Substructure-Aware Program Synthesis for Automated Chemical Reaction Network Discovery

Adam Piotr Szymaniak[1]

Supervisors: Sebastijan Dumančić[1]**, Reuben Gardos Reid**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfilment of the Requirements
For the Bachelor of Computer Science and Engineering
19th June 2026

Name of the student: Adam Piotr Szymaniak
Final project course: CSE3000 Research Project
Thesis committee: Sebastijan Dumančić, Reuben Gardos Reid, Jana Marie Weber

An electronic version of this thesis is available at http://repository.tudelft.nl/.

Abstract

Chemical Reaction Networks (CRNs) are essential for understanding complex reactive processes, yet incomplete experimental data often leave many networks only partially known. Grammar-driven program synthesis offers an approach to completing partial CRNs, but atom-by-atom construction of molecular candidates causes a severe combinatorial explosion, and the baseline synthesiser lacks awareness of the structural context of target molecules. It is not yet known whether substructure-aware heuristics can improve the computational tractability of CRN discovery via program synthesis. To investigate this, BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) fragments were incorporated into the molecular context-free grammar, and Tanimoto similarity of Morgan2 fingerprints was used to guide reaction and network synthesis. The results show that BRICS fragmentation achieves a 13.8 percentage-point improvement in completing partial reactions on a dataset missing complex organic compounds by encoding large substructures as single grammar rules. Conversely, the enhanced grammar solves 27.4 percentage points fewer problems than the baseline on reactions missing small non-carbon species, as increased branching delays discovery of small molecules. Moreover, molecular similarity guidance does not improve performance in reaction rebalancing from SynRXN datasets, but it substantially reduces the search space in an example esterification CRN synthesis problem, requiring 2,565 fewer candidate reactions and 409 fewer candidate networks before discovering the targets. Thus, BRICS fragmentation and similarity-guided heuristics have distinct strengths. Future frameworks should split the candidate molecule pool between fragment-enhanced and atom-by-atom methods to successfully capture both large structural fragments and small, dissimilar species.

1 Introduction

Effective models are essential for understanding and predicting complex chemical processes. In practice, chemical transformations rarely occur as a single, direct conversion of reactants into products. Instead, they unfold through a sequence of smaller reaction steps, passing through intermediate species along the way. A Chemical Reaction Network (CRN) formalises this structure by specifically defining two elements: a set of chemical species and a corresponding set of reactions that describe how these species transform and interact [1]. The CRN thus represents all possible routes by which a chemical system evolves from its initial reactants to its final products.

Nevertheless, nearly every reported CRN is incomplete. Data extracted from electronic lab notebooks, patents, or the literature often lack species or stoichiometric assignments [2]. Furthermore, it can be difficult to detect short-lived intermediate species via experimental spectroscopy [3].

Given these challenges, researchers turn to computational methods to address these data gaps. While identifying missing species and reactions can occasionally be done manually, this approach is highly labour-intensive and limited to small chemical systems [3]. Consequently, automated network exploration approaches have become essential to solve these problems [1].

However, computational automation introduces its own challenge: exploring chemical reaction space remains computationally expensive, since identifying viable pathways requires navigating an exponentially growing combinatorial space of molecular structures and reaction routes as their size and complexity increase [1]. Overcoming this bottleneck is therefore the main challenge in developing efficient, scalable, and automated tools for reaction network exploration.

To address these challenges, recent work by Wijers—drawing inspiration from the syntax-guided synthesis approach introduced by Cardelli et al. [4]—formalised the completion of partial CRNs as a program synthesis problem [5]. This approach proposes a modular pipeline that incrementally builds candidate molecules, reactions, and networks from scratch based on context-free grammars (CFGs).

In this framework, candidate molecules are constructed atom-by-atom via an iterative expansion of an abstract syntax tree (AST). This atom-by-atom construction presents a severe computational bottleneck: complex molecular structures require excessively deep ASTs, and the number of candidate programs grows exponentially with depth. As a result, the BFS iterator must exhaustively enumerate all shallower candidates before reaching more complex targets.

To address this bottleneck, retrosynthetic fragmentation algorithms such as BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) decompose complex, drug-like target molecules into chemically meaningful high-level building blocks [6]. By cleaving molecules specifically at strategic bonds, BRICS yields fragments that are highly likely to be a part of the missing species.

By incorporating these high-level building blocks directly into the CFG as single rules, the synthesiser no longer needs to construct these large substructures character by character. This reduces the overall depth of the ASTs required to represent complex candidate species. However, the introduction of BRICS fragments also presents a new challenge: by adding more rules to the grammar, it increases the number of candidate molecules and, in turn, the number of possible reactions.

To prevent the overproduction of non-viable reaction networks, molecular similarity scoring can serve as a heuristic to guide reaction synthesis. Instead of treating all candidate molecules as equally plausible, the synthesiser prioritises those structurally similar to observed molecules. For example, Coley et al. showed that molecular similarity alone can recover known reaction precedents with products similar to the target molecule, from which reaction templates are extracted to generate plausible precursor molecules [7]. Building on this, Guo and Schwaller demonstrated that such similarity measures effectively guide generative search toward molecules containing required substructures [8].

