$ can be computed from $\mathcal{K}_\pi(B)$, $\mathcal{K}_\sigma(\mathrm{Pt}(\mu_2)), \ldots, \mathcal{K}_\sigma(\mathrm{Pt}(\mu_p))$, as in Case 3 in Section 5.4. To compute $\mathcal{K}_\delta(B_\chi)$, we first compute the set Δ_0 of all δ-types of B_χ consistent with a δ-type in $\mathcal{K}_\delta(B)$, as in Case 2 in Section 5.4. After this, for each $\pi_h \in \mathcal{K}_\pi(B)$, we compute a set Δ_h as follows.

- *Case 2.3*: χ is an R-node. We use Lemma 21 to compute the unique pole-status S of the different embeddings of B that have a π-type DTC. Let $H = \mathrm{Sk}(\chi)$. Let v_1 and v_2 be the endpoints of e'_1. Let F_1 and F_2 be the end endpoints of the dual edge of e'_1. For each list $L = (a_1, a_2, b_1, b_2)$ such that $S = (\langle a_1, a_2 \rangle, \langle b_1, b_2 \rangle)$, we compute a set Δ_L as follows. For each vertex $v \notin \{v_1, v_2\}$ of H, we define v to be an *extreme vertex* of H as in Case 1.1. Similarly, for each vertex $F \notin \{F_1, F_2\}$ of H^*, we define

F to be an *extreme face* of H as in Case 1.1. For each $j \in \{1,2\}$, we say that v_j is an *extreme vertex* of H if (1) v_j is a pole of H, or (2) $a_j = 0$ and an odd number of edges $e'_i \neq e'_1$ incident to v_j satisfy that μ_i is an O-good node of T, or (3) $a_j = 1$ and an even number of edges $e'_i \neq e'_1$ incident to v_j satisfy that μ_i is an O-good node of T. Similarly, for each $j \in \{1,2\}$, we say that F_j is an *extreme face* of H if (1) F_j is a pole of H^*, or (2) $b_j = 0$ and an odd number of edges $e'_i \neq e'_1$ on F_j satisfy that μ_i is an O-good node of T^*, or (3) $b_j = 1$ and an even number of edges $e'_i \neq e'_1$ on F_j satisfy that μ_i is an O-good node of T^*. As in Case 2.1, H has at most four extreme vertices and at most four extreme faces, unless G has no valid embedding. As in Case 2.1, we define a set W of children μ_i of χ. Let $W' = W - \{\mu\}$. For each $\mu_i \in W'$, we set $\mathcal{K}_\pi(\mathrm{Pt}(\mu_i)) = \{\pi_{f(\ell_i)}\}$, and then compute the set Δ'_i of all δ-types of B_χ consistent with both Type π_h of $\mathrm{Pt}(\mu)$ and Type $\pi_{f(\ell_i)}$ of $\mathrm{Pt}(\mu_i)$, as in Case 1.1 in Section 5.4. Let $\Delta_L = \cup_{\mu_i \in W'} \Delta'_i$. Let Δ_h be the union of the computed sets Δ_L.

- *Case 2.4*: χ is an S-node (respectively, P-node). For each $j \in \{1, \ldots, 5\}$, let $I''_j = \{i \in \{2, \ldots, p\} \mid \sigma_{\ell_i} = \sigma_j\}$. For each $j \in \{1, \ldots, 5\}$ with $|I''_j| \geq 1$, we arbitrarily select one $i \in I''_j$, set $\mathcal{K}_\pi(\mathrm{Pt}(\mu_i)) = \{\pi_{f(\ell_i)}\}$, and compute the set Δ'_j of all δ-types of B_χ consistent with both Type π_h of $\mathrm{Pt}(\mu)$ and Type $\pi_{f(\ell_i)}$ of $\mathrm{Pt}(\mu_i)$, as in Case 1.2 (respectively, 1.3) in Section 5.4. Let Δ_h be the union of the computed sets Δ'_j.

