Jindalertudomdee *et al. BMC Bioinformatics* (2016) 17:113
DOI 10.1186/s12859-016-0962-4

BMC Bioinformatics

RESEARCH ARTICLE Open Access

Enumeration method for tree-like chemical compounds with benzene rings and naphthalene rings by breadth-first search order

Jira Jindalertudomdee, Morihiro Hayashida*, Yang Zhao and Tatsuya Akutsu

Abstract

Background: Drug discovery and design are important research fields in bioinformatics. Enumeration of chemical compounds is essential not only for the purpose, but also for analysis of chemical space and structure elucidation. In our previous study, we developed enumeration methods *BfsSimEnum* and *BfsMulEnum* for tree-like chemical compounds using a tree-structure to represent a chemical compound, which is limited to acyclic chemical compounds only.

Results: In this paper, we extend the methods, and develop *BfsBenNaphEnum* that can enumerate tree-like chemical compounds containing benzene rings and naphthalene rings, which include benzene isomers and naphthalene isomers such as ortho, meta, and para, by treating a benzene ring as an atom with valence six, instead of a ring of six carbon atoms, and treating a naphthalene ring as two benzene rings having a special bond. We compare our method with MOLGEN 5.0, which is a well-known general purpose structure generator, to enumerate chemical structures from a set of chemical formulas in terms of the number of enumerated structures and the computational time. The result suggests that our proposed method can reduce the computational time efficiently.

Conclusions: We propose the enumeration method BfsBenNaphEnum for tree-like chemical compounds containing benzene rings and naphthalene rings as cyclic structures. BfsBenNaphEnum was from 50 times to 5,000,000 times faster than MOLGEN 5.0 for instances with 8 to 14 carbon atoms in our experiments.

Keywords: Benzene ring, Naphthalene ring, Enumeration, Breadth-first search

Background

Enumeration of chemical compounds is important in bioinformatics, and has been adapted to several applications such as drug discovery and design [1–3], structure elucidation [4–6], and analyses of chemical spaces [7–13]. It is defined as a problem of generating all non-redundant chemical structures satisfying some constraints. For example, a chemical formula, which consists of the number of each atom included in the compound, is given as an input. There are several algorithms for enumerating chemical compounds from a chemical formula and most of them use a molecular graph to represent a chemical compound, where the nodes and edges of the graph refer to atoms and bonds of the chemical compound, respectively. Some of those algorithms are claimed to be able to enumerate various chemical structures without restriction of the structure, such as MOLGEN [14] and Open Molecule Generator (OMG) [15]. It was reported that OMG is able to deal with different valences for a kind of atom, and was not efficient for several instances compared with MOLGEN. While the remaining ones, such as EnuMol [16, 17] as well as *BfsSimEnum* and *BfsMulEnum* [18], have a limitation of the structure of enumerated compounds, such as acyclic compounds for BfsSimEnum and BfsMulEnum and compounds with no cycle except for benzene rings for EnuMol, the methods consume significantly less computational time. There are also related

*Correspondence: morihiro@kuicr.kyoto-u.ac.jp
Bioinformatics Center, Institute for Chemical Research, Kyoto University, Gokasho, Uji, Japan

application softwares, e.g. SmiLib [19] and CLEVER [20], that generate chemical compounds from given fragments. The limitation of these tools is that they require a library of desired chemical fragments, which can be generated by the enumeration tool.

Our previous methods, BfsSimEnum and BfsMulEnum, use a tree structure, instead of a general graph, to represent a chemical compound and call it *a molecular tree* so they can generate only tree-like chemical compounds. In this work, we develop *BfsBenNaphEnum*, which aims to reduce the limitation of previous methods by extending them such that they can enumerate chemical compounds containing only benzene rings and naphthalene rings as cyclic structures, which are six carbon atoms cyclic structures and ten carbon atoms bicyclic structures, respectively. Pólya proposed a group-theoretic method for isomer counting of single cyclic structures such as a benzene ring, a naphthalene ring, and an anthracene ring using the cycle index, from which many studies followed [21]. However, structures enumerated by these methods are restricted to certain types. Indeed, Meringer wrote that up to now the only way to calculate the number of isomers belonging to an arbitrary molecular formula is to use structure generators [22]. Suzuki et al. considered the problem of enumerating structures having monocyclic graph structures, each of which has exactly one cycle [23]. An enumeration method for tree-like chemical compounds containing only benzene rings as cyclic structures has been implemented on EnuMol web server (http://sunflower.kuicr.kyoto-u.ac.jp/tools/enumol/). On the other hand, our method can enumerate compounds containing naphthalene rings in addition to benzene rings. Moreover, the proposed algorithm can calculate the number of benzene rings and naphthalene rings from chemical formula, while users have to specify the number of benzene rings in EnuMol.

Chemical structures considered in this study can be represented by a molecular tree, where a benzene ring is converted to a node with valence six and a naphthalene ring is considered as two benzene nodes having a special bond. We name that special bond as *a merge bond*. Since a merge bond merges two carbon atoms of two benzene rings together, it reduces the number of carbon atoms with free valence electron of two benzene rings by two so we represent a merge bond by a double-edge. Moreover, benzene nodes cannot have double bonds with other nodes because they bond with other non-benzene atoms by a single bond [24]. This means that a double-edge represents a double bond if it connects two non-benzene nodes, while it represents a merge bond if it connects two benzene nodes. Therefore, bonds in a benzene ring and a naphthalene ring are considered as the same bond and Kekulé representation is not included in this work. Besides, this work uses a two-dimensional molecular tree to represent a chemical structure so it cannot deal with stereoisomers. For tautomeric, this work considers two structures in a pair of tautomeric as non- redundant compounds and generates both of them.

BfsSimEnum and BfsMulEnum are modified to return a set of molecular trees as the output, given a chemical formula, the number of benzene rings, and the number of naphthalene rings. After that, an attribute called *carbon position list* is added into benzene nodes in a molecular tree to represent the way that benzene nodes bond with their adjacent nodes. This attribute is important because bonding with different carbon atoms in a benzene ring may result in different chemical structures. Finally, for each molecular tree from BfsSimEnum and BfsMulEnum, we generate a set of molecular trees whose nodes adjacent to benzene nodes are labeled with a carbon position such that all chemical structures are enumerated without redundancy based on normal form rule.

For evaluating our proposed method, we perform computational experiments for several instances, and compare the execution time by our method with that by MOLGEN. We show that our proposed method is efficient for enumerating chemical compounds containing benzene rings and naphthalene rings, and is from 50 times to 5,000,000 times faster than MOLGEN for several instances in our experiments.

Preliminaries

Enumeration problem

Let Σ be a finite set of labels of atoms, for example, $\Sigma = \{C,N,O,H\}$, where 'C', 'N', 'O', and 'H' denote carbon, nitrogen, oxygen, and hydrogen atoms, respectively. A *molecular graph* is defined as a multi-graph $G(V, E)$, where V is a set of nodes and E is a set of multi-edges, also denoted by $V(G)$ and $E(G)$, respectively. Each node is labeled with an atom-label in Σ, while each edge represents the bond between two atoms and the multiplicity of edge represents the bond type. The degree of each node is equal to the valence of its atom. Let $deg(v)$ and $l(v)$ be the degree and the label of node v, respectively. Let $val(l_i)$ be the valence of the atom represented by label l_i in Σ. It should be noted that there exist different valences for a kind of atom, for example, carbon atoms of CO_2 and CO. For this case, it is sufficient to put two distinct labels C and $C^{(2)}$ in Σ, and to define $val(C) = 4$ and $val\left(C^{(2)}\right) = 2$. Let $num\ (G,l_i)$ be the total number of nodes labeled with label l_i in molecular graph G. Then, the enumeration problem is defined as follows.

Problem 1. *Given the numbers n_{l_i} of atoms for all labels $l_i \in \Sigma$, the number n_b of benzene rings, and the number n_n of naphthalene rings, enumerate all non-redundant molecular graphs G such that $num(G,l_i) = n_{l_i}$ for all $l_i \in \Sigma$, $deg(v) = val(l(v))$ for all nodes $v \in V(G)$, and G includes*

exactly n_b benzene rings, n_n naphthalene rings, and no other cyclic structures. It must be noted that n_b and n_n can be zero.

In the case that the input chemical formula contains five or less carbon atoms, BfsStructEnum can enumerate only tree-like chemical compounds by specifying the number of benzene rings and the number of naphthalene rings to be zero. Because we enumerate molecular trees such that degree of each node equals to valence of atom label of that node, charged molecules cannot be enumerated automatically. However, they can still be enumerated by specifying a charged atom as a new kind of atom type with appropriate valence value.

Since our enumeration methods deal with a chemical compound as a node-labeled rooted ordered tree for efficient enumeration, we contract cyclic structures appearing in a molecular graph to single nodes. Concretely, we contract a benzene ring to a node, called *benzene node*, labeled with a special label 'b', and contract a naphthalene ring to two benzene nodes connected by a special bond, called *merge bond*, represented by a double edge (see Fig. 1). Since six carbon atoms contained in a benzene ring are contracted into a benzene node, we need to remember which carbon atom in the benzene ring connects to its adjacent node in a molecular graph. Hence, we add an attribute called *carbon position list* to each benzene node. Figure 1b shows examples of carbon position lists using numbers assigned to carbon atoms in benzene rings in Fig. 1a. We call such a node-labeled rooted ordered tree whose benzene nodes are attributed with carbon position lists *a carbon position-assigned molecular tree*. We enumerate carbon position-assigned molecular trees instead of molecular graphs.

Center-rooted and left-heavy

In our previous work, we defined the normal form for molecular trees without any cyclic structures using *center-rooted* and *left-heavy* to avoid its redundant generation. In this work, we also utilize center-rooted and left-heavy for carbon position-assigned molecular trees, of which properties do not depend on carbon position lists.

A molecular tree T is called *center-rooted* if its root is the center node (see Fig. 2a) or one endpoint of the center edge of the longest path in T (see Fig. 2b). The center can be either a node or an edge depending on the length of the longest path.

In order to define a left-heavy tree, atom-labels must be ordered so that they can be compared with each other, for example, b> $C > N > O >$H for Σ = {b,C,N,O,H}, where 'b' denotes a special atom representing a benzene ring. Let $T(u)$ be the ordered subtree rooted at u in T. Let u and v be two nodes in a molecular tree T, $(u_1, u_2, \ldots, u_h)$ and $(v_1, v_2, \ldots, v_k)$ be lists of child nodes of u and v, respectively. It is defined that $T(u) >_s T(v)$ if $l(u) > l(v)$ (Fig. 3a) or there exists an integer i such that $T(u_j) =_s T(v_j)$ for all $j < i$ and $(T(u_i) >_s T(v_i)$ (Fig. 3b) or $i = k + 1 \leq h$ (Fig. 3c)). If $T(u) >_s T(v)$ or $T(v) >_s T(u)$ does not hold, it is said that $T(u) =_s T(v)$.