To address the computational limitations of atom-by-atom molecule synthesis and the lack of structural guidance in can-

didate generation, this study investigates the following question: *How do substructure-aware heuristics impact the computational tractability of Chemical Reaction Networks discovery via program synthesis?* To structure the research, the main question was broken down into two sub-questions:

1. How does the introduction of fragments from known molecules as high-level building blocks impact the molecule synthesiser's ability to construct complex structures?
2. How does molecular similarity scoring impact the number of candidate reactions and networks generated prior to target discovery?

To answer these questions, this study integrates two substructure-aware heuristics into the grammar-driven program synthesis CRN discovery pipeline proposed by Wijers [5]. First, BRICS fragments are directly added to the molecular CFG by combining complex substructures into single rules. Second, Tanimoto similarity of Morgan2 fingerprints is introduced to guide reaction and network synthesis toward candidates containing molecules similar to observed ones.

Finally, this study tests the synthesiser's effectiveness at rebalancing incomplete reactions using the curated SynRXN test sets [2]. The influence of the molecule-similarity guidance heuristic was also evaluated across the full CRN discovery process using an example esterification network.

2 Background and Related Works

This section introduces key concepts and literature supporting automated CRN discovery via program synthesis.

2.1 Chemical Reaction Networks

A Chemical Reaction Network (CRN) is a formal representation of chemical processes, defined by a set of species and reactions that describe their interactions. Each reaction includes stoichiometric coefficients that indicate the proportions of reactant and product species. A CRN can also be represented as a directed bipartite graph, where edges connect reactant species to reaction nodes and reaction nodes to product species [3]. Moreover, for a CRN to accurately model physical systems, every reaction must be atom- and charge-balanced.

2.2 SMILES Notation

To process CRNs computationally, the participating molecules are encoded with the Simplified Molecular Input Line Entry System (SMILES). It is a language that encodes two-dimensional molecular graph representations as compact one-dimensional linear strings of characters [9]. Although a single molecular graph could theoretically be written as several valid string permutations, there exist canonicalisation algorithms that generate a unique SMILES string for any two-dimensional graph structure [10].

This study builds upon the baseline prototype for CRN discovery using program synthesis developed by Wijers [5], which uses a subset of explicit SMILES language rules to provide algorithmic clarity. All the atoms are explicitly written individually in square brackets (e.g. [H], [O]). Similarly, all bonds are explicit (e.g. -, =). A more detailed explanation can be found in the author's thesis [5]. This study extends the SMILES language support to include formal charges and reactions for parsing the reaction rebalancing tasks from the SynRXN dataset [2].

2.3 Program Synthesis for Chemical Reaction Network Discovery

Program synthesis is the task of automatically finding a program in a given language that meets a user's intent, typically expressed through a set of constraints or high-level specifications [11]. Unlike compilers and interpreters, which translate code into lower-level instructions, program synthesisers approach program creation as a search problem.

In automated CRN discovery, the target program is the goal CRN being sought. It includes unobserved species (as SMILES strings), balanced reactions, and the full CRN structure [5]. The requirements are based on partial experimental data, such as observed species and their measured concentration time-series profiles. As shown in Figure 1, by treating CRN discovery as a program synthesis problem within the `Herb.jl` [12] framework, the solution searches for unobserved species and a valid reaction network. The simulated concentration profiles for this network are then compared with the empirical data.

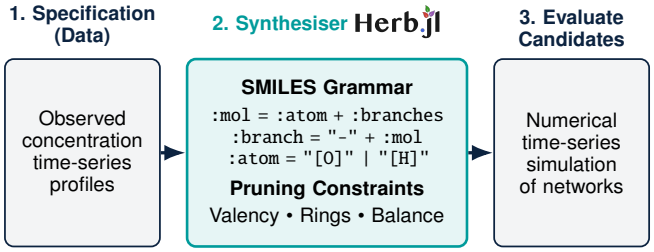

Figure 1: The Program Synthesis Framework applied to Chemical Reaction Network (CRN) Discovery

The search space of all possible CRNs is structured using CFGs. In `Herb.jl`, these grammars specify how candidate molecules, reactions, and entire networks are represented as ASTs: nodes correspond to grammar rules, and their children correspond to specific rule expansions [5, 13].

The baseline approach [5] breaks down program synthesis into a hierarchy of simpler components. At the lowest level, the molecule synthesiser uses a grammar based on a subset of SMILES. It creates molecules with terminal rules for atoms, bonds (e.g., -, =), and ring closures. Additional rules define how these elements connect into branches and chains. Figure 2 demonstrates a partial AST of a molecule constructed with that CFG.

Next, at the reaction level, the reaction synthesiser represents chemical transitions as two groups of molecules: one for reactants, one for products. Finally, at the network level, the grammar defines the whole CRN as a list of unique reactions. Depending on the setup, these synthesisers can work step by step or as a single unit.