By Lemmas 4, 22, 23, 24, and 21, $\mathcal{K}_\delta(B_\chi) = \Delta_0 \cup (\cup_{\pi_h \in \mathcal{K}_\pi(B)} \Delta_h)$. This completes the processing of χ; each ancestor of χ is processed in the same way. Once we finish processing the root of T, we have $\mathcal{K}_\delta(G)$; G has a valid embedding if and only if $\delta_0 \in \mathcal{K}_\delta(G)$.

Obviously, processing each node μ of T takes $O(|\mathrm{Sk}(\mu)|)$ time. Thus, we have:

Theorem 3 *Given a TTG G whose dual graph is not Eulerian, it takes $O(|G|)$ time to decide if G has a valid embedding.*

Acknowledgment: The authors would like to thank the referee for suggestions that considerably improved the readability of this paper.

References

[1] Bradley S. Carlson, C. Y. Roger Chen, and Dikran S. Meliksetian. Dual Eulerian properties of plane multigraphs. *SIAM J. Disc. Math.*, 8(1):33–50, Feb. 1995.

[2] G. Di Battista and R. Tamassia. On-line maintenance of triconnected components with SPQR-trees. *Algorithmica*, 15:302–318, 1996.

[3] G. Di Battista and R. Tamassia. On-line planarity testing. *SIAM J. Comput.*, 25(5):956–997, 1996.

[4] S. Even and R.E. Tarjan. Computing an *st*-numbering. *Theoretical Computer Science*, 2:339–344, 1976.

[5] Tung-Yang Ho, Ting-Yi Sung, Lih-Hsing Hsu, Chang-Hsiung Tsai, and Jeng-Yan Hwang. The recognition of double Euler trails in series-parallel networks. *J. Algorithm*, 28:216–257, 1998.

[6] A. Lempel, S. Even, and I. Cederbaum. An algorithm for planarity testing of graphs. In *Theory of Graphs, International Symposium*, pages 215–232, Gordon Breach, New York, 1967.

[7] Thomas Lengauer and Rolf Müller. Linear algorithms for optimizing the layout of dynamic CMOS cells. *IEEE Tran. on Circuits and Systems*, 35(3):279–285, March 1988.

[8] Robert L. Maziasz and John P. Hayes. Layout optimization of static CMOS functional cells. In *Proc. 24th ACM/IEEE Design Automation Conference (DAC'87)*, pages 544–551, 1987.

[9] Robert L. Maziasz and John P. Hayes. Layout optimization of static CMOS functional cells. *IEEE Tran. on Computer-Aided Design*, 9(7):708–719, July 1990.

[10] Ravi Nair, Anni Buss, and John Reif. Linear time algorithms for optimal CMOS layout. In P. Bertolazzi and F. Luccio, editors, *VLSI: Algorithms and Architectures*, pages 327–338, North Holland, 1985. Elsevier Science Publishers.

[11] T. Nishizeki and N. Chiba. *Planar Graphs: Theory and Algorithms.* North-Holland, 1988.

[12] P. Rosenstiehl and R. E. Tarjan. Rectilinear planar layouts and bipolar orientations of planar graphs. *Discr. and Comp. Geometry*, 1:343–353, 1986.

[13] Takao Uehara and William M. VanCleemput. Optimal layout of CMOS functional arrays. *IEEE Tran. on Computers*, C-30(5):305–312, May 1981.

[14] Shuichi Ueno, Kafsufumi Tsuji, and Yoji Kajitani. On dual Eulerian paths and in plane graphs. In *Proc. IEEE International Symposium on Circuits and Systems*, pages 1835–1838, Espoo, Finland, June 7-9, 1988.

[15] Shuichi Ueno, Kafsufumi Tsuji, and Yoji Kajitani. A note on dual trail partition of a plane graph. *IEICE Tran.*, E 74(7):1915–1917, July 1991.