ACS POLYMERS Au
AN OPEN ACCESS JOURNAL OF THE AMERICAN CHEMICAL SOCIETY

pubs.acs.org/polymerau

Article

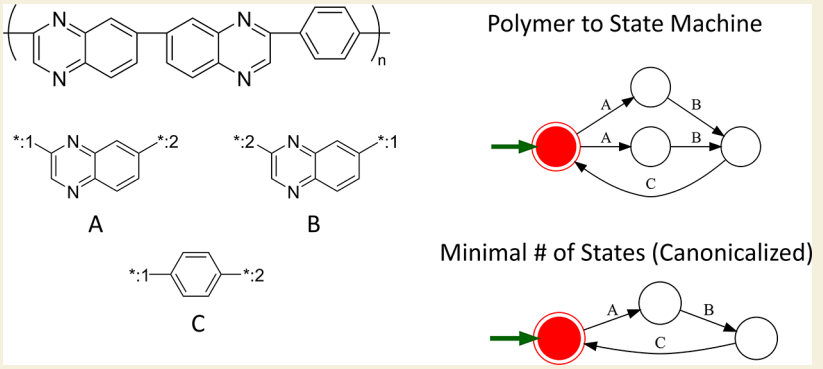
Canonicalizing BigSMILES for Polymers with Defined Backbones

Tzyy-Shyang Lin, Nathan J. Rebello, Guang-He Lee, Melody A. Morris, and Bradley D. Olsen*

Cite This: *ACS Polym. Au* 2022, 2, 486–500

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information



ABSTRACT: BigSMILES, a line notation for encapsulating the molecular structure of stochastic molecules such as polymers, was recently proposed as a compact and readable solution for writing macromolecules. While BigSMILES strings serve as useful identifiers for reconstructing the molecular connectivity for polymers, in general, BigSMILES allows the same polymer to be codified into multiple equally valid representations. Having a canonicalization scheme that eliminates the multiplicity would be very useful in reducing time-intensive tasks like structural comparison and molecular search into simple string-matching tasks. Motivated by this, in this work, two strategies for deriving canonical representations for linear polymers are proposed. In the first approach, a canonicalization scheme is proposed to standardize the expression of BigSMILES stochastic objects, thereby standardizing the expression of overall BigSMILES strings. In the second approach, an analogy between formal language theory and the molecular ensemble of polymer molecules is drawn. Linear polymers can be converted into regular languages, and the minimal deterministic finite automaton uniquely associated with each prescribed language is used as the basis for constructing the unique text identifier associated with each distinct polymer. Overall, this work presents algorithms to convert linear polymers into unique structure-based text identifiers. The derived identifiers can be readily applied in chemical information systems for polymers and other polymer informatics applications.

KEYWORDS: *automata, regular languages, digital search, macromolecules, cheminformatics*

1. INTRODUCTION

Line notations[1−4] that encode the molecular structures of chemical entities into text-based representations, such as the simplified molecular-input line-entry system (SMILES),[5,6] play instrumental roles in modern chemical informatics.[7] Like chemical nomenclature systems, line notations translate molecular connectivity information into text strings. As text format is widely supported in most digital systems, this feature enables the mapping of molecules into text identifiers, which can be easily stored in digital databases and forms without the need for additional graph manipulation packages.[8−11] Since identifiers derived from line notations are direct encodings of chemical graphs with atoms as vertices and bonds as edges of the associated molecules, the procedures involved in converting them into the original molecular graphs are often more straightforward than converting their counterparts encoded using schemes such as standard nomenclature,[12] product registry labelling,[13] or conventional names. Decoding the latter schemes often requires the use of complex name-resolving algorithms[14] or brute-force table look-ups.[11,15,16] Owing to these desirable features, line notations have been extensively deployed in chemical informatics and chemical modeling applications.[17−21]

Apart from enabling the encoding of chemical entities into formats that are compatible with digital systems, line notations also provide a means to simplify many essential operations in chemical information systems, such as comparing if two chemical entities are identical or retrieving records within a database that contains a specific molecule. For instance, with the

Received: March 25, 2022
Revised: August 16, 2022
Accepted: August 17, 2022
Published: October 14, 2022

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486–500

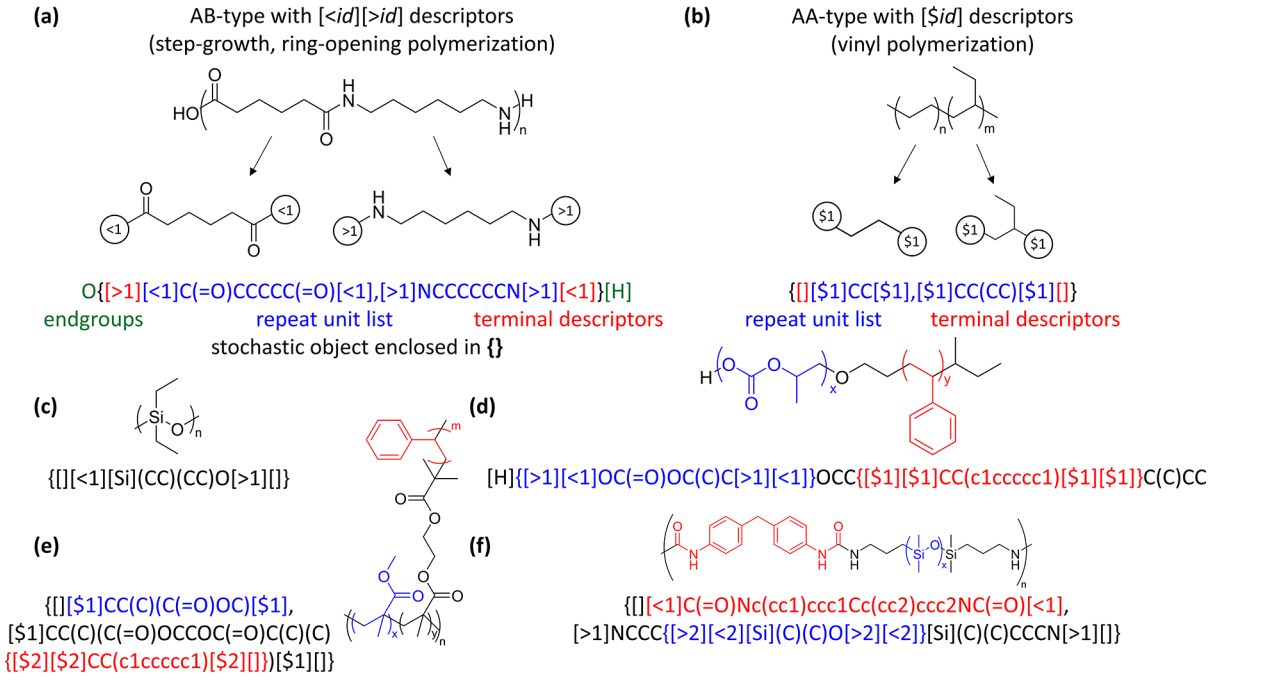


Figure 1. Illustration of the syntax and usage of BigSMILES stochastic objects and bonding descriptors. Panels (a)−(f) show examples of BigSMILES for polymer topologies with defined backbones for a condensation homopolymer, vinyl statistical copolymer, ring opening homopolymer, block copolymer, graft copolymer, and segmented copolymer, respectively. Stochastic objects, given by strings enclosed within a pair of curly brackets, denote a polymeric segment. Within a stochastic object, repeat units are denoted by a comma-delimited list. For each repeat unit in a linear polymer, two bonding descriptors are tethered to atoms through which the repeat units interconnect. The AA-type descriptor, denoted by $, represents sites that can be joined with any other $-descriptor with the same numeric index (labeled *id* in the figure). For example, vinyl polymerizations allow both head-to-tail and head-to-head connections. In contrast, AB-type descriptors < and > represent sites that can only be joined with the conjugate descriptors of the same numeric index. The terminal descriptors are additional descriptors immediately inside the curly brackets that indicate the rules for connectivity to endgroups. If empty terminal descriptors "[]" are written as in the right panel, this means that no endgroup is specified.

molecules transformed into text identifiers, the task of comparing if two given molecules are identical is reduced from a graph isomorphism problem in the original space of molecular graphs into the less computationally complex string comparison problem between two string identifiers.[6,22] Similarly, with the use of string identifiers, the searching of molecules within chemical documents becomes a string-matching task, which can be carried out very efficiently. However, the premise of these complexity reductions is that the adopted line notation provides a single unique representation for each chemically distinct substance. While line notations often allow multiple valid representations for a single molecule, the mapping can be made unique by imposing additional canonicalization rules to break the degeneracy between equally valid representations for a molecule, thereby assigning preference to a single canonical representation. For example, while the popular line notation SMILES provides a one-to-many map between a molecule and its string representations, the mapping can be made bijective by the introduction of additional tie-breaking rules, and the resulting Canonical SMILES provides a one-to-one function that can fulfill the uniqueness criterion.[6,7]

Recently, SMILES has been extended to support molecules with non-deterministic molecular connectivity, specifically polymers. The new line notation, BigSMILES,[22,23] specifies a series of standard syntactic rules to transform random graphs composed of sub-graph building blocks into text strings. A molecular graph is a labeled graph that can be used to represent a structural formula for a single molecule in which the atoms are nodes and the bonds are edges. A random graph is a probability distribution over a set of molecular graphs and can be used to represent polymers. BigSMILES provides compact encodings for stochastic molecules (random graphs) and their constituent units, such as polymers and the repeat units that make up the polymer. Like SMILES, BigSMILES requires additional canonicalization rules to provide a unique molecule-to-text mapping. Unlike canonicalization schemes for molecules with well-defined molecular connectivity, in which the problem of finding a canonical representation is equivalent to the identification of a preferred starting atom and a favored order to traverse the rest of the molecular graph,[7] canonicalizing the representation of a stochastic molecule is much more complicated because it involves coming up with a unique representation for random graphs. As such, out-of-the-box canonicalization schemes for SMILES only partially resolve the issue, and additional strategies must be developed to fully canonicalize BigSMILES.