Despite the structured approach provided by grammars, a key challenge remains: the primary computational bottleneck in CRN discovery with program synthesis is the sheer size

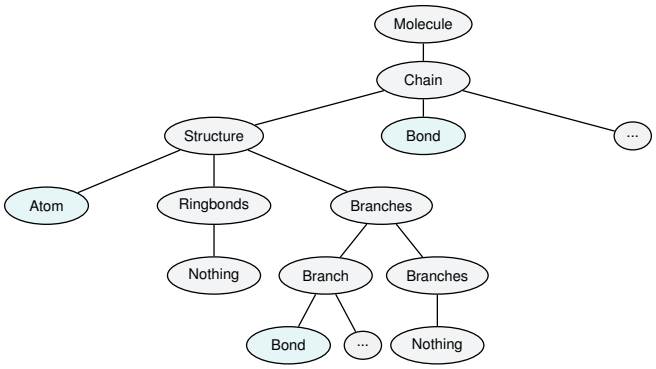

Figure 2: Partial AST of a molecule represented with the baseline grammar. The Atom and Bond nodes represent terminals that are part of a resulting SMILES string.

of the search space. Exhaustively generating and simulating every possible CRN is infeasible. To address this, the baseline approach uses constraint programming in `Herb.jl` [13] to prune the search space of candidate ASTs before programs are fully generated.

The implementation of these constraints begins at the molecular synthesis stage, where they ensure that the generated species are chemically valid. These also help the synthesiser explore as few symmetrical structures as possible. While the AST representing a molecule is forming, a special constraint gathers all bond type nodes to restrict neighbouring atom assignments. For example, if it detects two bond nodes with an atom between them, it instantly rules out hydrogen, which can only form a single bond. Additionally, a separate constraint is applied to restrict ring (cycle) formation. This constraint ensures each digit appears exactly twice and prevents rings between directly connected atoms, as this is equivalent to using a higher-order bond.

Following molecule synthesis, during reaction synthesis, constraints ensure that synthesised reactions are mass-balanced, and commutative duplicates are eliminated. The synthesiser calculates sets of possible molecule distributions to one side of an equation and computes intersections of atom counts to prune reactions that cannot be mass-balanced. Since molecule order on one side of a reaction does not affect outcomes, a strict molecule ordering is used to prevent formation of duplicate molecule lists.

Finally at the network synthesis stage, a constraint ensures that the set of included reactions contains all observed molecules.

2.4 BRICS Fragmentation

BRICS (Breaking of Retrosynthetically Interesting Chemical Substructures) is a set of rules for automated molecular decomposition into fragments and for subsequent recombination of these fragments [6]. BRICS uses 16 distinct chemical environments, each represented by a different link (or dummy) atom [6]. A link atom is a placeholder atom used to specify recombination connection points, as illustrated in Figure 3. These link atoms set the connection rules. They ensure that during recombination, only chemically compatible environments fuse [6].

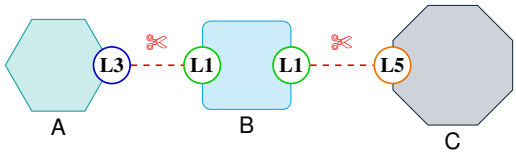

Figure 3: BRICS decomposition of two bonds. The bond between fragment A and B is replaced with link atoms L_3 and L_1. The bond between fragment B and C is replaced with link atoms L_1 and L_5

BRICS's embedded chemical intelligence makes it valuable for computer-assisted synthesis planning, as demonstrated by the FragmentRetro algorithm [14]. FragmentRetro is a bottom-up retrosynthetic search method that uses BRICS [6] and its revision r-BRICS [15] to decompose a target molecule into elementary fragments, which are then grouped into larger combinations and checked against an inventory of commercially available building blocks [14].

2.5 Molecular Similarity

Some automated chemical synthesis models [7, 8] use molecular similarity [16], which assesses the structural resemblance between two molecules to navigate the chemical reaction search space effectively. This is achieved by comparing fixed-length binary feature-vector molecular representations, in which the bits denote the presence or absence of specific structural motifs, known as fingerprints.

Works by Coley et al. [7] and Guo and Schwaller [8] specifically use Morgan circular fingerprints. A Morgan circular fingerprint is a molecular representation that captures chemical features of a molecule by enumerating submolecular neighbourhoods of each atom, limited in size by a maximum radius [7].