Let $mul(e)$ and $mul(u,v)$ be the multiplicity of edge $e = (u,v)$. Let $(e_1, e_2, \ldots, e_m)$ and $(e'_1, e'_2, \ldots, e'_m)$ be two lists of edges in $T(u)$ and $T(v)$ in breadth-first search (BFS) order (see Fig. 4), respectively. $T(u) >_m T(v)$ if $T(u) >_s T(v)$, or if $T(u) =_s T(v)$ and there exists an integer i such that $mul(e_j) = mul(e'_j)$ for all $j < i$, and $mul(e_i) > mul(e'_i)$ (Fig. 3d). If $T(u) >_m T(v)$ or $T(v) >_m T(u)$ does not hold, it is said that $T(u) =_m T(v)$.

Let $child(v) = (v_1, v_2, \ldots)$ be a list of all child nodes of node v in BFS order. It is defined that a molecular tree T is *left-heavy* if $T(v_i) \geq_m T(v_{i+1})$ holds for all nodes v in T and all $i = 1, \ldots, |child(v)| - 1$.

It should be noted that center-rooted and left-heavy are different from *centroid-rooted* and *left-heavy* defined by Fujiwara et al. [16], for example, the molecular tree in Fig. 1b is center-rooted and is not centroid-rooted because the number of nodes in the left subtree by removing the root, 4, is more than (total number of nodes $-1)/2 = (7 - 1)/2 = 3$. In addition, their left-heavy is defined

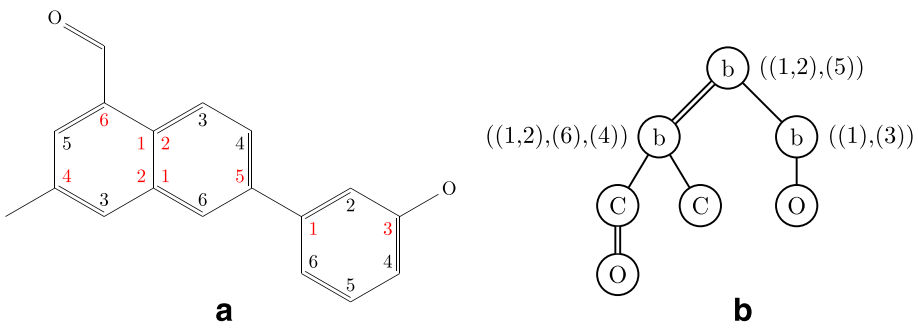

Fig. 1 Example of a molecular graph including benzene rings and naphthalene rings. **a** A molecular graph including one benzene ring and one naphthalene ring. **b** A rooted tree contracted from the left graph. It is noted that hydrogen atoms are omitted

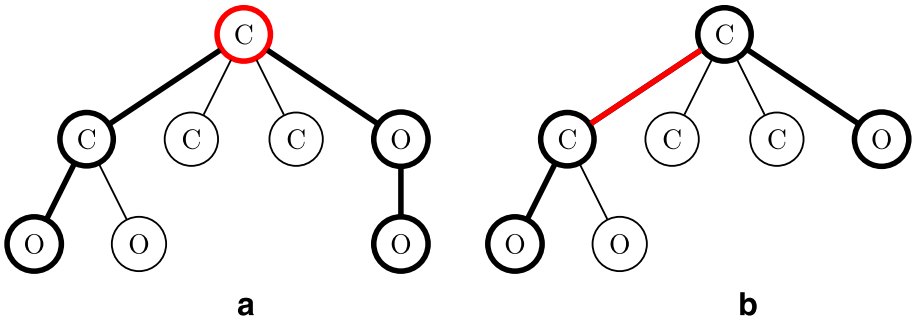

Fig. 2 Illustration of center-rooted molecular trees. **a** Center of the longest path is a node. **b** Center of the longest path is an edge. The *thick lines* indicate one of the longest paths and the center node/edge is shown in *red*

using depth-first search order, not our breadth-first search order.

Carbon position list

Let $s = (v_1, v_2, \ldots, v_n)$ be a list of nodes, $|s|$ and $s[i]$ denote the size and the i-th element of s, respectively. Let $T^{sub}(v_1, v_2)$ be the left-heavy tree rooted at v_1 that consists of the connected component including v_1 when the edge (v_1, v_2) is deleted from T (see Fig. 5). $T^{sub}(v_1, v_2) =_m T(v_1)$ if v_1 is a child of v_2 in T. Let *index*(v,T) be the order of $v \in V(T)$ by traversing a center-rooted left-heavy molecular tree T with BFS order, which is also denoted by *index*(v) if T is clear.

Proposition 1. *For a node v that has the parent node v_p and a child node v_c in a center-rooted molecular tree T, $T^{sub}(v_p, v) \neq_m T^{sub}(v_c, v)$.*

Proof. The height of $T^{sub}(v_p, v)$ is larger than that of $T^{sub}(v_c, v)$ because T is center-rooted. Hence, $T^{sub}(v_p, v)$ is always different from $T^{sub}(v_c, v)$. □

We define an equality $T_1 =_C T_2$ for two rooted carbon-position assigned trees T_1 and T_2 if $T_1 =_m T_2$, and $C_{v_1}^{T_1} = C_{v_2}^{T_2}$ for all benzene nodes $v_1 \in V(T_1)$, where $v_2 \in V(T_2)$ satisfies $index(v_1, T_1) = index(v_2, T_2)$, and C_v^T is a list of lists, called a carbon position list explained later, for a benzene node v in T. For convenience, we define another equality $T_1 =_{\underline{C}} T_2$ by removing the condition that $C_{r_1}^{T_1} = C_{r_2}^{T_2}$ for the roots r_1 and r_2 of T_1 and T_2, respectively, from the conditions of $T_1 =_C T_2$, if r_1 and r_2 are benzene nodes.

For a node v having the parent v_p and a child v_c, $T^{sub}(v_p, v) \neq_C T^{sub}(v_c, v)$ if $T^{sub}(v_p, v) \neq_m T^{sub}(v_c, v)$. Hence, only carbon position lists of descendent benzene nodes are needed to determine whether or not $T^{sub}(v_{c_1}, v) =_C T^{sub}(v_{c_2}, v)$ for child nodes v_{c_1} and v_{c_2} of v.

Definition 1. *An adjacent node list A_v^T of a benzene node v in a carbon position-assigned molecular tree T is defined as a list of lists of nodes adjacent to v using carbon position lists of descendent benzene nodes such that*

- $|A_v^T[i]| \leq |A_v^T[i+1]|$ *for all i,*

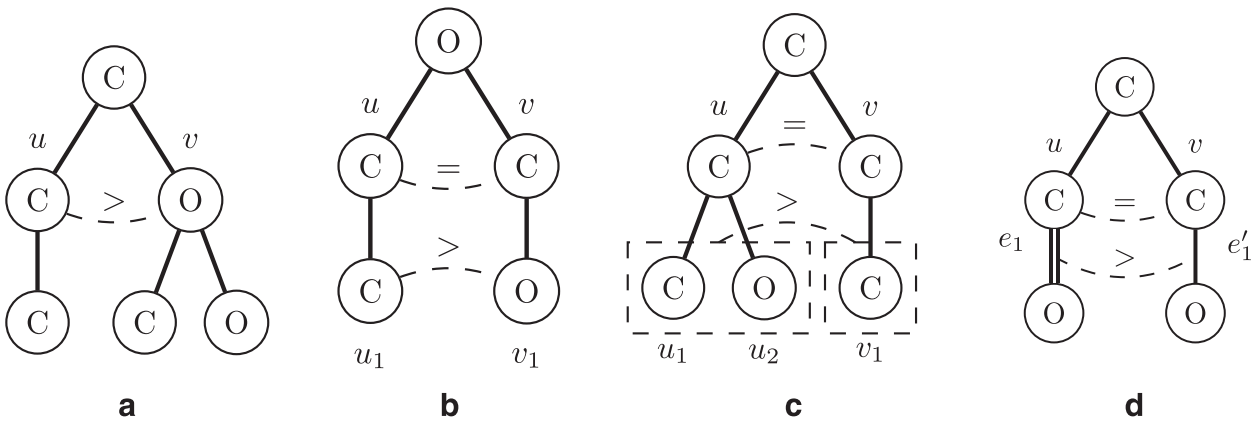

Fig. 3 Illustration of three molecular trees such that $T(u) >_s T(v)$ or $T(u) >_m T(v)$. **a** $l(u) > l(v)$. **b** $l(u) = l(v), T(u_1) >_s T(v_1)$. **c** $l(u) = l(v), T(u_1) =_s T(v_1), h = 2 > 1 = k$. **d** $T(u) =_s T(v), mul(e_1) > mul(e'_1)$

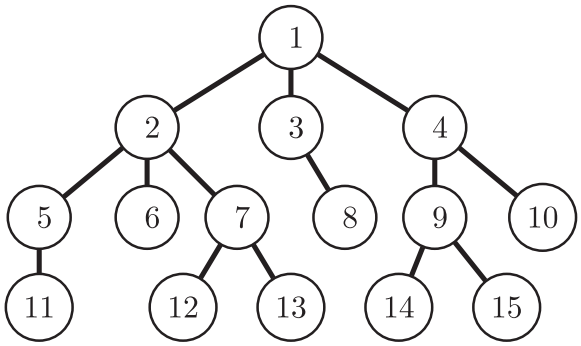

Fig. 4 Illustration of breadth-first search (BFS) order. *Numbers* indicate BFS order for this example

- $index\left(A_v^T[i][1]\right) < index\left(A_v^T[i+1][1]\right)$ *if* $|A_v^T[i]| = |A_v^T[i+1]|$,
- $index\left(A_v^T[i][j]\right) < index\left(A_v^T[i][j+1]\right)$ *for all* i, j,
- $A_v^T[i] = (v')$ *if* (v, v') *is a merge bond for some* i,
- $v' \in A_v^T[i]$ *if* (v, v') *is not a merge bond, and* $T^{sub}(v', v) =_C T^{sub}\left(A_v^T[i][1], v\right)$.

Figure 6 shows examples of carbon position-assigned molecular trees, where benzene node v_1 in each tree has adjacent nodes v_2, v_3, v_4, v_5. Then, $T_1^{sub}(v_2, v_1) =_C T_1^{sub}(v_3, v_1) \neq_C T_1^{sub}(v_4, v_1) \neq_C T_1^{sub}(v_5, v_1)$ and $index(v_4) < index(v_5)$, so we have $A_{v_1}^{T_1} = ((v_4), (v_5), (v_2, v_3))$. Also for T_2, $A_{v_1}^{T_2} = ((v_4), (v_5), (v_2, v_3))$. For T_3, $A_{v_1}^{T_3} = ((v_2), (v_3), (v_4), (v_5))$ because (v_2, v_1) is a merge bond. If (v_2, v_1) is not a merge bond and $C_{v_2}^{T_3} = C_{v_3}^{T_3}$, then $A_{v_1}^{T_3} = ((v_4), (v_5), (v_2, v_3))$.

Proposition 2. *For a benzene node* v *that has the parent node* v_p *in a center-rooted molecular tree* T, $A_v^T[1] = (v_p)$.