In this article, canonicalization schemes for BigSMILES line notation are developed. Section 2 provides a brief overview of the BigSMILES language and the challenge of multiple representations for identical polymers. In the first strategy (Section 3), a recipe for generating the canonical forms of individual BigSMILES stochastic objects is discussed that involves the user writing a canonical set of repeat units that aligns with the monomers used to synthesize the polymer. In the second strategy (Section 4), a more general procedure for canonicalizing linear polymers based on an analogy with regular language theory is proposed that does not depend on the choice of repeat units but is less easily interpretable. We summarize these approaches and their applications in Section 5. Major efforts of the current work are focused on standardizing the representations for polymers with architectures with defined

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

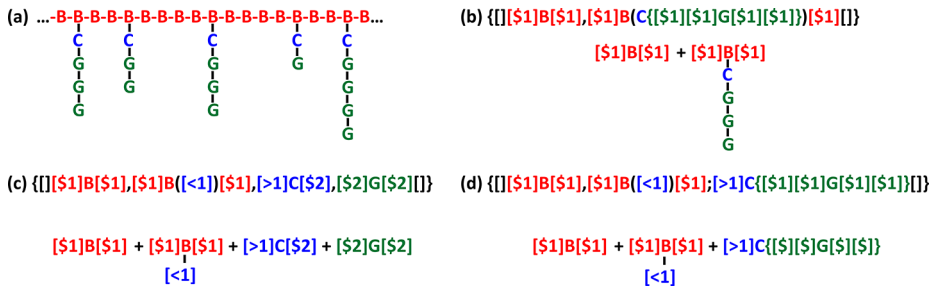


Figure 2. Illustration of multiple ways of converting a branched polymer into a BigSMILES string. (a) Sample polymer with its backbone units colored in red (-B-) partially grafted with linear grafts colored in blue and green (-C-G-G-...). (b) BigSMILES for the polymer with the stochastic object consisting of the un-grafted and grafted repeat units. (c) BigSMILES string for the same polymer with grafted sites represented as a trifunctional unit and the repeat unit for the graft specified in parallel to the backbone units. (d) BigSMILES for the polymer with the graft presented as endgroup. Since only bonding descriptors between the backbone units are allowed in the canonical choice of repeat units, the case presented in part (b) is the canonical form.

backbones, e.g., linear polymers or grafted polymers with well-defined backbones, due to their relative mathematical simplicity. Overall, this work presents an analysis on the necessary components in making a random graph representation unique and proposes two modes of implementations toward the canonicalization of the BigSMILES line notation.

2. OVERVIEW OF BIGSMILES LANGUAGE

Polymers are stochastic molecules that generally do not have well-defined molecular connectivity. Rather, a single polymer corresponds to an ensemble of molecular states, where each molecular state denotes a molecule with distinct chemical connectivity. To completely specify a polymer, the corresponding representation must be able to encode the entire ensemble associated with the given polymer. Since these ensembles are often infinite sets, instead of explicitly accounting for individual molecular states, an ensemble is generally described by enumerating the constituent building blocks of the polymer, given as chemical sub-graphs that reflect the structures of repeat units, and specifying a series of production rules that details how the building blocks can be joined to construct distinct molecular states. As such, line notation for polymers represents random graphs. More precisely, polymer line notation encapsulates the algorithm to draw realizations from the given random graph, where each realization corresponds to a molecular state.

The BigSMILES line notation[22,23] encodes random graphs by introducing stochastic objects into the popular SMILES line notation. A BigSMILES stochastic object represents a polymeric segment that is embedded within an otherwise non-polymeric SMILES context. Within a stochastic object, a list of repeat units is delineated by their SMILES strings. To specify how the given repeat units can be connected to form an individual polymer molecule, each repeat unit is assigned multiple BigSMILES bonding descriptors indicating the sites of connection. The bonding descriptors are tethered to the boundary atoms through which a repeat unit is connected to another repeat unit. The permissible connectivity patterns between individual repeat units are indicated by the type of bonding descriptor, as illustrated in Figure 1. BigSMILES stochastic objects provide a simple way of encapsulating the structures of a variety of polymeric segments formed via different chemistry. Similar to connectivity patterns between repeat units, the connections to endgroups are delineated by the terminal bonding descriptors found right within the curly brackets.

Overall, a BigSMILES string represents the set of molecules, excluding molecular fragments with open and unused bonding descriptors, generated by joining the repeat units and endgroups through connecting appropriate bonding descriptors. However, it not does not describe the probability of each molecule in the set. To quantify the distribution, the BigSMILES string must be linked to characterization information using data schemas such as PolyDAT published by the authors of this work.[24]

Finding a unique representation for a BigSMILES is a difficult problem, even for simple linear chains. This is because there are many ways to encode identical polymer ensembles, even if the SMILES string for each individual element (monomer, endgroup) is already canonicalized. For example, multiple ways of representing repeat units are often used in polymers such as nylon-6,6:

[<1]C(=O)CCCCC(=O)NCCCCCCN[>1]

[<1]C(=O)CCCCC(=O)[<1],[>1]NCCCCCCN[>1]

Problems such as this become more challenging with complex architectures such as graft polymers (Figure 2). Another complex canonicalization issue is frame shifting along the polymer backbone between monomers and endgroups (SI Section 5), and there are also simple degeneracies in the choice of bonding descriptor and the ordering of monomers. Canonicalization is therefore necessary to produce a standard representation, enabling faster processing of BigSMILES strings in algorithms such as polymer search.

3. CANONICALIZING BIGSMILES USING PRIORITY RULES

In general, the problem of deriving a canonical BigSMILES representation involves two major steps. First, a preferred set of repeat units is selected for each polymeric segment. Upon adjusting the endgroups according to the selection, this step would provide a unique atom partition for the exact stochastic object or endgroup each atom is assigned to, thereby allowing the overall polymer BigSMILES string to be uniquely defined up to the rearrangement of atom sequences within the SMILES-like string. This selection process is discussed in more detail in the following subsection. Notably, in some cases, polymeric segments may be embedded within a repeat unit or an endgroup, leading to nesting of stochastic objects reflecting hierarchy in the chemical structure. In many cases, such nesting can be written in multiple manners that are equally valid, yielding multiplicity in the string representation. For instance, as illustrated in Figure 2, a polymer graft can be written with the grafts nested within the repeat units, with the grafts as

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

independent polymeric units, or even with the grafts as polymeric endgroups. In principle, the repeat unit selection process serves as a tiebreaker that assigns preference to how the nesting should be written, leading to a unique representation. Then, upon defining a preferred set of repeat units, a BigSMILES string can be canonicalized by finding unique expressions for individual stochastic objects and applying SMILES canonicalization algorithm to the overall SMILES-like string containing the canonicalized stochastic objects. For nested BigSMILES strings, the second step would be iteratively performed, starting from the innermost layer, until the entire string is processed.

In this work, focus will be placed on the canonicalization of BigSMILES strings for polymers with well-defined backbones. Here, a polymer backbone is defined as a series of repeat units each having exactly two neighboring backbone repeat units or with one of the two neighbors being an endgroup. In the following sections, canonicalization will be defined over the context of a specific backbone, and the described procedures will only generate canonical BigSMILES strings that are unique over the specified backbone. For linear polymers, there is only a single choice for the backbone, and these procedures will lead to unique strings; however, for branched polymers, the generated canonical representations will be functions of the selected backbones. Here, no restriction is imposed on how the backbone should be defined, and the user is allowed to identify the backbone as appropriate. The only guideline to the delineation of the backbone is that the choice should lead to a backbone that is chemically sensible and relevant to the context in which it is used.

In general, a canonical BigSMILES representation is derived through the following procedure:

1. Selecting the Repeat Units (Section 3.1): A set of canonical repeat units is selected for each polymeric segment. Upon selection, adjust and obtain the corresponding sets of endgroups and rewrite the BigSMILES stochastic objects.
2. Canonicalizing Individual Endgroup Configurations (Section 3.2): For each endgroup configuration, derive the canonical BigSMILES string for the polymer associated with the given pair of endgroups by canonicalizing each stochastic object found within the BigSMILES string. The canonical representation for each stochastic object is derived by further reducing the canonical set of repeat units according to the specified connectivity to the endgroups, canonicalizing each repeat unit (3.2.1), reordering the repeat units (3.2.2), relabeling the bonding descriptors (3.2.3), and canonicalizing the endgroup configurations (3.2.4).
3. Canonicalizing the Overall BigSMILES String (Section 3.3): Once the BigSMILES representation for each end group configuration has been derived, reorder the obtained BigSMILES strings and join them into a dot-delimited string. The joined string is the canonical BigSMILES representation for the overall polymer.

3.1. Selecting the Repeat Units

Since only polymers with defined backbones are considered, the canonical choice of repeat units will be limited to those which contain only bonding descriptors that specify the connectivity patterns along the backbone. Any choice of the set that includes a repeat unit containing non-backbone bonding descriptors is disallowed. Because the backbone is linear, each repeat unit will possess exactly two bonding descriptors. (Similar logic applies to ladder polymers, but instead of having two bonding descriptors, each repeat unit will have two groups of bonding descriptors.) For non-linear polymers, pendant structures along the backbone must be embedded within repeat units as SMILES branches, and there should be no explicit repeat unit that corresponds to only the atoms found within the pendant chains, nor bonding descriptors that are irrelevant to the backbone connectivity. Figure 2c,d provides two examples of non-permissible choices of repeat units. For polymers with branch-on-branch structures, a backbone must be specified for each type of branch, and the canonicalization procedure for the branches will be defined analogously to that of the main backbone.

Over the defined backbone, there are usually multiple equally valid choices for repeat units. To canonicalize the polymer representation, one must establish a procedure to assign preference over different choices. One such procedure is the structure-based nomenclature specified by the IUPAC (the Purple Book[25] and other standards[26−33]). In the nomenclature, different choices of repeat units are explored through frame-shifts, and the most favorable choice is obtained by a series of precedence rules defined over different chemical motifs. While structure-based approaches like the IUPAC nomenclature can be useful for defining the canonical repeat unit for a homopolymer with a purely head-to-tail configuration, it will be increasingly challenging to apply the same procedure robustly to copolymers when the number of repeat units increases, especially when the repeat units come with an assortment of connectivity patterns. Since exploration of all possible modes of connectivity is a precursor to identifying the potential sets of repeat units, the complex combinatorial nature of the problem will make it overall a difficult task to enumerate and identify the canonical set using a purely structural approach.

To avoid such complexity, a source-based approach is taken here to identify the canonical set of repeat units. The canonical choice is defined as the set that satisfies the following conditions:

1. Within a single repeat unit, all backbone atoms must originate from the same source monomer prior to the polymerization of the backbone; and
2. Repeat units are selected so atoms on the same monomer all appear in the same repeat unit.

When treated with these rules, these repeat unit choices will return a valid canonical BigSMILES.

Notably, the repeat units identified using these algorithms do not always coincide with the structural repeat unit (SRU) identified along the infinitely long backbone. For instance, following condition two, the canonical repeat unit for poly(ethylene) is the two-carbon unit instead of the single carbon SRU. A few examples of the canonical choice are illustrated in Figure 3. While at first this may seem restrictive, it is necessary to enforce the fidelity of the BigSMILES representation; after all, linear polyethylene can only have even numbers of carbons along its backbone due to the mechanism of polymerization.

