IPSJ SIG Technical Report

Vol.2016-MPS-108 No.11
Vol.2016-BIO-46 No.11
2016/7/4

Grammar-based Compression for Unrooted Trees and Subclass of Plane Graphs Using Integer Linear Programming

Morihiro Hayashida[1,a)] Yue Cao[1] Yang Zhao[2]

Abstract: We address a problem of finding generation rules from biological data. In the previous study, the extraction of generation rules from glycans and RNAs represented by rooted tree structures was examined. Grammars were defined for rooted ordered and unordered trees, and methods for finding the minimum grammars that produce only given tree structures were developed. In nature and organisms, however, there are various kinds of structures other than rooted tree ones. In this study, we relax the limitation of structures to be compressed, and propose grammars representing unrooted trees and some subclass of plane graphs together with an integer linear programming-based method for finding its minimum grammars.

1. Introduction

Data compression is related with information that it contains. Our purpose is to extract useful information from data through compression. In the previous study, we developed integer linear programming-based methods for finding the minimum grammar that represents given multiple rooted trees, and applied to glycans and RNAs [1]. It, however, is considered that data in nature cannot be always represented by rooted trees. Fig. 1 shows rooted and unrooted trees. The rooted tree in the left is the same as the unrooted tree in the right if the rooted tree becomes unrooted. Although it is difficult to find substructures in the rooted tree, we can find several substructures in the unrooted tree. In this study, we relax the limitation of structures to be compressed, and propose grammars for unrooted trees and some subclass of planar graphs with an integer programming-based method for finding the minimum grammar for ordered version of its grammars.

2. Method

We define a grammar for an unrooted edge-labeled tree by 4-tuple $(\Sigma, \Gamma, S, \Delta)$ as an extension of [2], where Σ is a set of terminal symbols (labeled edges), Γ is a set of nonterminal symbols, S is the start symbol in Γ, and Δ is a set of production rules represented by Fig. 2. A nonterminal symbol corresponds to a connected unrooted subtree, which links to the remaining part with at most two nodes, called tags. If a subtree with two tags is not symmetric, either tag must be determined to link to the node of one side. That is the reason why direction is attached on a nonterminal symbol.

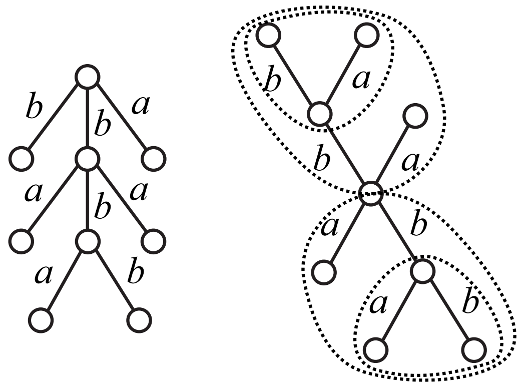

Fig. 1 Example of rooted and unrooted trees.

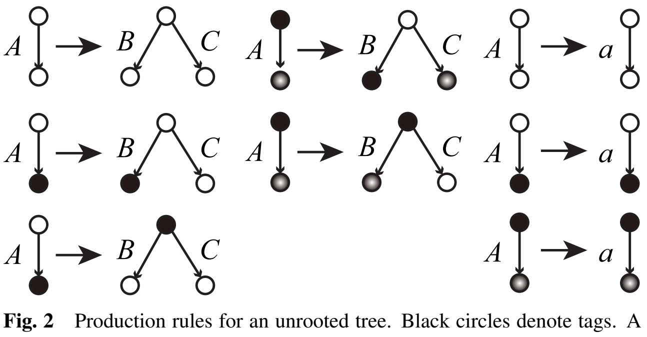

Fig. 2 Production rules for an unrooted tree. Black circles denote tags. A nonterminal symbol includes at most two tags.

In addition to the grammar, we also define a grammar for a subclass of planar graphs to deal with graphs including closed paths by adding production rules represented by Fig. 3.

Fig. 4 shows an example of a plane graph and its grammar that generates only the graph. The edge-labeled graph is transformed from the purine chemical compound.