Proof. If v has no child, it is clear because the adjacent node of v is only v_p. We assume that v has a child v_c. From Proposition 1 and $index(v_p) < index(v_c)$, $A_v^T[1] = (v_p)$ always holds. □

A *carbon position list* C_v^T of a benzene node v in T is a list of lists, where $C_v^T[i]$ is a list of carbon positions of the nodes in $A_v^T[i]$. It is sufficient to enumerate $C_v^T[i]$ in ascending order because each node in $A_v^T[i]$ has the same subtree. If $\left(A_v^T[i][1], v\right)$ is a merge bond, $C_v^T[i]$ has two carbon positions instead of one as usual. It should be noted that $C_v^T[i] \subseteq \{1, \ldots, 6\}$ and two carbon positions are assigned for a merge bond because a naphthalene ring shares two carbon atoms between two benzene rings. In the examples of Fig. 6, $C_{v_1}^{T_1} = ((3), (4), (1, 2))$ for $A_{v_1}^{T_1} = ((v_4), (v_5), (v_2, v_3))$, $C_{v_1}^{T_2} = ((1), (4), (2, 3))$ for $A_{v_1}^{T_2} = ((v_4), (v_5), (v_2, v_3))$, $C_{v_1}^{T_3} = ((1, 2), (3), (5), (4))$ for $A_{v_1}^{T_3} = ((v_2), (v_3), (v_4), (v_5))$.

Definition 2. *An adjacent node list* $A_{(v_1,v_2)}^T$ *for a naphthalene ring with two benzene nodes* v_1, v_2, *where* (v_1, v_2) *is a merge bond, is defined as a list of lists of nodes adjacent to* v_1 *or* v_2 *except* v_1 *and* v_2 *such that*

- $|A_{(v_1,v_2)}^T[i]| \leq |A_{(v_1,v_2)}^T[i+1]|$ *for all* i,
- $index\left(A_{(v_1,v_2)}^T[i][1]\right) < index\left(A_{(v_1,v_2)}^T[i+1][1]\right)$ *if* $|A_{(v_1,v_2)}^T[i]| = |A_{(v_1,v_2)}^T[i+1]|$,
- $index\left(A_{(v_1,v_2)}^T[i][j]\right) < index\left(A_{(v_1,v_2)}^T[i][j+1]\right)$ *for all* i, j,
- $v' \in A_{(v_1,v_2)}^T[i]$ *if* $T^{sub}(v', bn(v')) =_C T^{sub}\left(A_{(v_1,v_2)}^T[i][1], bn(A_{(v_1,v_2)}^T[i][1])\right)$, *where* $bn(v)$ *is* v_1 *or* v_2 *that is adjacent to* v.

For a benzene node v_2 that is connected by a merge bond with the parent node v_1, we suppose that the carbon atoms having positions 1,2 in v_2 are connected with the carbon atoms having positions $\overline{x+1}, \overline{x}$ in v_1, respectively, where x takes an integer between 1 and 6, and

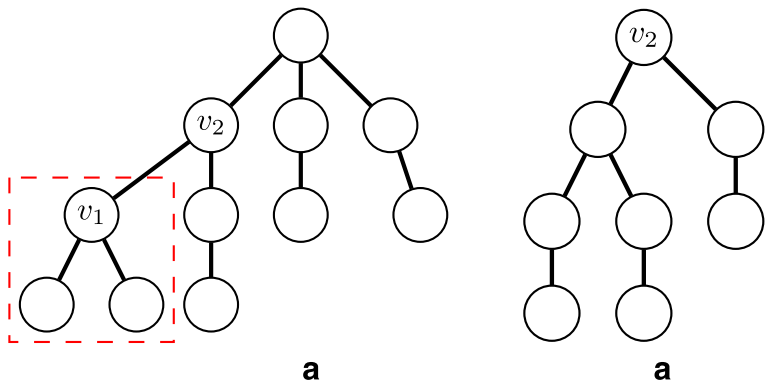

Fig. 5 Illustration of subtree $T^{sub}(v_1, v_2)$. **a** A molecular tree T and $T^{sub}(v_1, v_2)$, which is surrounded by a red rectangle. **b** $T^{sub}(v_2, v_1)$

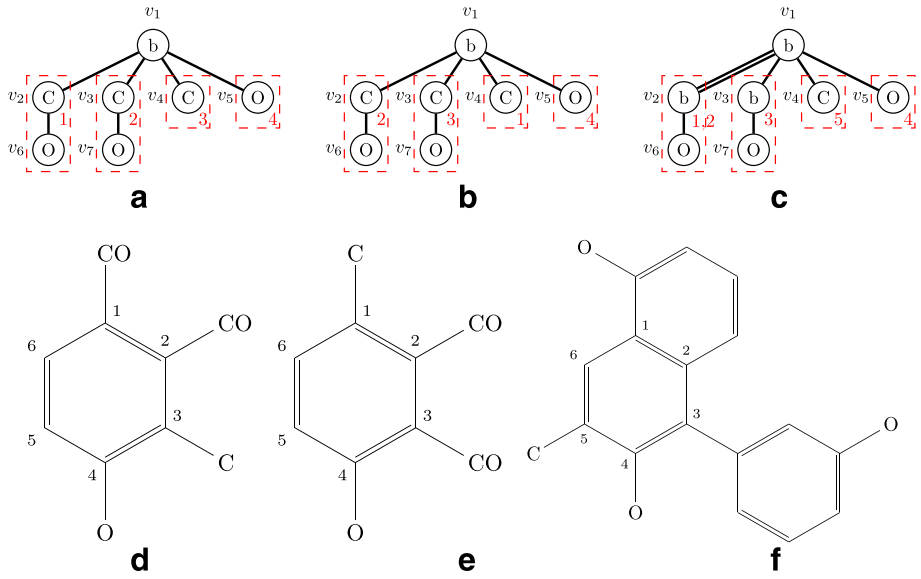

Fig. 6 Examples of adjacent node lists and carbon position lists. **a** T_1. **b** T_2. **c** T_3. **d** Molecular graph of T_1. **e** Molecular graph of T_2. **f** Molecular graph of T_3. Red numbers represent carbon positions of node v_1

$\overline{x} = (x \mod 6) + 1$ (see Fig. 7a). Here, consider the case that v_1 has the parent node v_p. If T is in normal form (Definition 6), position 1 is assigned to the carbon atom connected with v_p (Proposition 5). Then, from Proposition 1, $T^{sub}(v_p, v_1) \neq_C T^{sub}(v_c, v_2)$ for any child node v_c of v_2, $T^{sub}(v_p, v_1) \neq_C T^{sub}(v_c, v_1)$ for any child node v_c of v_1 except v_2, and the naphthalene ring is not symmetric. Consider the case that v_1 does not have a parent node, that is, v_1 is the root. If $T^{sub}(v_1, v_2) \neq_{\underline{C}} T^{sub}(v_2, v_1)$, the naphthalene ring can be symmetric only with respect to the axis denoted by the dashed red line in Fig. 7a. Then, it is not needed to consider the other symmetry for the naphthalene ring.

Consider the case that $T^{sub}(v_1, v_2) =_{\underline{C}} T^{sub}(v_2, v_1)$. We can prove that $x = 1$ if T is in normal form (see Proposition 4). Then, a carbon position list $C^T_{(v_1,v_2)}$ of a naphthalene ring consisting of two benzene nodes v_1, v_2 is a list of lists determined from $C^T_{v_1}$ and $C^T_{v_2}$ according to the following rule, where $C^T_{(v_1,v_2)}[i]$ is a list of carbon positions of nodes in $A^T_{(v_1,v_2)}[i]$ in ascending order.

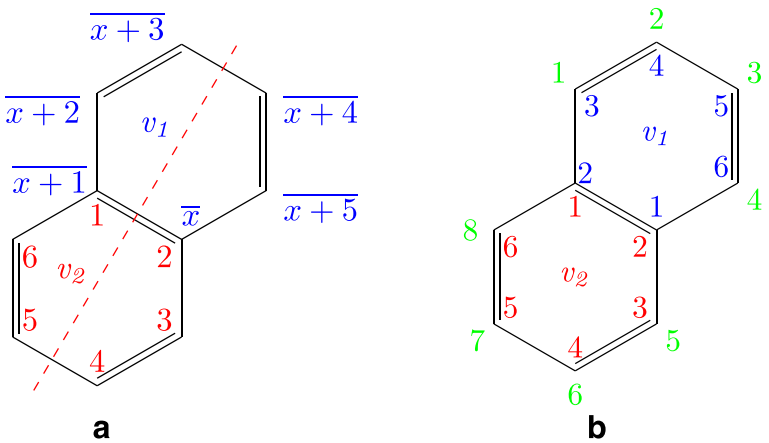

Fig. 7 Correspondence between carbon positions in a naphthalene ring. **a** Correspondence between carbon positions involved with a merge bond in two benzene rings. **b** Correspondence between carbon positions of a naphthalene ring and two benzene rings in the case of $T^{sub}(v_1, v_2) =_{\underline{C}} T^{sub}(v_2, v_1)$. The upper benzene ring v_1 is the parent of the lower benzene ring v_2. $\overline{x}$ denotes $(x \mod 6) + 1$. Blue, red, and green numbers are positions of $C^T_{v_1}$, $C^T_{v_2}$, and $C^T_{(v_1,v_2)}$, respectively. The dashed red line denotes the symmetric axis of ϕ_{ref}

Definition 3. *Carbon positions in a naphthalene ring correspond to carbon positions in two benzene nodes v_1, v_2, where v_1 is the parent node of v_2, if $T^{sub}(v_1, v_2) =_{\underline{C}} T^{sub}(v_2, v_1)$, as follows (see Fig. 7b).*

- *For the benzene ring of v_1, positions 1,2 are assigned to carbons of the merge bond in $C^T_{v_1}$. Position i $(i = 3, \ldots, 6)$ in $C^T_{v_1}$ corresponds to $i-2$ in $C^T_{(v_1,v_2)}$.*
- *For the benzene ring of v_2, positions 1,2 are assigned to carbons of the merge bond in $C^T_{v_2}$. Position i $(i = 3, \ldots, 6)$ in $C^T_{v_2}$ corresponds to $i+2$ in $C^T_{(v_1,v_2)}$.*

Figure 8 shows examples of carbon position lists for a naphthalene ring, where T'_4 is T_4 with $C^{T'_4}_{v_1} = ((1,2),(4),(3))$ and $C^{T'_4}_{v_2} = ((1,2),(4),(5))$, T''_4 is T_4 with $C^{T''_4}_{v_1} = ((1,2),(4),(5))$ and $C^{T''_4}_{v_2} = ((1,2),(4),(3))$. Then, $A^{T'_4}_{(v_1,v_2)} = A^{T''_4}_{(v_1,v_2)} = ((v_3,v_5),(v_4,v_6))$, $C^{T'_4}_{(v_1,v_2)} = ((2,6),(1,7))$, and $C^{T''_4}_{(v_1,v_2)} = ((2,6),(3,5))$.