Fingerprints of two molecules can be compared for similarity using a mathematical metric. One such metric is the Tanimoto coefficient, used by Coley et al. [7] and Guo and Schwaller [8]. The Tanimoto coefficient (also known as the Jaccard index), shown in Equation 1, is a statistical index that quantifies the overlap of enabled bits between two binary feature vectors. It yields a continuous similarity score ranging from 0.0 (no overlap) to 1.0 (an exact match) [7, 16]. The Tanimoto coefficient is independent of the disabled bits, making it well-suited for evaluating sparse chemical fingerprints, where most possible features are naturally absent from any given small molecule [16].

$$Tanimoto(A, B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \quad (1)$$

Coley et al. utilise molecular similarity within a deterministic, data-driven retrosynthesis framework, evaluating product similarity against a reaction knowledge base to identify precedents whose products resemble the target molecule. Reaction templates from these precedents are then applied to the target to generate candidate precursors, which are ranked by their similarity to precedent reactants — highlighting a focus on database-derived templates rather than generative design [7].

Guo and Schwaller use molecular similarity in a reinforcement learning framework, applying the Tanimoto Group Over-

lap (TANGO) reward to steer forward generative design of molecules toward given building blocks. TANGO evaluates intermediates by Tanimoto similarity to building blocks, equally weighted with a Fuzzy Matching Substructure score that measures the number of overlapping heavy atoms [8].

3 Methods

To address computational bottlenecks in the baseline program synthesis CRN discovery framework described by Wijers [5], this section introduces two proposed substructure-aware heuristics. In addition, it presents an extension for processing and rebalancing partial reactions within the framework.

3.1 BRICS Fragments in Molecule Synthesis

To enable the synthesis of large molecules, the molecular synthesis step was enhanced using BRICS. This approach compresses complex molecular substructures into single nodes within the AST. The methodology consists of three stages: fragment extraction, grammar translation, and constraint adaptation.

Fragment Extraction and Normalisation

The BRICS fragments are extracted from the observed molecules using the RDKit [17] library. Compared with the original BRICS specification [6], the RDKit implementation applies slightly modified connection rules, most notably omitting the `L2` chemical environment, thereby leaving 15 environments [18].

To ensure compatibility with the aforementioned explicit SMILES requirements of the synthesiser's base grammar [5], the extracted fragments are further parsed. Fragments are also kekulised (i.e., aromatic bonds are explicitly converted to single and double bonds), which ensures the placement of connections to cyclic structures can be explicitly defined to avoid formation of invalid molecules.

Grammar-Based Fragment Representation

Algorithm 1 Simplified BRICS fragment molecule grammar

molecule	::=	*startingFragment*
chain	::=	*structure fixedBond fragmentXEntry*
fragmentXEntry	::=	*"chunk" fragmentYExit "chunk" ...*
fragmentYExit	::=	*specialBond fragmentZEntry*
	|	*fixedBond digit*
	|	*fixedBond chain*

After parsing, the resulting BRICS fragments are further translated into CFG rules. Each BRICS fragment in SMILES notation always starts with a dummy node representing one of the fragment's connection points, which is stripped together with its bond and subsequently acts as an identifier for the entire fragment type and is encoded as `X` in a fragment's rule name: `fragmentXEntry`. Every dummy atom, except for the aforementioned first one, is replaced by a `fragmentYExit` rule, where `Y` corresponds to the chemical environment number. These exit rules enforce a fixed bond type: a double bond for the `L7` group and a single bond for the others. If allowed by RDKit's BRICS connection rules [18], an exit rule connects to another fragment via a `fragmentXEntry` rule (Rule 4 in Algorithm 1). The remainder of the SMILES string for a fragment is represented as fixed hardcoded strings, referred to as *chunks* (see Rule 3 in Algorithm 1). Once a full molecule is synthesised, these chunks require no further interpretation, as they form valid partial SMILES strings. As demonstrated in Figure 4, complex molecular structures are compressed into single nodes represented by a single grammar rule (`fragmentXEntry`), with children denoting the BRICS connection points.

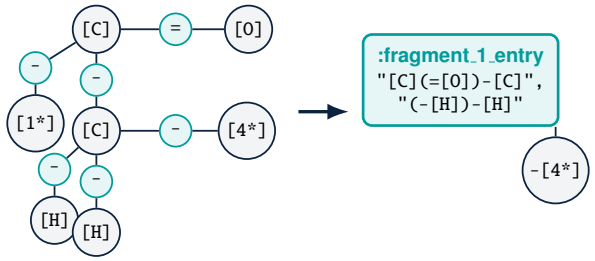

Figure 4: Representation of a BRICS fragment in the baseline molecular grammar on the left, and the proposed compressed grammar on the right.

Although this representation provides an elegant handling of fragment reconnections, introducing a new fragment (ignoring the base grammar) requires the presence of an existing one. Therefore, all obtained fragments are also introduced as additional `startingFragment` rules (Rule 1 in Algorithm 1). For these rules, the first dummy atom is not stripped but instead moved after the first atom in the SMILES notation and converted into an exit rule.

The BRICS fragments can also connect to the base grammar, enabling the synthesis of novel molecules containing these fragments, rather than solely their recombinations. Additional rules (Rules 5 and 6 in Algorithm 1) permit connections to either a ring digit or a general base grammar structure (denoted as `chain`) via fixed bond types. Conversely, the base grammar structures can also connect to fragment entry points (Rule 2 in Algorithm 1).