Let $G(V, E)$ be an undirected planar graph with a set V of ver-

1 Institute for Chemical Research, Kyoto University, Gokasho, Uji, Kyoto 611–0011, Japan
2 The National Institute of Advanced Industrial Science and Technology, 2-4-7, Aomi, Koto, Tokyo 135–0064, Japan
a) morihiro@kuicr.kyoto-u.ac.jp

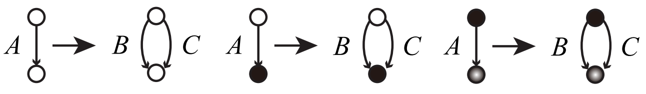

Fig. 3 Additional production rules for a planar graph.

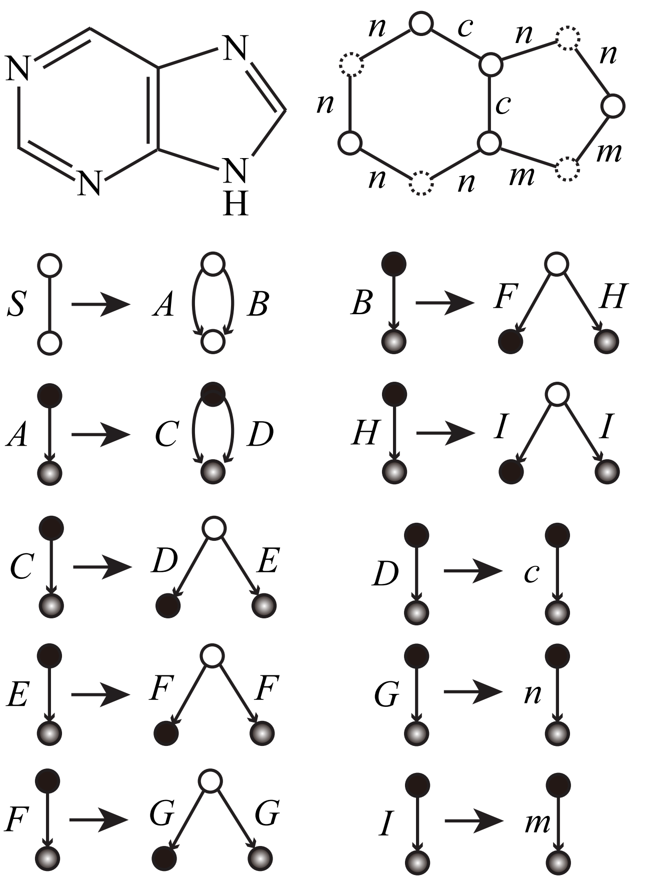

Fig. 4 Example of a plane graph and its grammar. The edge-labeled graph (the upper right) is transformed from the purine chemical compound (the upper left). Dashed circles denote nitrogen atoms.

tices and a set E of labeled edges. Consider the following cutting procedure from G to $G'(V', E')$ at a vertex i with a set S of vertices, where $S \subset E(i) = \{j \in V | (i, j) \in E\}$.

(i) add a new vertex i' to G (i.e., $V' = V \cup \{i'\}$).

(ii) replace edges connected to i which endpoints are in S with edges connected to i' (i.e., $E' = E \cup \{(i', j) | j \in S\} - \{(i, j) | j \in S\}$).