Definition 4. *For carbon position lists $C^{T_1}_v$, $C^{T_2}_v$, where $A^{T_1}_v = A^{T_2}_v$, it is defined that $C^{T_1}_v < C^{T_2}_v$ if there exist two integers i and j such that*

- $C^{T_1}_v[i'][j'] = C^{T_2}_v[i'][j']$ *for all* $i' < i$ *and all* $j' = 1, \ldots, |C^{T_1}_v[i']|$,
- $C^{T_1}_v[i][j'] = C^{T_2}_v[i][j']$ *for all* $j' < j$,
- $C^{T_1}_v[i][j] < C^{T_2}_v[i][j]$.

This definition is applied to comparison of $C^{T_1}_{(v_1,v_2)}$ and $C^{T_2}_{(v_1,v_2)}$ for a naphthalene ring with v_1 and v_2 in the same way.

In the example of Fig. 6, T_1 and T_2 have the same tree structure, and $C^{T_2}_{v_1} = ((1),(4),(2,3)) < ((3),(4),(1,2)) = C^{T_1}_{v_1}$ because $C^{T_2}_{v_1}[1][1] = 1 < 3 = C^{T_1}_{v_1}[1][1]$.

Let Aut_b and Aut_n be the automorphism groups of a benzene ring and a naphthalene ring, respectively (see Fig. 9). Aut_b is generated from rotation of $\pi/3$ radians and reflection. For $\phi_b \in Aut_b$, v_1 is adjacent to v_2 in a benzene ring if and only if $\phi_b(v_1)$ is adjacent to $\phi_b(v_2)$ in a benzene ring. Aut_n is generated from rotation of π radians and reflection. We suppose that a list $\phi(C^T_v[i])$ of carbon positions for a map ϕ and $i = 1, \ldots, |C^T_v|$ is in ascending order by sorting elements of the list because all nodes in $A^T_v[i]$ have the same subtree. For example, $\phi_b(C^{T_1}_{v_1}) = ((6),(5),(1,2))$ for $C^{T_1}_{v_1} = ((3),(4),(1,2))$ and the reflection map ϕ_b by the perpendicular bisector between carbon atoms of 1 and 2.

Normal form of a carbon position-assigned molecular tree

In order to prevent generating redundant molecular trees in enumeration, we define a normal form of a carbon position-assigned molecular tree.

Definition 5. *Let P be a path in T consisting of n nodes $(v_1, v_2, \ldots, v_n)$ $(n \geq 2)$. P is called a symmetric path if the following conditions are satisfied.*

- $T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right) = {}_m T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)$,
- $index\left(v_i, T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right)\right) = index\left(v_{n-i+1}, T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)\right)$ *for all* $i = 1, \cdots, \lfloor \frac{n}{2} \rfloor$, *where* $\lfloor x \rfloor$ *is the largest integer less than or equal to x,*
- $C^T_v = C^T_{v'}$ *for all benzene nodes* $v \in V\left(T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right)\right) \setminus V\left(T^{sub}(v_1, v_2)\right)$, *where* $v' \in V\left(T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)\right)$ *satisfies* $index\left(v', T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)\right) = index\left(v, T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right)\right)$, *and* $v \in V_1 \setminus V_2$ *means that* $v \in V_1$ *and* $v \notin V_2$.

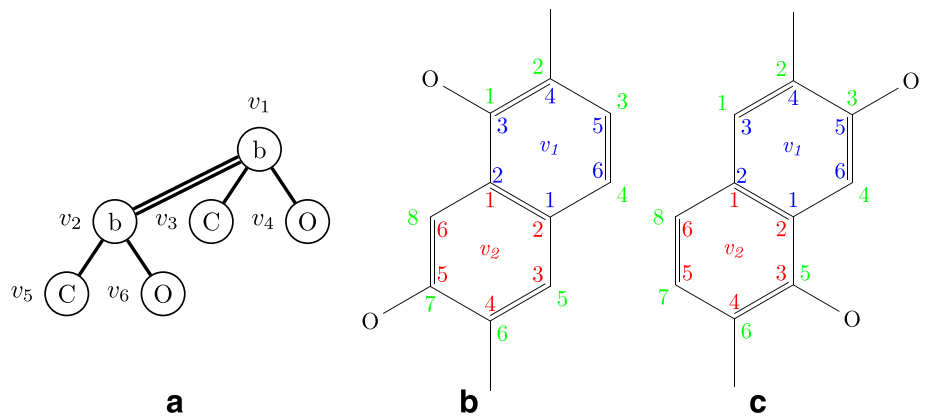

Fig. 8 Example of carbon position lists for a naphthalene ring. **a** T_4. **b** Molecular graph of T'_4, which is T_4 with $C^{T'_4}_{v_1} = ((1,2),(4),(3))$, $C^{T'_4}_{v_2} = ((1,2),(4),(5))$. **c** Molecular graph of T''_4, which is T_4 with $C^{T''_4}_{v_1} = ((1,2),(4),(5))$, and $C^{T''_4}_{v_2} = ((1,2),(4),(3))$

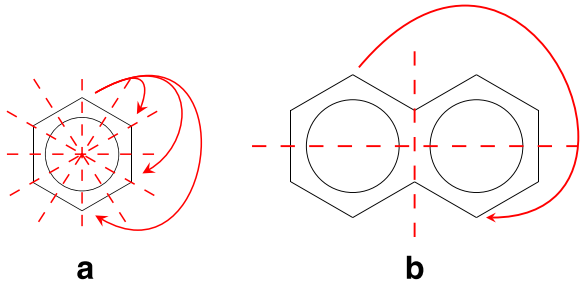

Fig. 9 Illustration of automorphism of a benzene ring and a naphthalene ring. **a** A benzene ring. **b** A naphthalene ring. *Dashed lines* indicate reflections, curves indicate rotations, where all automorphisms are not shown

Proposition 3. *For a center-rooted molecular tree, either of $v_{\frac{n}{2}}$ and $v_{\frac{n}{2}+1}$ is the root if the length of a symmetric path $(v_1, \cdots, v_n)$ is even. Otherwise, the depth of $v_{\frac{n+1}{2}}$ is less than that of any node in the path.*

Proof. For a path $(v_1, \cdots, v_n)$, v_{i+1} and v_{n-i} must be the parent nodes of v_i and v_{n-i+1}, respectively, for $i = 1, \cdots, \frac{n-1}{2}$ if n is odd and for $i = 1, \cdots, \frac{n}{2} - 1$ if n is even due to the center rooted property. Therefore, if the length of path is odd, $v_{\frac{n+1}{2}}$ is the parent node of both $v_{\frac{n+1}{2}-1}$ and $v_{\frac{n+1}{2}+1}$, which means that the depth of $v_{\frac{n+1}{2}}$ is less than that of any node in the path.

In the case that n is even, either $v_{\frac{n}{2}}$ or $v_{\frac{n}{2}+1}$ has the least depth among all nodes in the path and another node is the child node of that node. Assume that between these two nodes the parent node is v_a and the child node is v_b. v_a cannot have a parent node because the height of $T^{sub}(v_p, v_a)$, where v_p is the parent node of v_a, cannot be equal to the height of $T^{sub}(v_c, v_b)$ for any nodes v_c that are adjacent to v_b due the center-rooted condition, which means that $T^{sub}(v_a, v_b) =_m T^{sub}(v_b, v_a)$ cannot be hold and the first condition of symmetric path is violated. In other words, v_a, which is either $v_{\frac{n}{2}}$ or $v_{\frac{n}{2}+1}$, is the root node of the tree if n is even. □

We say that v_1 is *left* of v_n for a symmetric path $(v_1, \ldots, v_n)$ when $v_{n-\lfloor \frac{n}{2} \rfloor + 1}$ is the root, or $index(v_1) < index(v_n)$.

Figure 10 shows examples of symmetric paths, (v_2, v_1, v_3) in T_5 and (v_5, v_2, v_1, v_3) in T_6, where $T_5^{sub}(v_2, v_1) =_m T_5^{sub}(v_3, v_1)$, $T_6^{sub}(v_2, v_1) =_m T_6^{sub}(v_1, v_2)$, and $C_{v_4}^{T_6} = C_{v_6}^{T_6}$.

We define an inequality $T_1 >_C T_2$ for carbon position-assigned molecular trees T_1 and T_2 if $T_1 >_m T_2$, or $T_1 =_m T_2$, and there exists an integer i such that v_i is a benzene node, $C_{v_i}^{T_1} > C_{v_i'}^{T_2}$, and $C_{v_j}^{T_1} = C_{v_j'}^{T_2}$ for all benzene nodes v_j with $j > i$, where $index(v_k, T_1) = index(v_k', T_2)$ for all $k = 1, \ldots, |V(T_1)|$.

Definition 6. *Let ϕ_{ref} be the reflection map with the symmetric axis shown in Fig. 7a. A carbon position-assigned molecular tree T that contains a carbon position list C_v^T for each benzene node v is in normal form if the following conditions are satisfied.*

1. *T is center-rooted and left-heavy.*
2. *$T(v) \geq_m T^{sub}(r, v)$ if the center of the longest path in T with the root r is the edge (r, v).*
3. *Positions in each sublist of C_v^T for each benzene node v are in ascending order.*
4. *$C_v^T \leq \phi_b\left(C_v^T\right)$ for all benzene nodes v that is not connected by a merge bond with the parent node and all $\phi_b \in Aut_b$.*
5. *For benzene nodes v_1, v_2 connected by a merge bond such that v_1 is the root of T,*
   (a) *$C_{(v_1,v_2)}^T \leq \phi_n\left(C_{(v_1,v_2)}^T\right)$ for all $\phi_n \in Aut_n$ if $T^{sub}(v_1, v_2) =_C T^{sub}(v_2, v_1)$, where $C_{(v_1,v_2)}^T$ is related with $C_{v_1}^T$ and $C_{v_2}^T$ by Definition 3.*
   (b) *$C_{v_2}^T \leq \phi_{ref}\left(C_{v_2}^T\right)$ if $T^{sub}(v_1, v_2) \neq_C T^{sub}(v_2, v_1)$ and $C_{v_1}^T = \phi_{ref}\left(C_{v_1}^T\right)$.*
6. *$T^{sub}(v_1, v_2) \geq_C T^{sub}(v_n, v_{n-1})$ for all pairs v_1, v_n of nodes such that the path $(v_1, \ldots, v_n)$ is a symmetric path, v_1 and $v_n (= v_2)$ are not connected by a merge bond, and v_1 is left of v_n.*

We call a tree in normal form a *normal tree*.

Figure 8 also shows molecular trees in normal form and not in normal form. For condition 4 of the definition, $C_{v_1}^{T_4'} = ((1,2),(4),(3)) \leq \phi_b\left(C_{v_1}^{T_4'}\right)$, $C_{v_1}^{T_4''} = ((1,2),(4),(5)) \leq \phi_b\left(C_{v_1}^{T_4''}\right)$. T_4' and T_4'' satisfy conditions 1, 2, 3, and 4. For condition 5, $C_{(v_1,v_2)}^{T_4'} = ((2,6),(1,7)) \leq \phi_n\left(C_{(v_1,v_2)}^{T_4'}\right)$, whereas $C_{(v_1,v_2)}^{T_4''} = ((2,6),(3,5)) > ((2,6),(1,7)) = \phi_{rot}\left(C_{(v_1,v_2)}^{T_4''}\right)$ for rotation ϕ_{rot} of π radians, and T_4'' violates the condition. It is noted that T_4'' is rotated by π radians from T_4'. For condition 6, v_1 and v_2 are connected by a merge bond. Thus, T_4' is a normal tree, and T_4'' is not a normal tree.