Without modifications to the pendant groups such as grafting or functional modifications, the canonical choice is synonymous with the set of repeat units corresponding to the source monomers. For polymers made from monomers that generate only a single isomer, the number of repeat units found within the canonical set will equal the number of types of monomers involved in synthesizing the polymer. Meanwhile, for chiral repeat units, such as polypropylene, or monomers that can result in more than one isomer, such as isoprene, the number of repeat

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

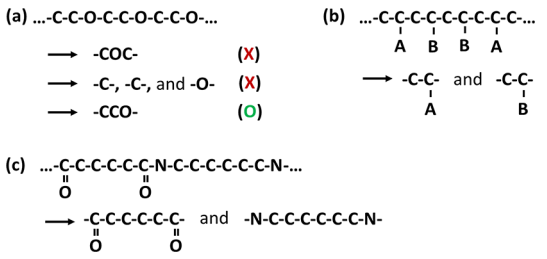


Figure 3. Illustration of canonical choices of repeat units. (a) Poly(ethylene glycol) synthesized from ring-opening polymerization of ethylene oxide. For this polymer, the canonical choice of repeat unit(s) is the set with a single -CCO- repeat unit. The frameshifted version of the repeat unit (-COC-) does not satisfy the first condition for selecting canonical set, as the two terminal carbons cannot originate from the same source monomer. Meanwhile, while the set with three repeat units (-C-, -C-, and -O-) satisfies the first condition (each repeat unit is a single atom unit, hence all atoms within the unit originate from a single monomer), it does not satisfy the second condition. Therefore, the first two choices of repeat units are not considered canonical. (b) Random copolymer synthesized by polymerizing the monomer (C═C(A)) and then partially functionalizing the pendant groups (A→B). For this polymer, both of its repeat units (-CC(A)- and -CC(B)-) correspond to the same type of monomer, and the number of canonical repeat units is larger than the number of types of source monomers. (c) Nylon-6,6 synthesized through polycondensation of hexamethylenediamine and adipic acid. In this case, because of the associated monomers, the canonical set contains two repeat units, corresponding to the diamine and the diacid monomers, respectively. Notably, in this case, the canonical set does not correspond to the SRU found within the infinitely long chain, which corresponds to the unit that constitutes both of the monomers.

units will be larger than the number of monomers. For multistep synthetic procedures, each repeat unit in each step should be written according to the monomer from which it was derived. This means that the BigSMILES can contain nested stochastic objects with the canonical choice of repeat units, as shown in the graft example in Figure 2b. Since the set of monomers is uniquely defined, the canonical set will always be uniquely determined. Similarly, in cases where pendant group modifications were observed, there may be multiple repeat units corresponding to a single type of monomer, as illustrated in Figure 3b. In this case, the number of repeat units in the canonical set will be larger than the number of types of monomers found in the pre-polymerization mixture, but it will still be a unique set.

For some polymerizations where reaction between monomers leads to rearrangement of bonds, the choice of repeat units by grouping all atoms together from a single monomer results in an ensemble defined by the BigSMILES that excludes short oligomers. This occurs in linear step-growth polymers where the reacted and unreacted moieties have different bonding patterns (*e.g.*, alkene *vs* alkane) among the atoms that remain within the polymer. Reactions in this class include thiol-Michael,[34] thiol-ene,[35] thiol-yne,[36] azide-alkyne cycloaddition (both copper-catalyzed[37] and strain-promoted[38]), and Diels–Alder reactions.[39] Figure 4 illustrates a relevant example.

For these reactions, selection rule 2 (all atoms on a monomer must be on the same repeat unit) is modified as follows:

2a A repeat unit that underwent a rearrangement reaction during step growth polymerization is decomposed into its fundamental alphabet using the algorithms presented below (*vide infra*).

2b The alphabet elements, which contain the atoms participating in the rearrangement, are broken into a separate repeat unit, while the remaining atoms are retained in a main repeat unit. An orthogonal set of linkages is specified with bonding descriptors that link the different repeat units derived from the same monomer.

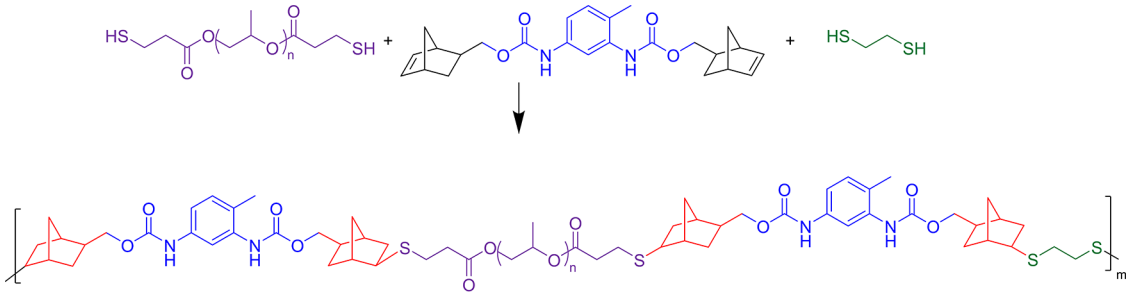
This process is illustrated in Figure 4. Supporting InformationSection 1 includes additional examples.

3.2. Canonicalizing Individual Endgroup Configurations

In principle, given a specific molecular ensemble, once the set of canonical repeat units have been selected, the endgroups become uniquely determined. In general, since the canonical set of repeat units corresponds to the monomers that make up the backbone chain, the endgroups will contain atoms found within precursors such as initiators or terminating agents. For instance, consider poly(ethylene glycol) (PEG) synthesized by reacting ethylene glycol with ethylene oxide. Following the rules for selecting repeat units, the canonical repeat unit for PEG reads -CCO-. Therefore, the (yet to be canonicalized) BigSMILES for this PEG polymer might read:

OCCO{[>1][<1]CCO[>1][<1]}

with an explicit endgroup OCCO and a second implicit hydrogen atom as the second endgroup. Since the polymer is



{[][<1]SCCC(=O)O{[>1][<1]CC(C)O[>1][<1]}C(=O)CCS[<1],[<1]SCCS[<1],
[>1]C1CC2CC1CC2[<2],[>2]COC(=O)Nc1ccc(C)c(c1)NC(=O)OC[>2];[>1][H],[<2]C1CC2C=CC1C2[]}

Figure 4. When the reaction between monomers leads to a rearrangement of bonds, the unit that forms due to bond rearrangement should be written separately in the BigSMILES. In this thiol-ene reaction, the red unit is formed by the reaction of a thiol (orange) and an alkene (black).[40] The empty terminal descriptors "[]" indicate that endgroups are not written outside of the stochastic objects.

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486–500

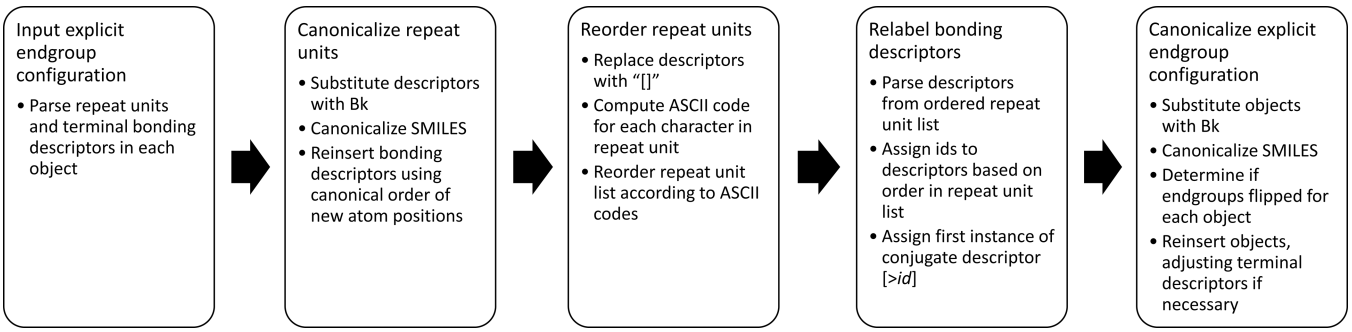


Figure 5. Flowchart of BigSMILES canonicalization for polymers with defined backbones by the priority rules approach. The coding language is Python.

strictly head-to-tail, the conjugate bonding descriptors [<1] and [>1] are used to capture the connectivity between units. Here, the BigSMILES string constitutes the set of molecules that includes n ethylene glycol repeat units, with n = 0, 1, 2, ... Notably, n = 0 is also included in this case because the terminal bonding descriptors leading up to the endgroups ([>1] and [<1]) can be directly joined. Since the BigSMILES string denotes the set of all possible molecular configurations satisfying the provided connectivity patterns between the endgroups and repeat units, the molecule with zero repeat units, corresponding to ethylene glycol, or OCCO, the smallest molar mass molecule contained within the ensemble. By comparison, the BigSMILES O{[>1][<1]CCO[>1][<1]} would represent a different ensemble, which would contain H_2O as its smallest molecule. In contrast, for an amine-terminated nylon-6,6 polymer, with BigSMILES written as

{[>1][<1]NCCCCCCN[<1],[>1]C(=O)CCCCC(=O)[>1][>1]}

the bonding descriptors associated with the endgroups cannot be directly joined. Therefore, the molecular ensemble does not include the case where no repeat unit is found within the molecule. In this case, the molecule with the smallest degree of polymerization is the molecule with one diamine repeat unit.

Notably, in the two examples provided above, each terminus of the polymer is explicitly associated with a well-defined endgroup. However, the termini of a polymer are not always deterministically specified. In many cases, implicit endgroups are provided in a BigSMILES instead of explicit endgroups. Implicit endgroups are endgroups that are not explicitly associated with either end of a polymeric segment. They are provided as molecular fragments containing a single bonding descriptor site, and they denote different probabilistic modes of terminating a polymer. For instance, the following BigSMILES

{[][<1]C(=O)CCCCC(=O)[<1],[>1]NCCCCCCN[>1];
[<1][H],[>1]O[]}

which denotes nylon-6,6 synthesized by polycondensation of diacid and diamine, contains two implicit endgroups. For this polymer, three distinct endgroup configurations are identified, corresponding to the cases of diamine terminated polymer, diacid terminated polymer, and polymer with mixed endgroups. To avoid complexities associated with having polymeric endgroups, upon the selection of canonical repeat units, a transformation is first carried out to convert BigSMILES strings with implicit endgroups into corresponding explicit versions before any further canonicalization procedure is performed. Any BigSMILES string with implicit endgroups is made explicit by first enumerating all possible endgroup configurations, corresponding to the distinct combinations of endgroup pairs. Then, canonicalization is independently performed on individual BigSMILES strings with explicit endgroups that each correspond to one of the endgroup configurations. Finally, the canonicalized BigSMILES are collected and compiled into a canonical expression for the overall polymer. The details of the compilation procedure are discussed in Section 3.2. Following the implicit−explicit transformation, the canonical expression associated with each BigSMILES string with explicit endgroups can be determined by canonicalizing the repeat units (Section 3.2.1), reordering the repeat units (Section 3.2.2), relabeling the bonding descriptors (Section 3.2.3), and canonicalizing the BigSMILES string (Section 3.2.4). Figure 5 shows a flowchart of this canonicalization procedure, and Supporting InformationSection 2 shows the pseudocode for the algorithms implemented.