Then, if G' is not connected and consists of two connected components, we say that G is separable with i and S. In a similar way, we consider the cutting procedure from G at two vertices i and j with two sets S and T, where $S \subset E(i)$ and $T \subset E(j)$. $G_{i,S,j,T}(V_{i,S,j,T}, E_{i,S,j,T})$ is defined as a connected subgraph of G obtained by the cutting procedure, which includes non-empty sets S, T, does not include $E(i) - S$ and $E(j) - T$, where $i, j \in V \cup \{\epsilon\}$, $S \subset E(i)$, $T \subset E(j)$, ϵ means nothing. G is represented by $G_{\epsilon,\emptyset,\epsilon,\emptyset}$. $G_{i,S,j,T}$ is distinguished from $G_{j,T,i,S}$ in almost all cases. Suppose that $\mathcal{S}(G_{i,S,j,T})$ and $\mathcal{I}(G_{i,S,j,T})$ are sets of all subgraphs $G_{i',S',j',T'}$ and its indices (i', S', j', T'), respectively, of $G_{i,S,j,T}$ by the cutting procedure. Consider the case that $G_{i,S,j,T}$ is decomposed into $G_{i',S',j',T'}$ and $G_{i'',S'',j'',T''}$. Let $C(G_{i,S,j,T})$ be a set of all index combinations $(i', S', j', T', i'', S'', j'', T'')$ that $V_{i',S',j',T'} \cup V_{i'',S'',j'',T''} = V_{i,S,j,T}$, $V_{i',S',j',T'} \cap V_{i'',S'',j'',T''} = \{i, j\}$, $E_{i',S',j',T'} \cup E_{i'',S'',j'',T''} = E_{i,S,j,T}$, $E_{i',S',j',T'} \cap E_{i'',S'',j'',T''} = \emptyset$, $E_{i',S',j',T'} \neq \emptyset$, and $E_{i'',S'',j'',T''} \neq \emptyset$.

$\mathcal{I}(G)$ contains a large number of elements for unordered trees and planar graphs, and the number of nonterminal symbols becomes also large. We consider an ordered version of these grammars, and represent $S \in E(i)$ by two nodes, starting and ending nodes in clockwise direction. Then, a nonterminal symbol is represented by $G_{i,s,e,j,t,f}$ for $s, e \in E(i)$ and $t, f \in E(j)$ instead of $G_{i,S,j,T}$, and we propose the following integer linear programming formulation for finding the minimum grammar representing only a given plane graph G.

$$\min \sum_{u \in \mathcal{S}(G)} p_u$$

subject to

$$x_{\epsilon,\emptyset,\epsilon,\emptyset} = 1,$$

$$x_{i,s,e,j,t,f} = 1 \quad \text{for all } (i, s, e, j, t, f) \in \mathcal{I}(G) \text{ s. t. } |E_{i,s,e,j,t,f}| = 1,$$

$$x_{i,s,e,j,t,f} \leq \sum_{(i',s',e',j',t',f',i'',s'',e'',j'',t'',f'') \in C(G_{i,s,e,j,t,f})} y_{i',s',e',j',t',f',i'',s'',e'',j'',t'',f''}$$

$$\text{for all } (i, s, e, j, t, f) \in \mathcal{I}(G) \text{ s. t. } |E_{i,s,e,j,t,f}| \geq 2,$$

$$y_{i',s',e',j',t',f',i'',s'',e'',j'',t'',f''} \leq \frac{1}{2}(x_{i',s',e',j',t',f'} + x_{i'',s'',e'',j'',t'',f''}),$$

$$s_u \leq p_u < 1 + s_u \quad \text{for all } u \in \mathcal{S}(G),$$

$$s_u = \frac{1}{|E|} \sum_{\{(i,s,e,j,t,f) \in \mathcal{I}(G) | G_{i,s,e,j,t,f} \text{ is isomorphic to } u\}} x_{i,s,e,j,t,f}.$$

3. Discussion

In this study, we proposed the definition of grammars for unrooted trees and some subclass of planar graphs, and integer linear programming-based method for finding the minimum grammar representing only a given plane graph. We can prove that the proposed grammar generates only a planar graph. It, however, is difficult to concretely show the subclass of planar graphs that the grammar can generate, and to determine whether or not our method can be applied to a given plane graph. As future work, we would like to uncover the subclass that the grammar can generate, apply the proposed method to actual plane graphs, and extend the grammar to more complicated structures.

Acknowledgements

This work was partially supported by Grant-in-Aid #16K00392 from MEXT, Japan.

References

[1] Zhao, Y., Hayashida, M., Cao, Y., Hwang, J. and Akutsu, T.: Grammar-based compression approach to extraction of common rules among multiple trees of glycans and RNAs, *BMC Bioinformatics*, Vol. 16, p. 128 (2015).

[2] Akutsu, T.: A bisection algorithm for grammar-based compression of ordered trees, *Information Processing Letters*, Vol. 110, pp. 815–820 (2010).