Proposition 4. *For a normal tree T with a benzene node v_1 that is connected by a merge bond with its child node v_2 and satisfies $T^{sub}(v_1, v_2) =_C T^{sub}(v_2, v_1)$, positions 1,2 are assigned to the merge bond in the benzene ring of v_1. Furthermore, if $C_{(v_1,v_2)}^T \leq \phi_n\left(C_{(v_1,v_2)}^T\right)$ for all $\phi_n \in Aut_n$, then $C_{v_1}^T \leq \phi_b\left(C_{v_1}^T\right)$ for all $\phi_b \in Aut_b$.*

Proof. We assume that there exists a node v_l as a left sibling of v_2, and v_l is the leftmost child of v_1. Since T

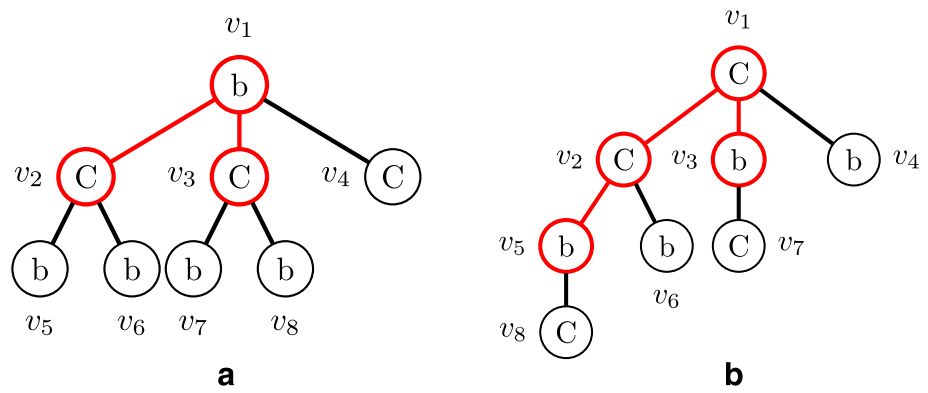

Fig. 10 Examples of symmetric paths. The *red lines* denote symmetric paths. **a** T_5, where (v_2, v_1, v_3) is a symmetric path, and $T_5^{sub}(v_2, v_1) =_m T_5^{sub}(v_3, v_1)$. **b** T_6, where (v_5, v_2, v_1, v_3) is a symmetric path, $T_6^{sub}(v_2, v_1) =_m T_6^{sub}(v_1, v_2)$ and $C_{v_4}^{T_6} = C_{v_6}^{T_6}$

is left-heavy, $T(v_l) \geq_m T(v_2)$, and $l(v_l) = l(v_2) =$'b' is needed. However, $T(v_l) =_C T(v_c)$, where v_c is the leftmost child of v_2, because $T^{sub}(v_1, v_2) =_C T^{sub}(v_2, v_1) =_C T(v_2)$. Hence, $T(v_l) <_m T(v_2)$. It contradicts the assumption, and v_2 is the leftmost child of v_1. Therefore, $A_{v_1}^T[1] = (v_2)$. From condition 4 of Definition 6, $C_{v_1}^T[1] = (1, 2)$, and positions 1,2 are assigned to the merge bond, that is $x = 1$ in Fig. 7a.

For a map $\phi_b \in Aut_b$ other than the identity and reflection map ϕ_{ref} for a benzene ring, $C_{v_1}^T < \phi_b\left(C_{v_1}^T\right)$ because each of $\phi_b(1)$ and $\phi_b(2)$ is at least 2. From $C_{(v_1,v_2)}^T \leq \phi_{ref}\left(C_{(v_1,v_2)}^T\right)$ and the correspondence between $C_{v_1}^T$ and $C_{(v_1,v_2)}^T$, $C_{v_1}^T \leq \phi_{ref}\left(C_{v_1}^T\right)$. Therefore, $C_{v_1}^T \leq \phi_b\left(C_{v_1}^T\right)$ for all $\phi_b \in Aut_b$. □

Proposition 5. *For a benzene node v of a normal tree T, $C_v^T[1][1]$ is always equal to 1.*

Proof. If v is not connected by a merge bond with the parent node, from condition 4, C_v^T must be the least possible carbon position list. Hence, $C_v^T[1][1] = 1$. Otherwise, from Definition 3, $C_v^T[1][1] = 1$. □

Lemma 1. *Given a molecular graph G without cyclic structures except benzene rings and naphthalene rings, G can be represented by a normal tree.*

Proof. We can assign numbers to carbons in benzene rings and naphthalene rings of G such that the conditions of Definition 6 are satisfied. □

Lemma 2. *Given two different molecular graphs G_1 and G_2, they cannot be represented by the same normal tree.*

Proof. We can unambiguously obtain a molecular graph from a normal tree by replacing all benzene nodes with benzene rings according to its carbon position lists. □

Proposition 6. *For a normal tree T with a path $(v_1, \ldots, v_n)$, G' is the molecular graph obtained from the tree T' by removing $T^{sub}(v_1, v_2)$ and $T^{sub}(v_n, v_{n-1})$ except v_1 and v_n from T, where v_1 is left of v_n. If there is a non-identity map ϕ of the automorphism group of G' satisfying $\phi(v_i) = v_{n-i+1}$ for all $i = 1, \ldots, n$, then $T^{sub}(v_1, v_2) \geq_C T^{sub}(v_n, v_{n-1})$, where ϕ in G' is naturally extended to T.*

Proof. If $T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right) >_m T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)$, then $T^{sub}(v_1, v_2) >_m T^{sub}(v_n, v_{n-1})$, and $T^{sub}(v_1, v_2) >_C T^{sub}(v_n, v_{n-1})$. We assume $T^{sub}\left(v_{\lfloor \frac{n}{2} \rfloor}, v_{\lfloor \frac{n}{2} \rfloor + 1}\right) =_m T^{sub}\left(v_{n-\lfloor \frac{n}{2} \rfloor + 1}, v_{n-\lfloor \frac{n}{2} \rfloor}\right)$. If the path $(v_1, \ldots, v_n)$ is a symmetric path, $T^{sub}(v_1, v_2) \geq_C T^{sub}(v_n, v_{n-1})$ from condition 6. We assume that $(v_{i+1}, \ldots, v_{n-i})$ is a symmetric path for some i, and $index(v_i, T^{sub}(v_{i+1}, v_{i+2})) > index(v_{n-i+1}, T^{sub}(v_{n-i}, v_{n-i-1}))$ (see Fig. 11). Then,

$$\begin{aligned} &T^{sub}(v_{i+1}, v_{i+2}) =_m T^{sub}(v_{n-i}, v_{n-i-1}), \\ &T^{sub}(v_{i+1}, v_{i+2}) \geq_C T^{sub}(v_{n-i}, v_{n-i-1}). \end{aligned} \tag{1}$$

Let u_j and w_j be child nodes of v_{i+1} and v_{n-i}, respectively. Then, $v_i = u_{j_2}$ and $v_{n-i+1} = w_{j_1}$, where $j_1 = index(v_{n-i+1}, T^{sub}(v_{n-i}, v_{n-i-1}))$ and $j_2 = index(v_i, T^{sub}(v_{i+1}, v_{i+2}))$. If v_{i+1} and v_{n-i} are benzene nodes, $T(u_{j_1}) =_C T(v_i)$, $T(v_{n-i+1}) =_C T\left(w_{j_2}\right)$, and $T(v_i) =_C T(v_{n-i+1})$ because $C_{v_{i+1}}^T = C_{v_{n-i}}^T$ and $\phi(v_i) = v_{n-i+1}$.

We assume that v_{i+1} and v_{n-i} are not benzene nodes. For child nodes u_j of v_{i+1}, $T\left(u_j\right) \geq_C T\left(u_{j+1}\right)$ because $\left(u_j, v_{i+1}, u_{j+1}\right)$ is a symmetric path. Also for child nodes w_j of v_{n-i}, $T\left(w_j\right) \geq_C T\left(w_{j+1}\right)$. From the definition of ϕ, $T\left(u_j\right) =_C T\left(\phi(u_j)\right)$ for all $u_j \neq v_i$. If $index\left(\phi(u_{j+l})\right) < index\left(\phi(u_j)\right)$ for $u_j, u_{j+l} \neq v_i$ and $l > 0$, $T\left(u_j\right) \geq_C T\left(u_{j+l}\right) =_C T\left(\phi(u_{j+l})\right) \geq_C T\left(\phi(u_j)\right) =_C T(u_j)$. It means $T(u_j) =_C T\left(u_{j+l}\right)$. We assume

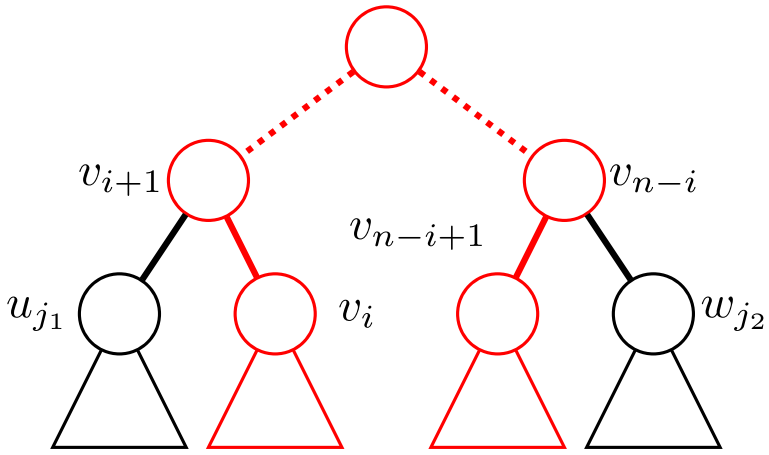

Fig. 11 Illustration of an automorphism ϕ in the proof. The *red path* indicates $(v_1, \ldots, v_n)$, where $\phi(v_i) = v_{n-i+1}$ for all $i = 1, \ldots, n$

that $index\left(\phi(u_j)\right) < index\left(\phi(u_{j+l})\right)$ for all $u_j \neq v_i$, that is, $\phi(u_j) = w_{j+1}$ for all $j = j_1, \ldots, j_2 - 1$. Then,

$$\begin{aligned} T\left(u_j\right) &=_C T\left(w_{j+1}\right) \leq_C T\left(w_j\right), \\ \text{and } T(v_i) &\leq_C T\left(u_{j_2-1}\right) =_C T\left(w_{j_2}\right). \end{aligned} \tag{2}$$

If $T^{sub}(v_{i+1}, v_{i+2}) >_C T^{sub}(v_{n-i}, v_{n-i-1})$, then there is an integer j ($j_1 \leq j \leq j_2$) such that $T(u_j) >_C T(w_j)$, and it contradicts Eq. (2). Therefore, $T^{sub}(v_{i+1}, v_{i+2}) =_C T^{sub}(v_{n-i}, v_{n-i-1})$, and $T(v_i) =_C T(v_{n-i+1})$. Also for the case that $(v_{i+1}, \ldots, v_{n-i})$ is a symmetric path for some i and $index(v_i, T^{sub}(v_{i+1}, v_{i+2})) < index(v_{n-i+1}, T^{sub}(v_{n-i}, v_{n-i-1}))$, then $T(v_i) =_C T(v_{n-i+1})$. Thus, $T^{sub}(v_1, v_2) \geq_C T^{sub}(v_n, v_{n-1})$. □

Lemma 3. *Given two different normal trees T_1 and T_2, T_1 does not represent the same molecular graph as T_2.*

Proof. We assume that T_1 represents the same molecular graph as T_2. Let G_1 and G_2 be molecular graphs transformed from T_1 and T_2, respectively, where each carbon in benzene rings and naphthalene rings is connected with adjacent atoms according to carbon position lists of T_1 and T_2. From the assumption, there is an isomorphism ψ from G_1 to G_2. It means that $l(v_1) = l(\psi(v_1))$ for all $v_1 \in V(G_1)$, $(\psi(v_1), \psi(v_2)) \in E(G_2)$ if and only if $(v_1, v_2) \in E(G_1)$, and $mul(\psi(v_1), \psi(v_2)) = mul(v_1, v_2)$.