Since `Herb.jl` [12] by default explores grammar rules in the order they are defined during enumeration, the extended fragment grammar rules are positioned before the base grammar rules. This arrangement naturally introduces a heuristic that prioritises the synthesis of molecules containing the fragments.

Constraint Adaptation for Chemically Valid Recombination

Integrating BRICS fragment rules into the base grammar required extending the molecular grammar constraints to prevent the formation of chemically invalid structures. Two adaptations were implemented to this end. First, since the fragments connect with fixed bond types, they are treated as special types of bonds, which restrict the valency domain of atoms that can connect with them. Second, to prevent the formation of ring bonds between two atoms already connected by a ring bond within a fragment, matching trailing ring bonds in chunks are collected to forbid the formation of duplicate ring bonds in the exit nodes following two already connected chunks.

Additionally, the formation of self-loops on trailing atoms is strictly forbidden. The fragments are fully parsed again when posting the constraint on ringbond construction to track cases where chunks' trailing atoms are already directly connected to forbid invalid formation of ring bonds between themselves and between the neighbouring atoms from the base grammar.

Finally, inclusion of a "special bond" (a special case of `fixedBond`) in Rule 4 in Algorithm 1 is explained in Appendix B.

3.2 Molecule Similarity Guidance

Once the molecule synthesis phase finishes, the reaction synthesiser provides an option to sort the available molecules in decreasing order of maximum similarity score for each observed molecule. The score is based on the Tanimoto coefficient between two Morgan fingerprints with a radius of 2 and a 1024-bit size calculated with RDKit [17]. A stable sort is used to ensure reproducibility and retain the insertion order of the synthesised molecules. As in the baseline, the known molecules are added to the reaction grammar first; the remaining candidates are then inserted in decreasing order of similarity, so that reactions involving more structurally similar molecules are explored first. As mentioned before, the molecule synthesiser naturally includes an analogous heuristic that prioritises BRICS fragment rules over the base grammar.

Furthermore, the network synthesiser supports sorting the generated reactions by their decreasing combined similarity of the included molecules before adding them to the grammar. For each candidate reaction, the individual similarity scores of all participating molecules are summed. To prevent the heuristic from artificially promoting reactions simply because they contain more species, the aggregated score is normalised by the total number of molecules in the reaction.

3.3 Completing Partial Reactions

The reaction grammar was extended with support for partial reaction data. If a partial reaction is provided, its input and output molecules are guaranteed to be included in any synthesised reaction. The missing molecules on both sides are filled using the already existing grammar rules. Furthermore, the atom counts of the prefilled molecules are also considered in the already established constraint enforcing formation of only atom balanced reactions [5].

Finally, to support parsing and solving the reaction rebalancing tasks in the SynRXN datasets [2], the required atoms and ions, along with their valency limits (calculated with RDKit [17]), are extracted from the problems and added to the molecule grammar. To complete the support for balancing the reactions with ions, the constraints were also extended to ensure the synthesised reactions are charge-balanced. This is handled by treating charge as a special case of atoms, in which opposite charge can also reduce the "atom" count.

4 Experimental Setup

To evaluate the performance and scalability of the extended program synthesis framework, two benchmark suites were designed.

All benchmarks were run on a Lenovo ThinkBook 14s Yoga ITL with an Intel Core i5-1135G7 and 16 GB of DDR4 memory at 3200 MHz. All the code contributions and benchmarks are available at: github.com/Herb-AI/CRNSynthesizer/tree/substructures and at: zenodo.org/records/20751522

4.1 Reaction Rebalancing

The first benchmark suite evaluates the synthesiser's performance on the reaction rebalancing task, which requires identifying missing chemical species to complete and balance partial reactions. This evaluation utilises test sets from the v1.0.0 GitHub release[1] of SynRXN [2]. The framework was tested on four categories of incomplete reactions[2]: the first 500 records of the `complex` dataset (missing large organic compounds), the `mnc` dataset (missing non-carbon species), and the `mos` dataset (where larger species are missing on one side, but small molecules like water can also be missing on the other side), as well as all 491 available records of the `mbs` dataset (species missing on both sides).

The evaluation process was divided into two distinct phases, beginning with the missing-molecule synthesis subproblem. This problem is considered solved if the synthesiser generates all target species within the first 250 candidates. This cap was implemented because the reaction synthesiser generates all possible molecule subset splits and stores them in memory up front; exceeding this limit would surpass the 7 GB memory constraint enforced by the benchmark machine's resource pool. During this phase, several metrics were recorded to evaluate performance: the success rate of synthesising all missing species within the first 250 candidates, the distribution of the number of candidate molecules synthesised until all targets are found for successful problems, the runtime distribution for successful molecule synthesis problems, and the size distributions of non-synthesised molecules.

Only full reaction candidates can be evaluated. Therefore, the synthesiser does not know when it has found all missing molecules. As a result, all 250 discovered species are added to the reaction grammar.