3.2.1. Canonicalizing the Repeat Units. In general, canonicalization of a repeat unit is similar to the canonicalization of a regular SMILES string. The only difference between the two tasks is that repeat units contain bonding descriptors that are not found in regular SMILES. Therefore, once additional rules supporting the bonding descriptors are added to the SMILES canonicalization algorithms, they can be readily extended to provide canonicalization for BigSMILES repeat units.

Generating a canonical SMILES string requires three core subroutines.[7] First, a labelling subroutine is called to assign canonical labels to each atom found within a molecule. Then, the molecular graph with labeled atoms is traversed in a canonical manner by calling the graph traversal subroutine. Finally, upon traversal of the molecular graph, the canonical SMILES string is generated through the generation subroutine. For each subroutine, many variants have been proposed, leading to multiple versions of SMILES canonicalization algorithms.[6,7,41,42] The only amendment to these algorithms for providing support to BigSMILES repeat units is to extend the labelling subroutine to cover the bonding descriptors.

Here, to provide such extension, the canonicalization algorithm for small molecules used in the cheminformatics toolkit RDKit is applied.[41,43] First, the bonding descriptors are replaced with berkelium (Bk) in the input repeat unit. Then, the graph is canonicalized (using RDKit's function *MolToSmiles*), and a list of how the atom ordering in the string changed as a result of canonicalization is generated by the RDKit function the *GetPropsAsDict*. Using this list, the descriptors are reinserted into the canonicalized string in place of berkelium. Different descriptors can be treated with the same Bk atom in this step because each atom is individually labeled. At the end of these

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

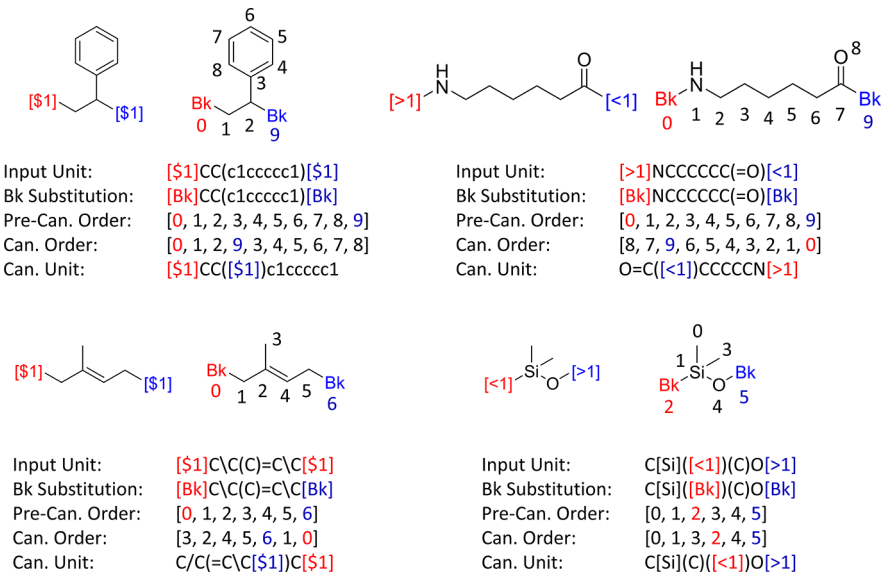


Figure 6. Illustration of repeat unit canonicalization. The descriptors in the input repeat unit are replaced with berkelium (Bk), the canonicalization algorithm from RDKit is applied, the atom indices are tracked in the pre-canonicalized and post-canonicalized strings (Pre-Can. and Can. Order) using RDKit, and the descriptors are reinserted into the string (Can. Unit.).

steps, the multiplicity in how a repeat unit can be expressed is removed. Figure 6 illustrates examples of this procedure.

3.2.2. Reordering the Repeat Units. Once the repeat units have been canonicalized, the list of repeat units is reordered into the canonical form. At the end of this step, the multiplicity in the ordering of repeat units will be eliminated. Here, the canonical order is determined by pairwise comparing the canonical repeat units using string comparisons, with the labels for bonding descriptors, namely, "$", "<", ">", or those also containing numeric labels such as "$1", omitted. A list of ASCII codes is generated for each character in the string using the *ord* function in Python. Then, the list of lists of integers is sorted in ascending order according to the *sorted* function in Python; precedence is determined by comparing the constituent character sequences (or ASCII indices), and seniority is assigned to the repeat unit for which the first non-identical character has the smaller ASCII code. Figure 7 illustrates an example that shows the canonical repeat units, ASCII codes, and sorted repeat unit lists.

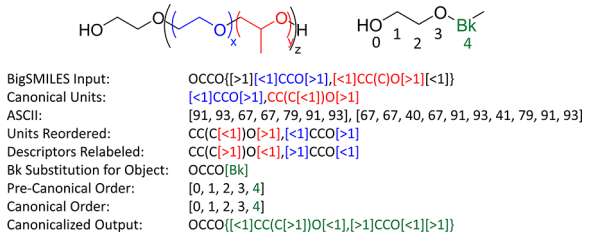


Figure 7. Illustration of the priority rule canonicalization approach for a statistical copolymer poly(ethylene glycol)-*co*-poly(propylene glycol). Each repeat unit is canonicalized, repeat units are reordered according to the list of ASCII values of their characters (bonding descriptors omitted), the descriptors are relabeled, and the entire polymer is canonicalized as a small molecule, encoded in SMILES, by replacing the stochastic objects with berkelium atoms. If the polymer contains nested objects, the procedure is evaluated on the nested objects first. Supporting InformationSection 3 presents examples for a block copolymer, a graft with nested objects in the sidechains (blue berkelium atom represents the nested object), and segmented polymers with nested objects along the backbone.

Supporting InformationSections 3 and 4 show more examples for block copolymers, grafts, and segmented copolymers.

3.2.3. Relabeling the Bonding Descriptors. Finally, once the units have been rewritten in canonical SMILES, the multiplicity in the labelling of bonding descriptors is eliminated by relabeling the bonding descriptors. Starting with the integer 1 for the first label, the numeric indices for the bonding descriptors are reassigned based on the order of appearance within the string (numeric indices are independently tracked for the AA-type descriptors "$" and AB-type descriptors "<", ">"). Furthermore, for the AB-type bonding descriptors, the "<" and ">" symbols are reassigned so that the first appearance of a particular AB-type descriptor is always ">" within the list. Upon the swap between the conjugate descriptors, the terminal descriptors are also adjusted.

Take the ethylene glycol-propylene glycol copolymer as an example from Figure 7. Each terminal bonding descriptor is associated with a specific endgroup ([>1] being associated with the endgroup OCCO, while [<1] being associated with the implicit hydrogen end). Because the bonding descriptors on the repeat units are flipped, the leading and trailing terminals in the final canonical polymer (Can. Polymer) are flipped.

In general, in determining the order of appearance during relabeling of the numeric indices and symbols for bonding descriptors, only the bonding descriptors found within the list of repeat units are considered. The terminal bonding descriptors are not considered. Their positions are dependent on the canonicalization of the BigSMILES string for the entire polymer, discussed in the next subsection, and not solely a function of the contents of the stochastic object.

In the ethylene glycol-propylene glycol copolymer example (Figure 7), which terminal bonding descriptor serves as the leading descriptor will depend on the canonicalization of the entire polymer. As shown in Figure 7, the canonical string reads OCCO{...}:

$$\mathrm{OCCO\{[<1]CC(C[>1])O[<1],[>1]CCO[<1][>1]\}}$$

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

However, if the canonical string reads {...}OCCO, then the positions at which the terminal bonding descriptors appear will be reversed in order to satisfy the correct endgroup connectivity:

{[>1]CC(C[>1])O[<1],[>1]CCO[<1][<1]}OCCO

3.2.4. Canonicalizing Individual Endgroup Configurations. Once the contents of the stochastic object have been canonicalized, the overall BigSMILES string, including the two endgroups and the embedded stochastic object, can be canonicalized by the SMILES (with berkelium) canonicalization algorithm described in Section 3.2.1, with a minor modification in which the stochastic objects will be treated as if they are berkelium atoms.

In the ethylene glycol-propylene glycol copolymer example provided in the previous subsections, Figure 7 illustrates the canonicalization procedure with end groups, and the canonicalized BigSMILES reads:

OCCO{[<1]CC(C[>1])O[<1],[>1]CCO[<1][>1]}

As discussed in the previous subsection, if the end groups were flipped, the terminal descriptors would have to be adjusted.

3.3. Canonicalizing the Overall BigSMILES String

Following the canonicalization of the BigSMILES strings corresponding to individual endgroup configurations (Section 3.2), the canonical BigSMILES string for the overall polymer is obtained by first establishing the canonical order among the canonicalized endgroup configurations and joining the derived strings into a dot-delimited string in this canonical order. The ordering task is done by invoking pairwise string comparison algorithm described in Section 3.2.2.

For example, the BigSMILES encoding for syndiotactic polypropylene synthesized with VCl_4 and $Al(C_2H_5)_2Cl$ as the catalyst reads:

CC{[>1][<1]C[C@@H](C)C[C@H](C)[>1];
[<1]C=CC,[<1]C[C@@H](C)C=CC[]}

Inside the stochastic object, there are two endgroups in this string after the semicolon:

[<1]C=CC

[<1]C[C@@H](C)C=CC

The algorithm automatically converts this representation into two individual end group representations:

CC{[>1][<1]C[C@@H](C)C[C@H](C)[>1][<1]}C=CC

CC{[>1][<1]C[C@@H](C)C[C@H](C)[>1][<1]}
C[C@@H](C)C=CC

Each representation is canonicalized separately according to the procedure in Section 3.2. Then, the two strings must be combined and separated by a dot. The final dot-delimited string would be:

CC=C[C@H](C)C{[<1]C[C@H]([>1])C[C@H](C)C[<1]
[>1]}CC.CC=C{[<1]C[C@H]([>1])C[C@H](C)C[<1]
[>1]}CC

This is because first unique character in the first string listed is "[" (ASCII index of 91) versus "{" (ASCII index of 123).

3.4. Limitations and Future Directions

The procedures in this section can be extended to treat more complex topologies because parsing functions can easily convert a BigSMILES with any number of descriptors and stochastic objects into a SMILES by substituting them with heavy atoms. The types of polymers that can be treated include ring polymers, long chain branched polymers, and H-polymers, as long as a preferred backbone is chosen. Systems such as networks and dendrimers without a preferred backbone are beyond the scope of this method.

A second factor is that the preferred set of repeat units is determined by the monomer chemistry from which the polymer is synthesized. While this definition provides a straightforward way to determine the canonical set and affords a method that maintains readable canonical BigSMILES, in practice, it is challenging to implement this chemistry-based algorithm without significant domain expertise. The algorithm involves information that is not natively found within the BigSMILES string. External input from a domain expert is required in order to write the correct repeat units. A canonicalization procedure for which all needed information is contained in the BigSMILES string, based on connecting polymers to formal languages, is discussed in Section 4.