Consider the case that the automorphism group $Aut(G_1)$ of G_1 has only elements ϕ such that $\phi(v_1) \neq v_2$ for v_1 and v_2 belonging to distinct benzene rings. Let $T(G)$ be the molecular tree without carbon position lists, obtained from G by contracting benzene rings and naphthalene rings to benzene nodes, and satisfying conditions 1, 2 of Definition 6. We suppose that maps ψ and ϕ in G_1 are naturally extended to $T(G_1)$. Since T_1 is different from T_2, there is a benzene node $v_1 \in V(T_1)$ such that

$$C_{v_1}^{T_1} \neq C_{\psi(v_1)}^{T_2}. \tag{3}$$

If v_1 is not connected by a merge bond with the parent node, there is a non-identity map $\phi_b \in Aut_b$ such that $C_{v_1}^{T_1} = \phi_b\left(C_{\psi(v_1)}^{T_2}\right)$ because T_1 and T_2 represent the same molecular graph. It contradicts condition 4 of Definition 6. Suppose that v_1 is connected by a merge bond with the parent node v_p and $C_{v_p}^{T_1} = C_{\psi(v_p)}^{T_2}$. If $T^{sub}\left(v_p, v_1\right) =_{\underline{C}} T^{sub}\left(v_1, v_p\right)$, then v_p is the root, and there is a non-identity map $\phi_n \in Aut_n$ such that $C_{(v_p,v_1)}^{T_1} = \phi_n\left(C_{(\psi(v_p),\psi(v_1))}^{T_2}\right)$ because T_1 and T_2 represent the same molecular graph. It contradicts condition 5a. Otherwise, $T^{sub}\left(v_p, v_1\right) \neq_{\underline{C}} T^{sub}\left(v_1, v_p\right)$. If v_p is not the root, then T_1 does not represent the same molecular graph as T_2 because $T^{sub}\left(v_a, v_p\right)$, where v_a is the parent of v_p, is different from other subtrees connected to the naphthalene ring. It contradicts the assumption. If v_p is the root, $C_{v_p}^{T_1} = \phi_{ref}\left(C_{v_p}^{T_1}\right)$ and $C_{v_1}^{T_1} = \phi_{ref}\left(C_{\psi(v_1)}^{T_2}\right)$ because T_1 and T_2 represent the same molecular graph. It contradicts condition 5b.

Consider the case that there is an element $\phi \in Aut(G_1)$ such that $\phi(v_1) = v_2$ for v_1 and v_2 belonging to distinct benzene rings. Since T_1 is different from T_2, there is a benzene node $v_1 \in V(T_1)$ such that

$$C_{v_1}^{T_1} \neq C_{\psi(v_1)}^{T_2}. \tag{4}$$

Here, we suppose that conditions 3, 4, 5 are satisfied for all benzene nodes in T_1 and T_2. Then, there is a path from v_1 to $\phi(v_1) = v_n$, $(v_1, \ldots, v_n)$, in T_1. Since T_1 and T_2 represent the same molecular graph,

$$\begin{aligned} &T_1^{sub}(v_1, v_2) =_C T_2^{sub}(\psi(v_n), \psi(v_{n-1})) \text{ and} \\ &T_1^{sub}(v_n, v_{n-1}) =_C T_2^{sub}(\psi(v_1), \psi(v_2)). \end{aligned} \tag{5}$$

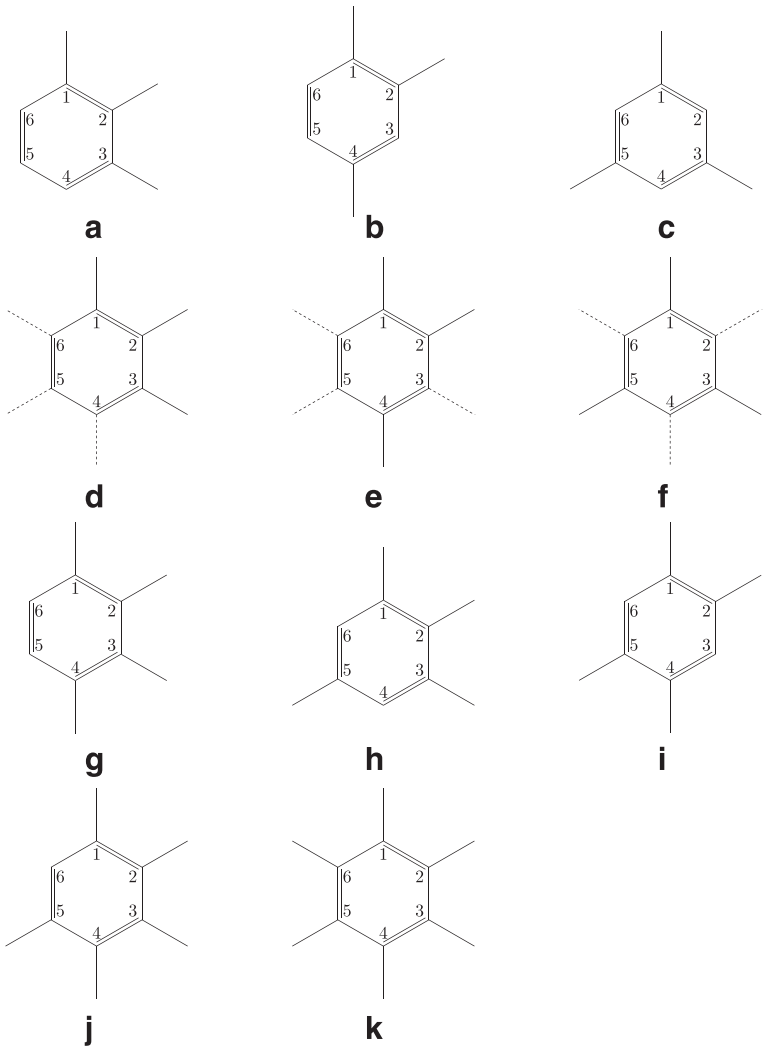

Fig. 12 Illustration of benzene rings having each carbon position list in Table 1. **a** ((1,2,3)). **b** ((1,2,4)). **c** ((1,3,5)). **d** ((1,2,3),(4,5,6)). **e** ((1,2,4),(3,5,6)). **f** ((1,3,5),(2,4,6)). **g** ((1,2,3,4)). **h** ((1,2,3,5)). **i** ((1,2,4,5)). **j** ((1,2,3,4,5)). **k** ((1,2,3,4,5,6)). *Solid* and *dashed lines* correspond to $A_v^T[1]$ and $A_v^T[2]$, respectively

Here, we can assume that v_1 is left of v_n and $\psi(v_1)$ is left of $\psi(v_n)$ without loss of generality. Then, from Proposition 6, for paths of $(v_1, \ldots, v_n)$ and $(\psi(v_1), \ldots, \psi(v_n))$,

$$T_1^{sub}(v_1, v_2) \geq_C T_1^{sub}(v_n, v_{n-1}) \text{ and} \quad (6)$$
$$T_2^{sub}(\psi(v_1), \psi(v_2)) \geq_C T_2^{sub}(\psi(v_n), \psi(v_{n-1}))$$

because T_1 and T_2 are normal trees. There is no carbon position lists that satisfy Eqs. (4), (6) and (7).

Therefore, T_1 does not represent the same molecular graph as T_2. □

Methods

We propose an algorithm BfsBenNaphEnum for enumerating chemical compounds containing benzene rings and naphthalene rings as cyclic structures. BfsBenNaphEnum utilizes our previously developed algorithms BfsSimEnum, BfsMulEnum [18], and assigns carbon position lists.

Modification of BfsSimEnum and BfsMulEnum

Suppose that the numbers n_{l_i} of atoms with label l_i for all $l_i \in \Sigma$, the numbers n_b, n_n of benzene rings and naphthalene rings are given. BfsBenNaphEnum introduces a special label 'b' representing a benzene node to Σ with $b > l_i \in \Sigma$ and $val(b) = 6$, and executes BfsSimEnum to generate all non-redundant molecular trees T such that $num(T, l_i) = n_{l_i}$ for $l_i \in \Sigma$ except $l_i = b, C$ and $num(T, b) = n_b + 2n_n$, $num(T, C) = n_C - 6n_b - 10n_n$. At this time, all edges of enumerated trees are single because BfsSimEnum generates only simple trees. Then, we modify BfsMulEnum to assign n_n merge bonds to edges between benzene nodes in each

tree enumerated by BfsSimEnum in addition to adding $1 + \sum_{l_i \in \Sigma, l_i \neq b} num(T, l_i)(val(l_i) - 2)/2$ bonds to edges between usual nodes. It should be noted that multiple bonds cannot be assigned to edges connected to benzene nodes since a carbon atom in benzene rings and naphthalene rings is connected with another adjacent atom by a single bond.

Assignment of carbon positions for molecular trees

In this algorithm, we traverse along the tree T from the rightmost deepest benzene node to the root in reverse BFS order because an adjacent node list depends on carbon position lists of descendant nodes. For each benzene node v we found, we assign a carbon position list not to violate the conditions of normal form.

The pseudocode of assignment part in BfsBenNaphEnum is given in Algorithms 1 and 2. We always assign carbon position 1 to the first node in A_v^T (line 20 in ASSIGN function) due to Proposition 5, which is the parent node of v if v is not the root (Proposition 2). If v is the root and $|A_v^T[1]| \geq 3$, we assign carbon position lists in Table 1 (see also Fig. 12) to v immediately for the sake of efficiency. Carbon position lists in Table 1 satisfy condition 4 of the normal form, and all the cases are included in the table.