In the second phase, successful subproblems proceed to reaction synthesis, where the framework is evaluated on its ability to identify the target balanced reaction within a maximum limit of 30,000 candidates. For this stage, the recorded metrics include: the success rate of synthesising the target reaction within the first 30,000 candidates, the number of candidate reactions synthesised before finding the target for successful problems, and the runtime distribution for reaction synthesis. Finally, all results are compared among configurations using only the base-molecule grammar, configurations enhanced with BRICS fragments from known molecules, and combinations that evaluate the inclusion of molecule sorting by Tanimoto similarity of Morgan2 fingerprints during reaction synthesis.

[1] https://github.com/TieuLongPhan/SynRXN/releases/tag/v1.0.0

[2] https://synrxn.readthedocs.io/en/latest/data_records.html#reaction-rebalancing

4.2 Molecule Similarity Guidance in CRN Discovery

The second benchmark suite also evaluates the molecule-similarity-guidance heuristic across the full CRN discovery process, unlike the previous benchmark, which focused only on the reaction rebalancing subproblem. An esterification reaction network is used as the benchmark target, originally proposed by Wijers [5], as illustrated in Figure 5. The goal is to reconstruct this two-step reaction pathway from incomplete observations.

$$\mathrm{CH_2O_2} + \mathrm{C_2H_6O} \longrightarrow \mathrm{C_3H_6O_2} + \mathrm{H_2O}$$
$$\mathrm{C_3H_6O_2} + \mathrm{CH_4O} \longrightarrow \mathrm{C_2H_4O_2} + \mathrm{C_2H_6O}$$

Figure 5: The target esterification Chemical Reaction Network. The synthesiser must identify the missing species (H_2O, CH_2O_2, CH_4O) and reconstruct both reactions.

The benchmark tracks the framework's performance across both the reaction synthesis and network synthesis stages. In the first reaction synthesis run, the first 250 synthesised molecule candidates are introduced into the grammar, just as in the first benchmark suite. In both stages, the molecule grammar is not enhanced with BRICS fragments.

For each synthesis stage, the process is executed twice: once utilising the Tanimoto similarity metric based on Morgan2 fingerprints, and once without any similarity guidance (relying solely on the baseline breadth-first search). To quantify the heuristic's efficiency, the number of candidate reactions synthesised before the target reactions are generated is recorded. Subsequently, the number of candidate networks assembled before the target network is discovered is also collected.

During the network synthesis stage, the molecule synthesiser and reaction synthesiser are configured to halt immediately upon generating the required target species and target reactions. This evaluation strategy, established by Wijers [5], prevents the network synthesiser's search space from becoming computationally intractable.

5 Results

In line with the evaluation frameworks established in the experimental setup, this section presents the performance results of the two substructure-aware heuristics across both benchmark suites.

5.1 Reaction Rebalancing Performance

First, the impact of introducing BRICS fragments into the molecular grammar is assessed by comparing their effects across two datasets for the missing-molecule synthesis subproblem in SynRXN [2] reaction rebalancing tasks. As shown in Figure 6, the BRICS-enhanced synthesiser achieves a 13.8 percentage-point improvement over the baseline on the `complex` dataset, demonstrating its superior performance on more complex cases. In contrast, for the `mnc` dataset, using a fragment grammar results in a 27.4 percentage-point decrease in solved problems relative to the baseline, clearly demonstrating underperformance in this context. Table 1 supports these findings: the baseline primarily fails on larger molecules in the `complex` dataset (92.5% of unsynthesised missing species have 7 or more atoms), while the BRICS approach fails on 52 fewer molecules, only 42.7% of which have 7 or more atoms, underscoring its comparative effectiveness with large molecules in that dataset.

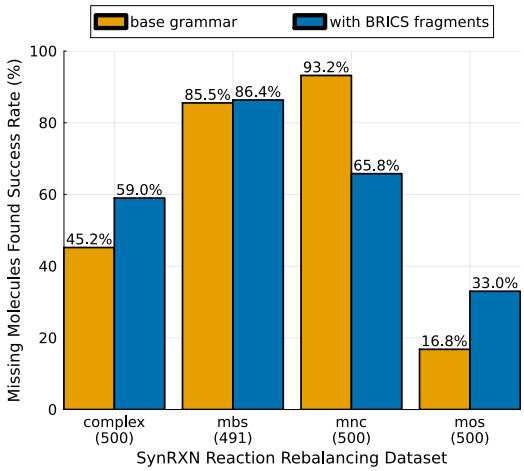

Figure 6: Success rates of synthesising all missing molecules within the first 250 candidates for SynRXN reaction rebalancing datasets.

Table 1: Distribution by atom count of candidate molecules not synthesised within the candidate limit at the molecule synthesis stage, across SynRXN reaction rebalancing datasets. Grammar abbreviations — Base: Base Molecule Grammar; +BRICS: Molecule Grammar with BRICS Fragments.