4. CANONICALIZING BIGSMILES USING FORMAL LANGUAGES

4.1. Connecting Polymers to Formal Languages

Another approach to deriving a canonical expression for the ensemble of molecular states spanned by a polymer is to treat the collection of molecular states as a formal language. In computation theory, a formal language is defined as a collection of sequences that are composed by joining a series of building blocks, and the set of available building blocks is referred to as the alphabet. Meanwhile, the constitutive rules that describe the construction of the set are denoted as the grammar associated with the language. Consider the English language. For English, the building blocks are words, and the alphabet is the entire English vocabulary. Meanwhile, sequences of building blocks correspond to sentences, and collectively, the set of proper English sentences constitutes the English language. Equivalently, the language can also be defined by the English grammar, which provides specifications for generating proper sentences using the English vocabulary.[44,45] Supporting InformationSection 5 provides more information on this topic.

Each polymer ensemble represents a formal language. Individual polymer molecules are sequences constructed by concatenating chemical building blocks, the alphabet. The chemical rules that dictate the allowable connectivity patterns for a polymer are the grammar associated with each polymer language. Figure 8 illustrates this analogy. From this perspective, polymers can be mapped onto formal languages, and the problem of finding unique representations for polymers can be reduced to finding unique representations for individual languages or grammars that define the languages. Because molecules found within polymers with defined backbones can be generated by appending the building blocks one at a time, the grammars that define polymers constitute simple grammatical rules that only involve incremental growth or termination. Given a set of available building blocks, this property of the chemistry allows polymers to be defined and differentiated unambiguously through the specification of their associated grammatical rule sets. As detailed in the following sections, this property provides

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

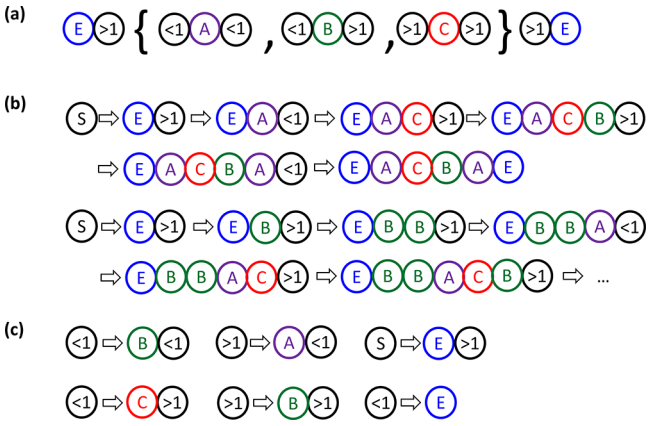


Figure 8. (a) BigSMILES corresponding to a specific polymer language, as defined by the ensemble of molecules formed by joining building blocks or alphabets according to the connectivity patterns specified by the bonding descriptors. (b) Illustrative examples of molecules found within the polymer ensemble, generated by invoking the permissible connectivity patterns specified in the BigSMILES. They symbol "S" denotes the starting symbol. (c) Distilled grammatical rules for defining the polymer.

a solution to canonicalization independent of chemical domain knowledge.

4.2. Converting BigSMILES into Finite Automata

In this section, we describe how polymers with defined backbones can be mapped to graphical representations of a regular language, finite automata. These automata can be thought of as computer machines where the outputs of the machine are the different atomic structures of individual polymer molecules. Figure 9 shows a flowchart of this procedure. To perform this transformation, along the backbone, a linear polymer can be decomposed into a sequence of fundamental building blocks, which are the alphabets or output symbols of the finite automaton. In most cases, a repeat unit or monomer used in polymerization consists of multiple building blocks in the alphabet. A building block is defined as a fragment of the polymer in which the backbone atoms are correlated, but the backbone atoms between building blocks are uncorrelated. This construct ensures that none of the fundamental building blocks can be constructed by the joining of other building blocks, thereby guaranteeing the minimum number of building blocks necessary to construct the polymer. Figure 10 illustrates several such examples of decomposition of the polymer into its alphabet.

An algorithm is constructed to convert a BigSMILES into a set of building blocks that satisfies the rules in the previous paragraph. In the released code, the function *generate_alphabets* generates a sequence of alphabets by finding and breaking bonds between adjacent (1) non-ring atoms, (2) ring and non-ring atoms, and (3) ring atoms only if there is one path between them (adjacent ring atoms that have more than one path in between them are part of the same ring or ring systems). This function uses RDKit,[43] NetworkX,[46] a Python package for the creation and manipulation of networks, the SMARTS grammar that specifies substructure queries,[47] and Chemprop's *extract_subgraph* function for breaking bonds.[48] Figure 11 shows an example, and Supporting InformationSections 3 and 4 show more examples of alphabets generated for homopolymers, statistical, alternating, block copolymers, graft polymers, segmented polymers, and ladders. The pseudocode for this algorithm is included in Supporting InformationSection 2.

This alphabet is now used to transform the BigSMILES into a finite automaton representing the regular language of the polymer. In general, starting from one end of the molecule, the molecular chain can be derived and generated by sequentially appending fundamental building blocks onto the chain. Since in both polymers and regular languages, the permissible candidates that could be appended depend only on the previous unit, this correspondence between the molecular sequences and regular language is an exact one.

In order to convert a BigSMILES into an automaton, we must conceptually think about polymers as directed sequences or labeled paths of concatenated alphabets. First, the function *generate_paths* parses the BigSMILES from left endgroup to right endgroup and generates a list of directed path fragments that connect the two endgroups. Each directed path fragment consists of a left descriptor, SMILES string, and right descriptor. If the right descriptor in one fragment is compatible with the left descriptor in another, then the path fragments can connect. The function also outputs a dictionary that links the SMILES to a directed sequence of alphabets. The function *generate_transitions* takes as input the list of directed path fragments and alphabet sequences, generates a directed graph from the path fragments, and using a graph traversal algorithm from NetworkX, outputs the state-alphabet transitions for the automaton. The machine states of the automaton are unlabeled because the automaton generates molecular realizations by concatenating the alphabets only, which are the edge labels, during a path traversal of the graph from the start state to the end state. The pseudocode for these functions is included in Supporting InformationSection 2.

As an example, consider poly(ethylene glycol)-*co*-poly(propylene glycol) in Figure 12:

OCCO{[>1][<1]CCO[>1],[<1]CC(C)O[>1][<1]}

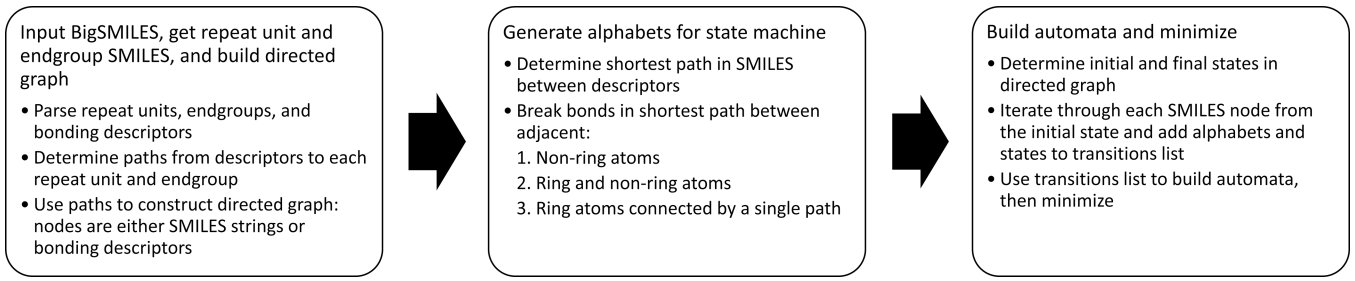


Figure 9. Flowchart of BigSMILES canonicalization for polymers with defined backbones by the formal languages approach. The coding language is Python.

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

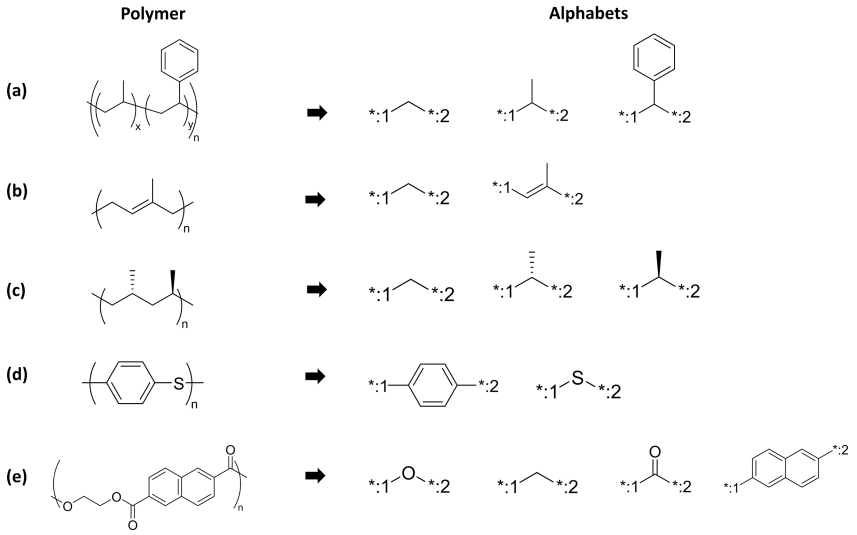


Figure 10. Illustration of the decomposition of the linear backbone into a sequence of alphabets (building blocks). Here, the alphabets are presented so that "*:1" on a building block connects to "*:2" on the previous unit. Therefore, for asymmetric building blocks, the orientation matters. (a) In most cases, fundamental building blocks are composed of single backbone atoms and backbone atoms with pendant groups. (b) Multiatom/bond building blocks may be used to delineate structures that exhibit correlation beyond a single atom/bond such as *cis/trans* isomers. (c) Building blocks may incorporate chirality along the backbone. (d,e) When cyclic structures are found on the backbone, building blocks may also include multiatom rings and fused rings.