For other carbon positions from 2 to 6, we use ASSIGN_CHILD to assign such positions to the remaining adjacent nodes. For example, let T_1 in Fig. 6 be output without any carbon position list by BfsMulEnum. T_1 has a benzene node v_1, and $A_{v_1}^{T_1} = ((v_4), (v_5), (v_2, v_3))$. First, carbon position 1 is assigned to $A_{v_1}^{T_1}[1][1] = v_4$, that is, $C_{v_1}^{T_1}[1][1] = 1$. Since v_1 is the root and $|A_{v_1}^{T_1}[1]| = 1 < 3$, Table 1 is not used, and the other nodes v_5, v_2, v_3 are assigned by ASSIGN_CHILD. For v_5, each carbon position from 2 to 6 is examined (line 26 in ASSIGN_CHILD). For v_2, each position from 2 to 6 except the position assigned to v_5 is examined (line 27). For v_3, each position from 2 to 6 that is more than the position assigned to v_2 except the position assigned to v_5 is examined (line 27) because v_2 and v_3 have the same subtree and condition 3 must be satisfied. Thus, $C_{v_1}^{T_1} = ((1), (2), (3, 4)), ((1), (2), (3, 5)), ((1), (2), (3, 6)), \ldots,$ ((1), (3), (2, 4)), ((1), (3), (2, 5)), ((1), (3), (2, 6)), . . . , ((1), (6), (4, 5)) are examined, where ((1), (6), (2, 3)), ((1), (6), (2, 4)), ((1), (5), (2, 3)) and so on are discarded in the next step.

Table 1 Carbon position lists for A_v^T, where v is the root, and $|A_v^T[1]| \geq 3$

$\lvert A_v^T[1] \rvert$	$\lvert A_v^T[2] \rvert$	C_v^T
3	0	((1,2,3)), ((1,2,4)), ((1,3,5))
3	3	((1,2,3),(4,5,6)), ((1,2,4),(3,5,6)), ((1,3,5),(2,4,6))
4	0	((1,2,3,4)), ((1,2,3,5)), ((1,2,4,5))
5	0	((1,2,3,4,5))
6	0	((1,2,3,4,5,6))

Algorithm 1 Assignment algorithm of carbon positions for a molecular tree T

```
1: function ASSIGN_CARBON_POSITIONS(T)
2:     v := the last benzene node of T in BFS order
3:     ASSIGN(T, v)
4: end function
1: function ASSIGN(T, v)
2:     if v is null then
3:         P := the set of all pairs of nodes (v_1, ..., v_n) such that v_1 is left of v_n, the path from v_1 to v_n is a symmetric path, and v_1 and v_n are not connected by a merge bond
4:         if T^sub(v_1, v_2) ≥_C T^sub(v_n, v_{n-1}) for all (v_1, v_n) ∈ P then
5:             output T
6:         return
7:     if the next benzene node of v in reverse BFS order exists then
8:         v' := the next benzene node of v in reverse BFS order
9:     else
10:        v' := null
11:    if |A_v^T| = 0 then
12:        ASSIGN(T, v')
13:        return
14:    if v is the root of T then
15:        if |A_v^T[1]| ≥ 3 then
16:            for each valid carbon position list p in Table 1 do
17:                C_v^T := p
18:                ASSIGN(T, v')
19:            return
20:    C_v^T[1][1] := 1
21:    if (v, A_v^T[1][1]) is a merge bond then
22:        C_v^T[1][2] := 2
23:    ASSIGN_CHILD(T, v, A_v^T[1][1], v')
24: end function
```

For each benzene node v, after assignment of a carbon position list to A_v^T, whether or not C_v^T violates conditions 4, 5 of the normal form is confirmed (lines 5, 11, 14 in ASSIGN_CHILD). After carbon position lists are assigned to all benzene nodes, condition 6 is confirmed (line 4 in ASSIGN).

Since an input of this part, that is, an output of BfsMulEnum, satisfies conditions 1, 2 of the normal form, BfsBenNaphEnum always outputs normal trees. In ASSIGN_CHILD, a distinct carbon position list is always

Algorithm 2 Assignment algorithm for adjacent nodes of a benzene node v

1: **function** ASSIGN_CHILD(T, v, w, v')
2: **if** w is *null* **then**
3: $flag := true$
4: **if** v is not connected by a merge bond with the parent node **then**
5: **if** $\phi_b \in Aut_b$ such that $C_v^T > \phi_b(C_v^T)$ exists **then**
6: $flag := false$
7: **if** v is the root of T **then**
8: **if** a benzene node connected by a merge bond with v exists **then**
9: $v_c :=$ the benzene node connected by a merge bond with v
10: **if** $T^{sub}(v, v_c) =_{\underline{C}} T^{sub}(v_c, v)$ **then**
11: **if** $\phi_n \in Aut_n$ such that $C^T_{(v,v_c)} > \phi_n\left(C^T_{(v,v_c)}\right)$ exists **then**
12: $flag := false$
13: **else**
14: **if** $C_v^T = \phi_{ref}(C_v^T)$ and $C_{v_c}^T > \phi_{ref}\left(C_{v_c}^T\right)$ **then**
15: $flag := false$
16: **if** $flag$ **then**
17: ASSIGN(T, v')
18: **return**
19: **if** the next node of w in A_v^T exists **then**
20: $w' :=$ the next node of w in A_v^T
21: **else**
22: $w' := null$
23: **if** w has been already assigned **then**
24: ASSIGN_CHILD(T, v, w', v')
25: **return**
26: **for** $p = 2, \ldots, 6$ **do**
27: **if** p has not been assigned and $p > \max_{j'<j} C_v^T[i][j']$ for $w = A_v^T[i][j]$ **then**
28: $C_v^T[i][j] := p$
29: **if** $(v, A_v^T[i][j])$ is a merge bond **then**
30: $C_v^T[i][j+1] := p+1$
31: ASSIGN_CHILD(T, v, w', v')
32: **end function**

assigned, and all patterns are assigned (line 28). Hence, BfsBenNaphEnum outputs all distinct normal trees.

Theorem 1. *BfsBenNaphEnum outputs all non-redundant molecular graphs that are solutions of Problem 1.*

Figure 13 shows another example T_7 of molecular trees. T_7 includes four benzene nodes v_5, v_4, v_3, v_2 in reverse BFS order, and edges (v_2, v_4), (v_3, v_5) are merge bonds. First, our algorithm assigns carbon position lists for $A^{T_7}_{v_5} = ((v_3), (v_7))$ as $C^{T_7}_{v_5} = ((1,2),(3)), ((1,2),(4)), ((1,2),(5)), ((1,2),(6))$. In a similar way, for $A^{T_7}_{v_4} = ((v_2), (v_6))$, $C^{T_7}_{v_4} = ((1,2),(3)), ((1,2),(4)), ((1,2),(5)), ((1,2),(6))$. For $A^{T_7}_{v_3} = ((v_1), (v_5))$, $C^{T_7}_{v_3} = ((1),(2,3)), ((1),(3,4)), ((1),(4,5)), ((1),(5,6))$ are examined. In line 5 of ASSIGN_CHILD, $((1),(4,5))$ and $((1),(5,6))$ are discarded because $\phi_b(((1),(4,5))) = ((1),(3,4))$, $\phi_b(((1),(5,6))) = ((1),(2,3))$ for the reflection map ϕ_b with respect to the axis through positions 1 and 4, and these violate condition 4. In a similar way, for $A^{T_7}_{v_2} = ((v_1), (v_4))$, $C^{T_7}_{v_2} = ((1),(2,3)), ((1),(3,4))$ are assigned. After carbon position lists are assigned to all benzene nodes, condition 6 is confirmed in line 4 of ASSIGN. If $C^{T_7}_{v_2} \neq C^{T_7}_{v_3}$, then there is one symmetric path, $\mathcal{P} = \{(v_2, v_3)\}$, and $T_7(v_2) \geq_C T_7(v_3)$ must be satisfied. It means that $C^{T_7}_{v_4} = C^{T_7}_{v_5} = ((1,2),(3)), ((1,2),(4)), ((1,2),(5)), ((1,2),(6))$ and $C^{T_7}_{v_2} = ((1),(3,4)) > C^{T_7}_{v_3} = ((1),(2,3))$, or $C^{T_7}_{v_4} > C^{T_7}_{v_5}$ and $C^{T_7}_{v_2} \neq C^{T_7}_{v_3}$. Hence, there are $4 + \binom{4}{2} \cdot 2 = 16$ structures. If $C^{T_7}_{v_2} = C^{T_7}_{v_3} = ((1),(2,3))$ (or $C^{T_7}_{v_2} = C^{T_7}_{v_3} = ((1),(3,4))$), then $\mathcal{P} = \{(v_2, v_3), (v_4, v_5)\}$, and both of $T_7(v_2) \geq_C T_7(v_3)$ and $T_7(v_4) \geq_C T_7(v_5)$, that is, $C^{T_7}_{v_4} \geq C^{T_7}_{v_5}$, must be satisfied. Hence, there are $4 + 3 + 2 + 1 = 10$ structures. In total, $16 + 10 \cdot 2 = 36$ structures are generated by BfsBenNaphEnum for T_7.

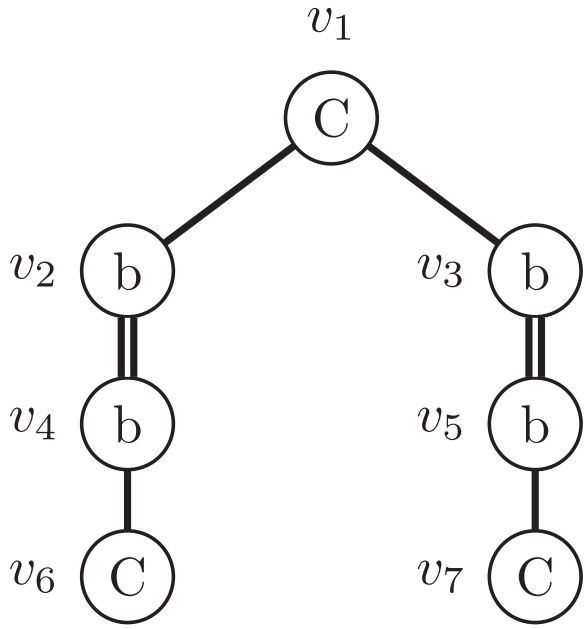

Fig. 13 Example of a molecular tree T_7

Results

In this section, we show that our proposed method can enumerate chemical compounds with benzene rings and

naphthalene rings correctly and efficiently. For the evaluation, although MOLGEN 3.5 is more suitable than MOLGEN 5.0 to enumerate tree-like compounds because MOLGEN 3.5 offered the possibility to define substructures like benzene or naphthalene as macro atoms but MOLGEN 3.5 cannot handle all the cases provided in Table 2, we compared proposed tool with MOLGEN 5.0. Thereby, we implemented it and installed another well-known general purpose structure generator, MOLGEN 5.0, on a computer with 3.47 GHz intel Xeon CPU and 23.5 GiB memory, and compared their computational time. The implementation of BfsBenNaphEnum is available on our supplementary web site, http://sunflower.kuicr.kyoto-u.ac.jp/jira/bfsenum/.