Dataset	Grammar	Unsynth. Mols.	Atom Count Distribution (%)			
			≤ 3	4–6	7–9	≥ 10
complex	Base	278	0.0	7.6	45.0	47.5
	+BRICS	220	40.0	17.3	7.7	35.0
mbs	Base	71	0.0	0.0	98.6	1.4
	+BRICS	104	74.0	2.9	23.1	0.0
mnc	Base	46	0.0	84.8	0.0	15.2
	+BRICS	190	72.6	23.7	0.0	3.7
mos	Base	416	0.0	3.8	34.1	62.0
	+BRICS	402	59.2	3.2	0.5	37.1

Unlike its improvement in `complex`, the BRICS-enhanced synthesiser performs worse on `mnc`, solving 27.4 percentage points fewer species-discovery subproblems than the original method (see Figure 6). Table 1 clarifies this: the BRICS grammar in `mnc` fails on 144 more molecules than the baseline, with 72.6% of unsynthesised species having no more than 3 atoms, highlighting a relative weakness for smaller molecules and a benefit for larger ones in `complex`.

Although the BFS iterator explores programs with fewer grammar rules first, the heuristic prioritising molecules with fragments still causes the synthesiser to examine these before targeting small structures without such fragments. This leads the fragment-enhanced synthesiser to require significantly more steps to discover missing small molecules common

in the mbs and mnc datasets, as shown in Figure 7. Because the synthesis stage is capped at 250 candidates due to reaction synthesiser limitations, small molecules are often not found within this limit.

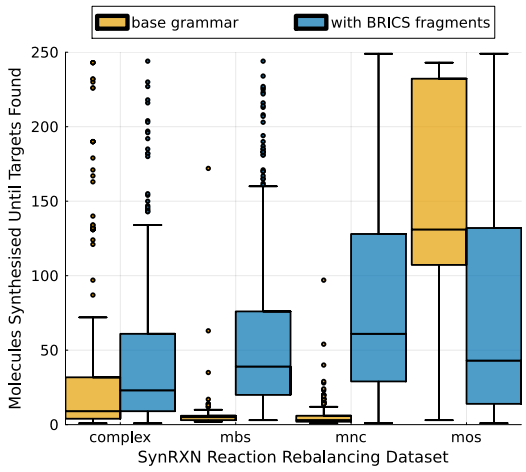

Figure 7: Distributions of the number of molecules synthesised before finding all missing molecules across successfully solved subproblems from reaction rebalancing SynRXN datasets.

As shown in Figure 7, for the mbs dataset, the BRICS fragment synthesiser achieves only a 0.9 percentage point improvement over the baseline in solving missing-species subproblems, indicating a minimal comparative gain. Table 1 provides further comparison: the enhanced synthesiser fails on 33 more molecules than the baseline, 76.9% of which have at most 6 atoms, while all molecules missed by the baseline have at least 7 atoms. This demonstrates a disadvantage of the fragment method for smaller molecules but also shows it reaches different targets, suggesting potential complementarity. Combining the first 175 BRICS candidates with 75 from the baseline may improve results, since most problems are likely solved within these sets.

Conversely, results from the mos dataset show that using a complementary approach may not always yield superior performance. As shown in Figure 7, for problems that are successfully solved, the BRICS synthesiser finds all targets using fewer candidates compared to the baseline, highlighting a specific advantage in candidate efficiency. However, Table 1 indicates that 59.2% of molecules missed by the enhanced approach are still small (at most 3 atoms), underlining a comparative weakness for smaller species relative to the baseline.

Success rates for finding target-balanced reactions follow the trends seen for missing species subproblems (Figure 8), but fewer reaction rebalancing tasks are solved, as finding the missing molecules does not guarantee the target will be among the first 30,000 candidates. The biggest drop in success rates is for the complex dataset. This means there are significantly more possible balanced reactions that can be synthesised for this set than for the other three, which leads the reaction synthesiser to reach the 30,000 limit more frequently.

Figure 8 shows that the Tanimoto similarity heuristic does not improve identification of balanced target reactions before

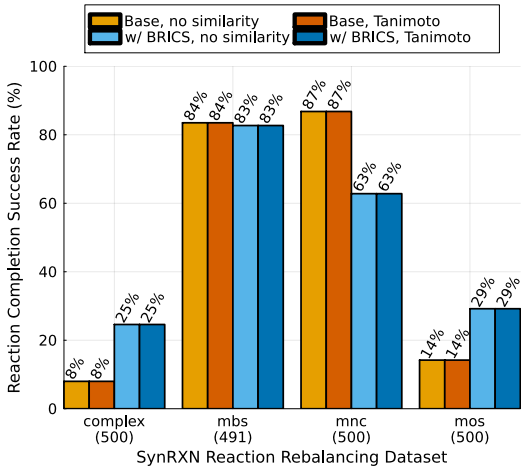

Figure 8: Reaction rebalancing success rates for the SynRXN reaction rebalancing datasets.

the 30,000 limit, as success rates remain unchanged. Furthermore, Figure 9 demonstrates that the heuristic sometimes increases the number of reactions needed to reach the target. This suggests that missing molecules in SynRXN [2] tasks are structurally dissimilar to known species, leading the heuristic to prioritise reactions with larger, more similar molecules, even when smaller ones are required.