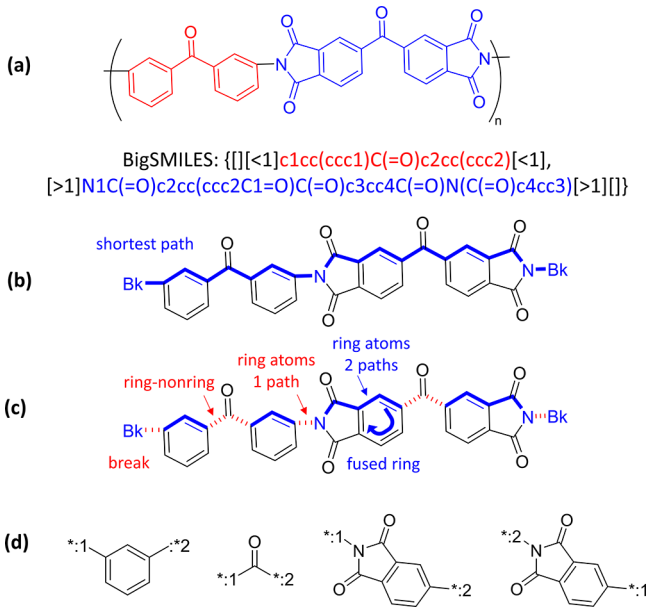


Figure 11. Illustration of how alphabets are determined so that a polymer with a defined backbone can be converted into an automaton. (a) Chemical structure drawing for the polymer of interest and its corresponding BigSMILES. (b) Each BigSMILES repeat unit is converted into an atomistic graph, and the shortest path is found between bonding descriptors (labeled Bk). (c) To generate the alphabets, bonds are broken between adjacent non-ring atoms, adjacent ring and non-ring atoms, and adjacent ring atoms that are connected by a single path. Adjacent ring atoms that have more than one path connecting them are part of the same ring or ring system, and the bond between them should not be broken. Adjacent ring atoms that have exactly one path connecting them are part of different rings or ring systems. (d) After bond breaking, the alphabets are determined and stored as canonicalized SMILES strings.

Conceptually, the function *generate_paths* parses each endgroup and each stochastic object in the order that it is written. The "Start" state points to the OCCO endgroup, which points to the left terminal of the object [>1]. This could be represented as the path:

["Start", "OCCO", "[>1]"]

The algorithm will then determine the path fragments in the repeat unit list that are compatible with the right descriptor [>1] in this endgroup path fragment. There are two possible paths:

["[<1]", "CCO", "[>1]"]

["[<1]", "CC(C)O", "[>1]"]

The left descriptors [<1] in both paths are compatible. This process is the same for nested objects on the backbone and continues until the right-most endgroup, which is an implicit hydrogen, is parsed with the right terminal descriptor of the object:

["[<1]", "", "End"]

The function *generate_transitions* iterates through the path fragments list, builds a directed graph that connects the descriptors to the SMILES, traverses this directed graph, and outputs a list of state-alphabet transitions for the automaton, constructed using *graphviz* visualization.[49] The directed graph is shown in Figure 12b. Using this transitions list, the state machine can be constructed in Figure 12c, with nine machine states as the unlabeled nodes and the directed edges labeled with the appropriate alphabet. Figure 12 also includes atactic polypropylene as a second example.

Supporting InformationSections 3 and 4 show more examples on the conversion between BigSMILES strings and their corresponding automata involving other types of connectivity patterns. Overall, the proposed procedure serves as a constructive proof that any linear polymer ensemble is a regular language. Nevertheless, it should be noted that precisely speaking, the language associated with the finite automaton created using the procedure is not identical to the molecular

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

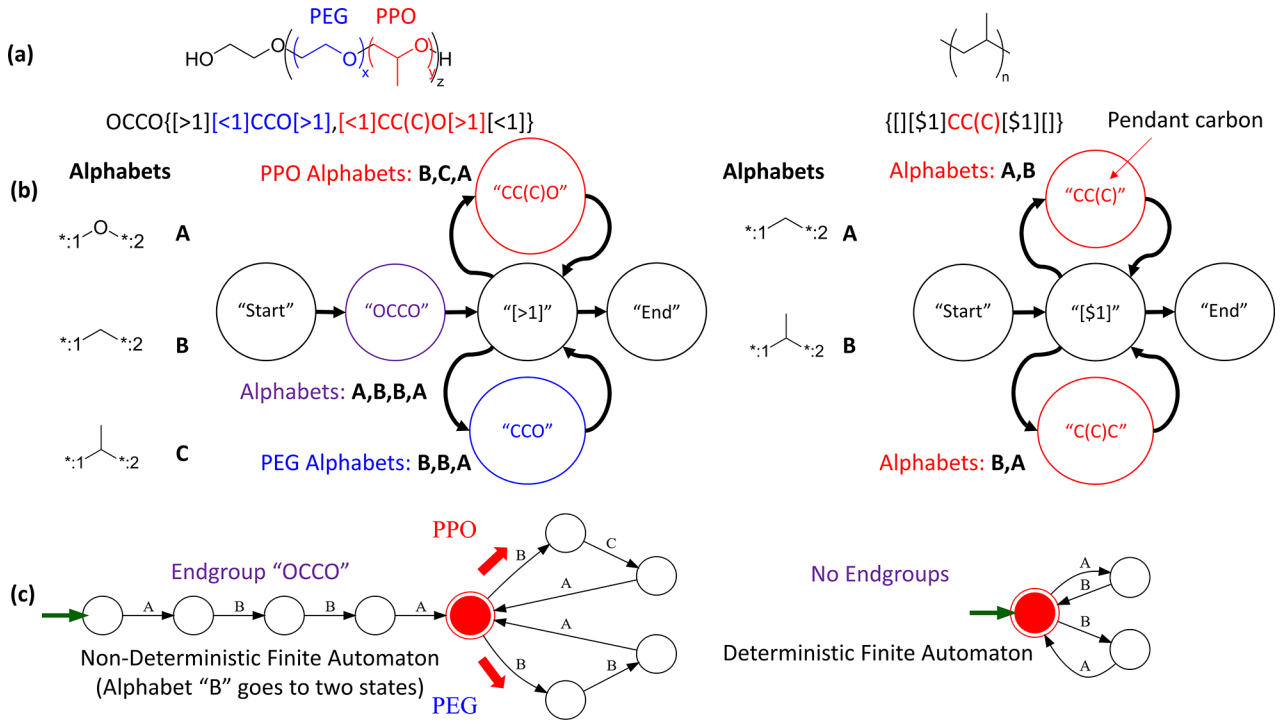


Figure 12. Translating a BigSMILES into the corresponding finite automaton that describes the same language. (a) Two examples are poly(ethylene glycol)-*co*-poly(propylene glycol) and atactic poly(propylene). (b) Algorithm parses the BigSMILES and uses the connectivity information to build a directed graph. Each node is labeled with either a bonding descriptor, repeat unit SMILES, or endgroup SMILES. For each SMILES, alphabets are generated. In the copolymer example, the graph reads from the left endgroup OCCO to the left terminal descriptor, which can connect either to PEG (poly(ethylene glycol)) with repeat unit -CCO- or PPO (poly(propylene glycol)) with repeat unit -CC(C)O-. The graph in panel (b) is converted into a state machine in panel (c). Alphabets are on the edges; nodes or machine states have no labels since the automaton generates molecules by concatenating alphabets only. The green arrow points to the start state (left hand side of the BigSMILES), and the red state is the accept/end state. From the start state to the end state, the machine runs a path traversal, generating a sequence of alphabets. An alphabet from a single machine state can point to more than one state (nondeterministic finite automaton or NFA) or exactly one state (deterministic finite automaton or DFA).

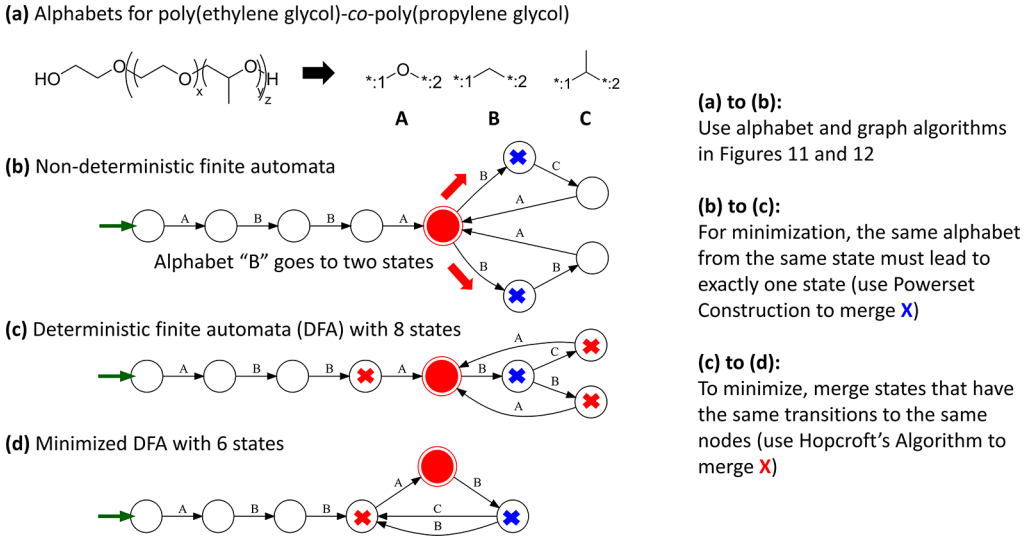


Figure 13. Conversion of a BigSMILES string into a minimized DFA. A green arrow points to the initial state, and the red double circle is the accept state. The example is poly(ethylene glycol)-*co*-poly(propylene glycol), with BigSMILES OCCO{[>1][<1]CCO[>1],[<1]CC(C)O[>1][<1]}.

ensemble delineated by the BigSMILES string. Rather, the language is the union of the molecular ensemble and a null molecule corresponding to the empty sequence. The difference is trivial; henceforth, the molecular ensemble described by BigSMILES and the language described by the automaton will be treated as effectively identical.

4.3. Canonicalizing BigSMILES Using Finite Automata

Each regular language is uniquely associated with a minimal deterministic finite automaton (DFA) with a minimized number of states. Thus, any polymer molecular ensemble can be mapped onto its corresponding minimal DFA as a canonical representation, and any two equivalent polymers will have