Since MOLGEN can enumerate chemical compounds without restriction on the structure, we must specify a benzene ring and a naphthalene ring as a substructure so that the enumerated structures contain only benzene rings and naphthalene rings as cyclic structures. As can be seen from Table 2, where 'n' and 'b' denote a naphthalene ring and a benzene ring, respectively, BfsBenNaphEnum enumerated chemical compounds much faster than MOLGEN while giving the same number of enumerated structures. BfsBenNaphEnum was from 50 times to 5,000,000 times faster than MOLGEN for instances with 8 to 14 carbon atoms. Table 2 also compares the number of discovered compounds in PubChem, which are not limited to tree-like chemical compounds, with the number of compounds enumerated by the proposed algorithm for several chemical formulas. When the number of carbon atoms is large (greater than 8 in this case), the number of discovered compounds is much less than the number of enumerated compounds. This implies that there are still a numerous number of unknown compounds to be discovered, which possibly include some essential compounds. In this study, we examined chemical formulas including up to two benzene rings and one naphthalene ring because MOLGEN was not able to output results in practical time for chemical formulas including more benzene rings and naphthalene rings.

We plotted the relation between the number of enumerated structures and the computational time for both methods in Fig. 14, where both x-axis and y-axis are in a log scale. It is seen from the figure that the execution time of BfsBenNaphEnum is much smaller than that of MOLGEN.

Discussion

Our algorithm is limited to tree-like chemical structures without any cyclic structures except benzene rings and naphthalene rings while MOLGEN does not have such limitation. Therefore, in the future, we would like to extend the algorithm such that it can enumerate more complex cyclic structures, such as polycyclic aromatic compounds and nucleotides. Besides, in order to make enumeration tools practical, we need to rank enumerated structures because a large number of structures are usually enumerated. For that purpose, it might be useful to

Table 2 Results on execution time (sec), the number of enumerated structures by BfsBenNaphEnum and MOLGEN, and the number of chemical compounds exist in PubChem database for several instances

Chemical formula	#atoms						#all compounds in PubChem	#enumerated structures	Computational time (sec)	
	n	b	C	N	O	H			BfsBenNaphEnum	MOLGEN
$C_7O_2H_8$	0	1	1	0	2	8	728	19	0.001	0.053
$C_8O_3H_{10}$	0	1	2	0	3	10	1602	307	0.002	0.124
$C_9O_4H_{10}$	0	1	3	0	4	10	1469	6406	0.010	1.699
$C_{10}N_2O_4H_{10}$	0	1	4	2	4	10	1592	8,333,991	12.260	957.53
	1	0	0	2	4	10		7980	0.031	69.51
$C_{11}N_2H_{10}$	0	1	5	2	0	10	790	9012	0.021	630.44
	1	0	1	2	0	10		56	0.005	24.061
$C_{12}N_1O_1H_{11}$	0	1	6	1	1	11		80,883	0.155	2,611.57
	0	2	0	1	1	11	1582	33	0.001	98.99
	1	0	2	1	1	11		888	0.009	560.98
$C_{13}O_2H_{12}$	0	1	7	0	2	12		162,122	0.289	6,497.55
	0	2	1	0	2	12	1239	190	0.002	2,069.3
	1	0	3	0	2	12		2458	0.013	1,731.92
$C_{14}O_4H_{12}$	0	1	8	0	4	12		19,514,480	35.655	197,264.54
	0	2	2	0	4	12	1 397	15,581	0.021	107,509.42
	1	0	4	0	4	12		337,178	1.061	97,326.71

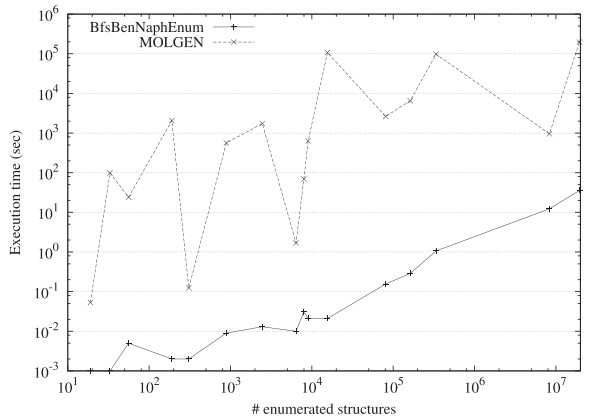

Fig. 14 Relation between the number of enumerated structures and the computational time (sec)

employ drug likeness filters such as Lipinski RO5, and QED score. Incorporation of such filters into our system is also important future work.

Conclusions

We proposed a way to represent a benzene ring in a molecular tree by regarding it as a new defined atom with valence six and introducing a new attribute named carbon position list to benzene nodes. Carbon position of an atom specifies which carbon in a benzene ring that the corresponding atom bonds with. We also proposed a new kind of bond called *merge bond* that merges two benzene rings together to form a naphthalene ring. With merge bond a molecular tree can represent a structure containing naphthalene rings without defining new kind of atom. Moreover, since a benzene ring and a naphthalene ring are symmetric structures, we defined a rule to assign carbon position lists such that no redundant structures due to the symmetry of a benzene ring and a naphthalene ring are enumerated.

The algorithm of this work consists of two main steps. Given the number of benzene rings, the number of naphthalene rings as well as a chemical formula, BfsSimEnum and BfsMulEnum are applied such that they can enumerate molecular trees with benzene nodes. Next, the new extension *BfsBenNaphEnum* assigns carbon position lists to benzene nodes in normal molecular trees.

To show the performance of our algorithm, all non-redundant chemical structures were enumerated for several chemical formulas by BfsBenNaphEnum and MOLGEN 5.0, a well-known general purpose structure generator. It is shown that our algorithm is reliable since it generated the same number of structures as MOLGEN, while expended much less computational time. BfsBenNaphEnum was from 50 times to 5,000,000 times faster than MOLGEN for instances with 8 to 14 carbon atoms in our experiments. This is mainly because the number of nodes decreases from six to one for each benzene ring and from ten to two for each naphthalene ring in a chemical structure and because we enumerate chemical structures in the form of trees instead of graphs.

Competing interests
The authors declare that they have no competing interests.

Authors' contributions
JJ and MH developed, implemented the methods, and drafted the manuscript. YZ and TA participated in the discussions during the development of the methods and helped draft the manuscript. All authors read and approved the final manuscript.

Acknowledgements
This work was partially supported by Grants-in-Aid #26240034, #24500361, and #25-2920 from MEXT, Japan.

Received: 27 April 2015 Accepted: 19 February 2016
Published online: 01 March 2016

References

1. Ward RA, Kettle JG. Systematic enumeration of heteroaromatic ring systems as reagents for use in medicinal chemistry. J Med Chem. 2011;54(13):4670–7.
2. Blum LC, Reymond JL. 970 million druglike small molecules for virtual screening in the chemical universe database gdb-13. J Am Chem Soc. 2009;131(25):8732–3.
3. Mishima K, Kaneko H, Funatsu K. Development of a new de novo design algorithm for exploring chemical space. Mol Inform. 2014;33(11-12):779–89.
4. Funatsu K, Sasaki S. Recent advances in the automated structure elucidation system, chemics. utilization of two-dimensional NMR spectral information and development of peripheral functions for examination of candidates. J Chem Inform Comput Sci. 1996;36(2):190–204.
5. Meringer M, Schymanski EL. Small molecule identification with MOLGEN and mass spectrometry. Metabolites. 2013;3:440–62.
6. Koichi S, Arisaka M, Koshino H, Aoki A, Iwata S, Uno T, Satoh H. Chemical structure elucidation from ^{13}C NMR chemical shifts: Efficient data processing using bipartite matching and maximal clique algorithms. J Chem Inform Model. 2014;54:1027–35.
7. Bytautas L, Klein DJ, Schmalz TG. All acyclic hydrocarbons: Formula periodic table and property overlap plots via chemical combinatorics. New J Chem. 2000;24(5):329–36.
8. Faulon J, Visco DP, Roe D. Enumerating molecules. Rev Comput Chem. 2005;21:209.
9. Koch MA, Schuffenhauer A, Scheck M, Wetzel S, Casaulta M, Odermatt A, Ertl P, Waldmann H. Charting biologically relevant chemical space: A structural classification of natural products (sconp). Proc Natl Acad Sci U S A. 2005;102(48):17272–7.
10. Mauser H, Stahl M. Chemical fragment spaces for de novo design. J Chem Inf Model. 2007;47(2):318–24.
11. Andricopulo AD, Guido RV, Oliva G. Virtual screening and its integration with modern drug design technologies. Curr Med Chem. 2008;15(1):37–46.
12. Reymond JL, van Deursen R, Blum LC, Ruddigkeit L. Chemical space as a source for new drugs. MedChemComm. 2010;1(1):30–8.
13. Bürgi JJ, Awale M, Boss SD, Schaer T, Marger F, Viveros-Paredes JM, Bertrand S, Gertsch J, Bertrand D, Reymond JL. Discovery of potent positive allosteric modulators of the $\alpha 3\beta 2$ nicotinic acetylcholine receptor by a chemical space walk in chembl. ACS Chem Neurosci. 2014;5(5):346–59.
14. Gugisch R, Kerber A, Kohnert A, Laue R, Meringer M, Rücker C, Wassermann A. MOLGEN 5.0, a molecular structure generator. Sharjah, United Arab Emirates: Bentham Science Publishers Ltd.; 2012.
15. Peironcely JE, Rojas-Chertó M, Fichera D, Reijmers T, Coulier L, Faulon JL, Hankemeier T. OMG: Open Molecule Generator. J Cheminformatics. 2012;4:21.

16. Fujiwara H, Wang J, Zhao L, Nagamochi H, Akutsu T. Enumerating treelike chemical graphs with given path frequency. J Chem Inf Model. 2008;48(7):1345–57.
17. Shimizu M, Nagamochi H, Akutsu T. Enumerating tree-like chemical graphs with given upper and lower bounds on path frequencies. BMC Bioinformatics. 2011;12:14–3.
18. Zhao Y, Hayashida M, Jindalertudomdee J, Akutsu T. Breadth-first search approach to enumeration of tree-like chemical compounds. J Bioinformatics Comput Biol. 2013;11:1343007.
19. Schüller A, Hähnke V, Schneider G. SmiLib v2.0: A Java-based tool for rapid combinatorial library enumeration. QSAR Comb Sci. 2007;26(3):407–10.
20. Song CM, Bernardo PH, Chai CLL, Tong JC. CLEVER: Pipeline for designing in silico chemical libraries. J Mol Graph Model. 2009;27(5):578–83.
21. Trinajstić N. Chemical Graph Theory, 2nd edn. Boca Raton, Florida: CRC Press; 1992, pp. 275–391. Chap. 11 Isomer Enumeration.
22. Meringer M. Handbook of Chemoinformatics Algorithms. Boca Raton, Florida: CRC Press; 2010, pp. 233–67. Chap. 8 Structure Enumeration and Sampling.
23. Suzuki M, Nagamochi H, Akutsu T. Efficient enumeration of monocyclic chemical graphs with given path frequencies. J Cheminformatics. 2014;6:31.
24. Hardinger SA, University of California LADoC. Biochemistry: Chemistry 14D: Organic Reactions and Pharmaceuticals : Course Thinkbook, Lecture Supplements, Concept Focus Questions, OWLS Problems, Practice Problems. Plymouth, MI 48170: Hayden-McNeil Pub; 2008.