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

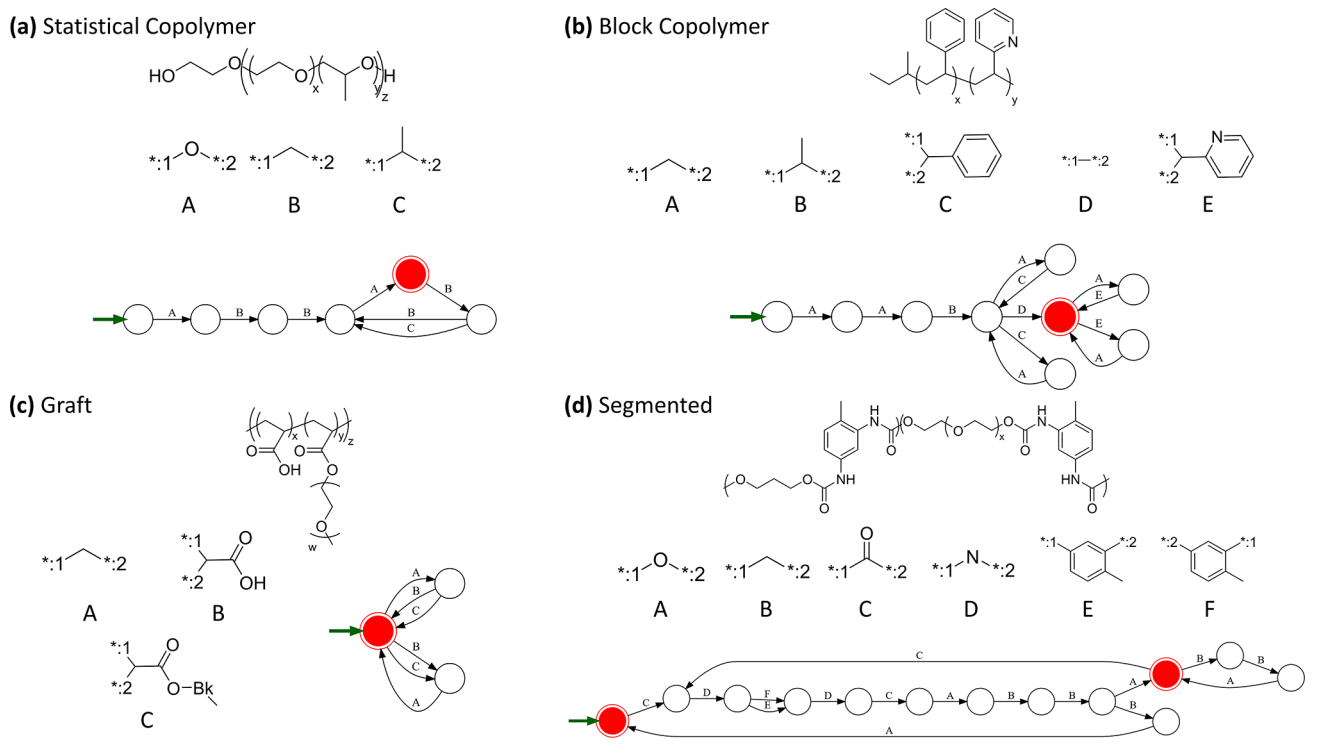


Figure 14. Minimal DFAs for (A) poly(ethylene glycol)-*co*-poly(propylene glycol); (B) *sec*-butyl lithium initiated poly(styrene)-*b*-poly(2-vinylpyridine); (C) poly(acrylic acid)-*graft*-poly(ethylene oxide); (D) polyurethane with oligomeric chain extenders. A green arrow points to the initial state, and the red double circle is the accept state. The side chain object in panel (c) is represented as Bk in the alphabet.

identical minimal DFAs. Upon this transformation, the original problem of finding a unique expression for the random molecular graph has been reduced to finding a unique expression for the directed graph given by the minimal DFA. To arrive at this representation from the automaton constructed in Section 4.2, the automaton is transformed from an NFA or DFA into a DFA using powerset construction,[50] and the minimal DFA is computed using algorithms such as Hopcroft's algorithm (used in this work).[51] Moore's algorithm,[52] or Brzozowski's algorithm.[53]Figure 13 shows an example of this process for poly(ethylene glycol)-*co*-poly(propylene glycol). Code for converting an NFA into a DFA and minimizing it were taken from GitHub, with graph visualization from *graphviz*.[49] The code in this repository has been thoroughly validated on an extensive BigSMILES dataset of different chemistry shown in Supporting InformationSections 3 and 4.

While different BigSMILES representations can encode the same ensemble, depending on the choice of repeat units, each ensemble has only one minimal DFA. Thus, different BigSMILES representations that encode the same ensemble can be mapped to the same minimal DFA, providing a canonical representation of the ensemble. For example, for polymers capped with endgroups, frameshifts in the BigSMILES repeat units into the endgroups can still encode the same ensemble and regular language and are mapped to the same minimal DFA. An example is shown in Supporting InformationSection 6.

Under the context of a defined backbone, the same procedure can be extended to provide unique representations for branched polymers with nested stochastic objects. For graft polymers with stochastic objects in the pendant groups, the building block and automata algorithms are applied to the main backbone, and the building blocks contain the stochastic objects from the side chains. For segmented polymers with stochastic objects nested along the backbone, the algorithm incorporates the building blocks for these objects in the NFA or DFA representation. Figure 14 shows examples of the canonicalization of different polymers with varying chemistry and architecture canonicalized using this approach, and Supporting InformationSections 3 and 4 provides more examples for linear polymers, block copolymers, segmented polymers, grafts, and ladders.

4.4. Limitations and Future Directions

It should be noted that while the DFA describes the same molecular ensemble as the original BigSMILES string, it is generally not a trivial task to transform the DFA back into a BigSMILES string. Therefore, the formal language approach to canonicalization sacrifices human readability. However, it offers potential advantages in terms of the way the algorithm could generate unique representations without explicit knowledge on the synthetic chemistry, making polymer structures more easily processed computationally. Notably, since this approach directly operates on top of the corresponding regular language, the procedure is robust against operations such as frame shifts on the repeat units, and the *a priori* selection of a canonical set of repeat units is not necessary.

As mentioned earlier, the presented procedure is designed for polymer molecules with their endgroups specified. While the procedure can be readily applied to construct automata that generate ensembles of polymeric fragments, for polymers that are not properly capped with endgroups, the derived automata will be susceptible to frameshifts in the selection of repeat units. For instance, the ensemble of poly(ethylene glycol) fragments composed of -CCO-repeat units and the ensemble composed from -COC-units will not reduce to the same DFA. This is because these two entities represent distinct ensembles, with the former composed of molecular fragments having a carbon terminal on one end and oxygen terminal on the other, whereas

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

the latter ensemble consists of molecular fragments with two carbon terminals. In general, for proper polymer molecules, such frameshifts in the repeat unit will be compensated by the corresponding change in the endgroups, thereby allowing the reduction to a unique automaton.

In this approach, polymeric chains are decomposed into units that span a few backbone atoms, with most of the units spanning only a single backbone atom. Upon this decomposition, structural information pertaining to the long-range correlation across the backbone is destroyed. As a result, while some local structural features, such as the cis-trans configuration or the stereochemistry of local chiral centers, can be captured, long-range correlational patterns cannot generally be captured and resolved.

As mentioned in the building block algorithm, to ensure the uniqueness of the resulting output, each fundamental building block must be assigned a unique label. While this requirement is met by labelling each building block with its corresponding canonical SMILES string, an alternative approach to this is to assign hashed representations, such as the Morgan fingerprint, as labels to each block. While in theory the use of hashed labels compromises the uniqueness of the final derived string, in practice the compromise is often negligible if a good hash function with low collision rates is selected.

The minimal DFA provides a unique representation of a BigSMILES string, and in digital databases, the connectivity table of the minimal DFA, with canonicalized SMILES strings as the alphabets, can be archived and searched, but this is not a textual representation. However, this procedure of converting molecular ensembles into finite state machines is powerful because it becomes possible to assign unique textual identifiers to the minimal DFA. For example, graph traversal algorithms can be used to generate a unique SMILES string from the minimal DFA. Such algorithms have been developed for small molecule graphs.[6,7] Moreover, hashing functions can be applied to convert the graph into a fixed sized vector representation, which is generally more memory compact. This is especially important if the generated representations are to be deployed as identifiers for locating database entries, indexing documents, or evaluating if two polymers are equivalent. However, after hashing, the identifiers could no longer be used to recreate the molecular ensemble, nor could it be used to perform substructure matches in structure-based searches.

5. DISCUSSION OF CANONICALIZATION APPROACHES

The approaches discussed in the previous two sections outline strategies for deriving unique representations for polymers. In Section 3, a canonicalization scheme for the BigSMILES string is developed. The scheme has two major components. In the first part, a strategy for identifying the preferred set of canonical repeat units is specified. To retain chemical legibility of the canonicalized BigSMILES strings, the canonical repeat units are designed to align with the monomers that were used to synthesize the polymer. This choice will prevent the standardization procedures from leading to expressions that do not make much chemical sense, maintaining the chemical intuitiveness that makes SMILES a successful line notation, and this is a key advantage of this approach. Following the selection of the repeat units, a second algorithm that canonicalizes the stochastic object as well as the overall BigSMILES string is proposed. The expression generated by the second algorithm is only unique up to the choice of repeat units, and the uniqueness of the overall output largely depends on the selection of the preferred set of repeat units. However, in the proposed design, as the selection of the canonical set of repeat units requires chemical knowledge that is not explicitly contained within a BigSMILES string, implementation of the proposed chemistry-based algorithm can be challenging if the BigSMILES string is provided in isolation. In practice, this issue is often mitigated because the BigSMILES strings are frequently provided alongside other data. For instance, in the case where the polymer is referenced in a lab notebook file or in a PolyDAT file,[24] the context surrounding the original reaction is often available in the form of the synthetic graph. In this case, the algorithm provides a robust way to derive canonical representations.

In Section 4, a regular-language-based approach is proposed to convert polymer molecular state ensembles into minimized finite state machines. While neither SMILES nor BigSMILES conform to regular grammar (rather, both form context-free languages because they require a matching set of parentheses and brackets), the nature of linear polymers allows the backbone connectivity to be cast into a regular language. In this approach, since the object of operation is the molecular ensemble, the algorithm is capable of reducing BigSMILES strings with noncanonical repeat units into a unique representation. Practically, this feature makes the algorithm robust to frame shift operations and unusual choices of bracket-crossing bonds a key advantage of this approach. As such, the unique representations generated by this algorithm can serve as searchable identifiers in chemical databases. Nevertheless, the conversion of these graph representations into strings require additional algorithms, such as graph traversal algorithms or hash functions.

6. CONCLUSIONS

In this work, strategies to obtain unique BigSMILES strings for polymers with defined backbones are developed. First, a scheme to derive canonical BigSMILES strings is proposed. This approach provides a general strategy to generate standardized BigSMILES representations that are highly legible and chemically intuitive for users. However, as the algorithm relies on knowing the chemical history of the polymer, its utility can be limited in cases where the polymer is considered in isolation. To this end, a second strategy that leverages the random graph nature of BigSMILES line notation is proposed to compliment the first approach. In the second approach, a polymer is treated as a collection of the possible molecular connectivity states specified by the BigSMILES string. For a linear polymer, the ensemble constitutes a regular language, with the building blocks along the polymer backbone as the alphabets to the language. This property is exploited, and the minimal DFA for the corresponding regular language can be hashed to provide a unique representation for the polymer of interest.

In summary, the two proposed strategies provide both an algorithm to generate canonical BigSMILES stochastic objects as well as a one-to-one mapping function between the ensemble of permissible connectivity states for a given polymeric system and a text representation. Together, the proposed algorithms provide both amendments to the base BigSMILES line notation system that can be useful for constructing readable standardized representations as well as unique identifiers that could be used in chemical information systems.

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acspolymersau.2c00009.

Database validation of canonicalization approaches for homopolymers, statistical and AA-BB-type linear copolymers, block copolymers, graft copolymers, and segmented copolymers; explanations of formal languages, regular languages, and finite automata (PDF)

■ AUTHOR INFORMATION

Corresponding Author

Bradley D. Olsen − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-7272-7140; Email: bdolsen@mit.edu

Authors

Tzyy-Shyang Lin − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-8265-6702

Nathan J. Rebello − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0002-0178-7701

Guang-He Lee − *Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States*

Melody A. Morris − *Department of Chemical Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, United States;* orcid.org/0000-0001-5597-154X

Complete contact information is available at: https://pubs.acs.org/10.1021/acspolymersau.2c00009

Notes

The authors declare no competing financial interest.

This code is available on the GitHub repository https://github.com/olsenlabmit/bigsmiles-canon. All packages and their versions used in this code, such as NetworkX and RDKit, will be released. The user inputs a BigSMILES, and the software outputs the alphabet, NFA, equivalent DFA, minimized DFA, and the canonicalized string from the priority rules approach.

■ ACKNOWLEDGMENTS

This work was funded by the Center for the Chemistry of Molecularly Optimized Networks, a National Science Foundation (NSF) Center for Chemical Innovation (CHE-1832256).

■ REFERENCES

(1) Ash, S.; Cline, M. A.; Homer, R. W.; Hurst, T.; Smith, G. B. SYBYL line notation (SLN): A versatile language for chemical structure representation. *J. Chem. Inf. Comput. Sci.* **1997**, *37*, 71−79.

(2) Gakh, A. A.; Burnett, M. N. Modular chemical descriptor language (MCDL): composition, connectivity, and supplementary modules. *J. Chem. Inf. Comput. Sci.* **2001**, *41*, 1494−1499.

(3) Heller, S.; McNaught, A.; Stein, S.; Tchekhovskoi, D.; Pletnev, I. InChI-the worldwide chemical structure identifier standard. *J. Cheminf.* **2013**, *5*, 1−9.

(4) Vollmer, J. J. Wiswesser line notation: an introduction. *J. Chem. Educ.* **1983**, *60*, 192.

(5) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.

(6) Weininger, D.; Weininger, J. L. SMILES. 2. Algorithm for generation of unique SMILES notation. *J. Chem. Inf. Comput. Sci.* **1989**, *29*, 97−101.

(7) O'Boyle, N. M. Towards a Universal SMILES representation-A standard method to generate canonical SMILES based on the InChI. *J. Cheminf.* **2012**, *4*, 22.

(8) Irwin, J. J.; Sterling, T.; Mysinger, M. M.; Bolstad, E. S.; Coleman, R. G. ZINC: a free tool to discover chemistry for biology. *J. Chem. Inf. Model.* **2012**, *52*, 1757−1768.

(9) Sterling, T.; Irwin, J. J. ZINC 15−ligand discovery for everyone. *J. Chem. Inf. Model.* **2015**, *55*, 2324−2337.

(10) Quirós, M.; Gražulis, S.; Girdzijauskaitė, S.; Merkys, A.; Vaitkus, A. Using SMILES strings for the description of chemical connectivity in the Crystallography Open Database. *J. Cheminf.* **2018**, *10*, 1−17.

(11) Ihlenfeldt, W. D.; Bolton, E. E.; Bryant, S. H. The PubChem chemical structure sketcher. *J. Cheminf.* **2009**, *1*, 1−9.

(12) Favre, H. A.; Powell, W. H.*Nomenclature of organic chemistry: IUPAC recommendations and preferred names 2013*; Royal Society of Chemistry, 2013.

(13) *CAS REGISTRY*, <https://www.cas.org/cas-data/cas-registry> (accessed Jan 5, 2022).

(14) Lowe, D. M.; Corbett, P. T.; Murray-Rust, P.; Glen, R. C. Chemical name to structure: OPSIN, an open source solution. *J. Chem. Inf. Model.* **2011**, *51*, 739−753.

(15) *ChemSpider: Search and Share Chemistry*; <http://www.chemspider.com/> (accessed Feb 20, 2022).

(16) *Chemical Identifier Resolver*; <https://cactus.nci.nih.gov/chemical/structure> (accessed Feb 21, 2022).

(17) Coley, C. W.; Green, W. H.; Jensen, K. F. Machine learning in computer-aided synthesis planning. *Acc. Chem. Res.* **2018**, *51*, 1281−1289.

(18) Montavon, G.; et al. Machine learning of molecular electronic properties in chemical compound space. *New J. Phys.* **2013**, *15*, No. 095003.

(19) Wu, Z.; et al. MoleculeNet: a benchmark for molecular machine learning. *Chem. Sci.* **2018**, *9*, 513−530.

(20) Napolitano, F.; et al. Drug repositioning: a machine-learning approach through data integration. *J. Cheminf.* **2013**, *5*, 1−9.

(21) Coley, C. W.; et al. A graph-convolutional neural network model for the prediction of chemical reactivity. *Chem. Sci.* **2019**, *10*, 370−377.

(22) Lin, T.-S.; et al. BigSMILES: A Structurally-Based Line Notation for Describing Macromolecules. *ACS Cent. Sci.* **2019**, *5*, 1523−1531.

(23) *The BigSMILES Line Notation*; <https://olsenlabmit.github.io/BigSMILES/docs/line_notation.html#the-bigsmiles-line-notation> (accessed May 5, 2022).

(24) Lin, T.-S.; et al. PolyDAT: a generic data schema for polymer characterization. *J. Chem. Inf. Model.* **2021**, *61*, 1150−1163.

(25) Jones, R. G.; Division, I. U.; Wilks, E. S.*Compendium of polymer terminology and nomenclature: IUPAC recommendations, 2008*; RSC Pub., 2009.

(26) Hiorns, R. C.; et al. A brief guide to polymer nomenclature. *Polymer* **2013**, *54*, 3−4.

(27) Gilbert, R.; et al. Dispersity in polymer science. *Pure Appl. Chem.* **2009**, *81*, 351−353.

(28) Kahovec, J.; et al. Source-based nomenclature for non-linear macromolecules and macromolecular assemblies (IUPAC Recommendations 1997). *Pure Appl. Chem.* **1997**, *69*, 2511−2522.

(29) Maráchal, E.; Wilks, E. Generic source-based nomenclature for polymers (IUPAC Recommendations 2001). *Pure Appl. Chem.* **2001**, *73*, 1511−1519.

(30) Ring, W.; Mita, I.; Jenkins, A.; Bikales, N. Source-based nomenclature for copolymers (Recommendations 1985). *Pure Appl. Chem.* **1985**, *57*, 1427−1440.

(31) Kahovec, J.; Fox, R.; Hatada, K. Nomenclature of regular single-strand organic polymers (IUPAC Recommendations 2002). *Pure Appl. Chem.* **2002**, *74*, 1921−1956.

(32) Fox, R.; Bikales, N.; Hatada, K.; Kahovec, J. Structure-based nomenclature for irregular single-strand organic polymers (IUPAC Recommendations 1994). *Pure Appl. Chem.* **1994**, *66*, 873−889.
(33) Metanomski, W. V.; Bareiss, R. E.; Kahovec, J.; Loening, K. L.; Shi, L.; Shibaev, V. P. Nomenclature of regular double-strand (ladder and spiro) organic polymers (IUPAC Recommendations 1993). *Pure Appl. Chem.* **1993**, *65*, 1561−1580.
(34) Allen, C.; Fournier, J.; Humphlett, W. The thermal reversibility of the Michael reaction: IV, Thiol adducts. *Can. J. Chem.* **1964**, *42*, 2616−2620.
(35) Griesbaum, K. Problems and possibilities of the free-radical addition of thiols to unsaturated compounds. *Angew. Chem., Int. Ed.* **1970**, *9*, 273−287.
(36) Lowe, A. B.; Hoyle, C. E.; Bowman, C. N. Thiol-yne click chemistry: A powerful and versatile methodology for materials synthesis. *J. Mater. Chem.* **2010**, *20*, 4745−4750.
(37) Rostovtsev, V. V.; Green, L. G.; Fokin, V. V.; Sharpless, K. B. A stepwise huisgen cycloaddition process: copper (I)-catalyzed regioselective "ligation" of azides and terminal alkynes. *Am. Ethnol.* **2002**, *114*, 2708−2711.
(38) Agard, N. J.; Prescher, J. A.; Bertozzi, C. R. A strain-promoted [3+ 2] azide− alkyne cycloaddition for covalent modification of biomolecules in living systems. *J. Am. Chem. Soc.* **2004**, *126*, 15046−15047.
(39) Briou, B.; Ameduri, B.; Boutevin, B. Trends in the Diels−Alder reaction in polymer chemistry. *Chem. Soc. Rev.* **2021**, *50*, 11055−11097.
(40) Jin, K.; Leitsch, E. K.; Chen, X.; Heath, W. H.; Torkelson, J. M. Segmented thermoplastic polymers synthesized by thiol−ene click chemistry: examples of thiol−norbornene and thiol−maleimide click reactions. *Macromolecules* **2018**, *51*, 3620−3631.
(41) Schneider, N.; Sayle, R. A.; Landrum, G. A. Get Your Atoms in Order - An Open-Source Implementation of a Novel and Robust Molecular Canonicalization Algorithm. *J. Chem. Inf. Model.* **2015**, *55*, 2111−2120.
(42) *OpenSMILES specification*; <http://opensmiles.org/opensmiles.html> (accessed April 16, 2022).
(43) Landrum, G. Rdkit documentation. *Release* **2013**, *1*, 1−79.
(44) Hopcroft, J. E.; Motwani, R.; Ullman, J. D.Automata theory, languages, and computation. *Int. Ed.***2006**, *24*.
(45) Sipser, M.*Introduction to the Theory of Computation*; Cengage Learning, 2012.
(46) Hagberg, A.; Swart, P.; Chult, D.*Exploring network structure, dynamics, and function using NetworkX*; Los Alamos National Lab. (LANL): Los Alamos, NM (United States), 2008.
(47) *SMARTS - A Language for Describing Molecular Patterns*; <https://www.daylight.com/dayhtml/doc/theory/theory.smarts.html> (accessed March 8, 2022).
(48) Yang, K.; et al. Analyzing learned molecular representations for property prediction. *J. Chem. Inf. Model.* **2019**, *59*, 3370−3388.
(49) Ellson, J.; Gansner, E.; Koutsofios, L.; North, S. C.; Woodhull, G.*International Symposium on Graph Drawing*; Springer, 483−484.
(50) Rabin, M. O.; Scott, D. Finite automata and their decision problems. *IBM J. Res. Dev.* **1959**, *3*, 114−125.
(51) Hopcroft, J.An n log n algorithm for minimizing states in a finite automaton. In *Theory of machines and computations*; Academic Press, 189−196. 1971.
(52) Moore, E. F. Gedanken-experiments on sequential machines. *Autom. Stud.* **1956**, *34*, 129−153.
(53) Brzozowski, J.A.: *Canonical regular expressions and minimal state graphs for definite events. In: Proceedings of the Symposium on Mathematical Theory of Automata*; MRI Symposia Series, Polytechnic Press: Polytechnic Institute of Brooklyn, N.Y, 1963, Vol. *12*, pp. 529−561.

https://doi.org/10.1021/acspolymersau.2c00009
ACS Polym. Au 2022, 2, 486−500