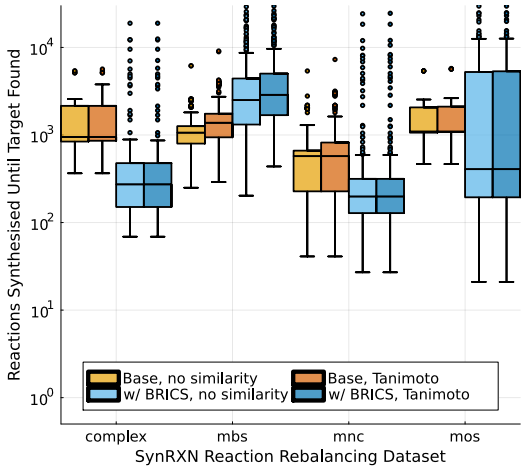

Figure 9: Distributions of the number of reactions synthesised until the target reaction is found across successfully solved SynRXN reaction rebalancing datasets.

Even for successful problems, the synthesiser must explore thousands of candidate reactions before the goal is found (Figure 9), making manual evaluation infeasible and automated evaluation computationally expensive. Thus, the molecule synthesis step needs further improvement to limit the search space, and the reaction synthesiser could be optimised to focus on more plausible candidates.

Finally, the runtime distributions of reaction and molecule synthesis for respective successful problems are presented in Appendix A.

5.2 Impact of Molecule Similarity Guidance on CRN Discovery

Although molecular similarity did not improve the synthesiser's efficiency on the SynRXN [2] reaction rebalancing tasks, it did yield promising improvements in the example esterification CRN synthesis problem introduced earlier (see Figure 5). When solving the reaction synthesis subproblem using the first 250 synthesised candidate molecules (as shown in Table 2), the synthesiser with Tanimoto similarity based on Morgan2 fingerprints explores 2,565 fewer reactions before synthesising the two targets. Furthermore, as shown in Table 2, the full CRN synthesiser with early halting of both molecule and reaction synthesis stages explores 26 fewer reactions. More significantly, it explores 409 fewer networks before reaching the goal.

Table 2: Number of candidate reactions synthesised before finding the target reactions within the first 250 candidate molecules, and the number of networks synthesised before identifying the example esterification network with early synthesiser halting.

Max Stage	Similarity Guidance	Synthesised Until Target	
		Reactions	Network
Reactions	None	22196	N/A
	Tanimoto w/ Morgan2	19631	N/A
Networks	None	144	435
	Tanimoto w/ Morgan2	118	26

The molecular similarity heuristic prioritises the inclusion of the missing formic acid (CH_2O_2) and methanol (CH_4O) due to their non-zero similarity scores: 0.25 and 0.29, respectively. Although the missing water (H_2O) receives a zero similarity score, it remains near the top of the list of non-similar molecules. This is because stable sorting is used, and water is synthesised early due to its small size. Reactions containing known and similar molecules are also considered earlier in the network synthesis stage. As a result, fewer CRN candidates are explored before the goal is found.

6 Responsible Research

To ensure the transparency, integrity, and safe application of the proposed framework, this section outlines the responsible research practices adhered to throughout this study.

6.1 Reproducibility

The reproducibility of the results was improved compared with the prototype developed by Wijers [5]. All the instances of standard Julia dictionary and set implementations were replaced with equivalent data structures that preserve insertion order when iterating over their contents. This approach removed a source of non-determinism that affected the order of synthesised candidate programs between runs.

Instead of enforcing a runtime limit on the reaction synthesis stage, a limit of 30,000 candidates was set. This method limits the maximum runtime and the number of reactions evaluated, while keeping results reproducible.

Furthermore, all code improvements and benchmarks developed in this study are available on GitHub[3] and Zenodo[4]. In addition, each benchmark run on the SynRXN [2] datasets is tagged with metadata, including the SynRXN version and source used, the commit SHA corresponding to the synthesiser code version used to execute the benchmarks, and the task type and dataset associated with each run. Similarly, each run of the esterification benchmark is also tagged with a commit SHA. Finally, the full results, figures, tables, and raw CSVs are published in the aforementioned repository, allowing all data elements to be compared.

6.2 Limitations and Ethical Implications

The CRN synthesiser is designed to support beneficial research (e.g., drug discovery) by enabling the automatic synthesis of potentially novel molecules. Nonetheless, it may theoretically be misused by malicious actors to identify chemical reaction networks that could lead to the synthesis of illicit drugs, explosives, or chemical weapons.

Still, the tool proposes only a theoretical reaction network, without providing laboratory instructions for carrying it out. Therefore, domain knowledge is still required to use these results.

Furthermore, the automatically synthesised reactions might not have been studied before and could lead to unpredictable side effects. As a result, they should be verified by experts before being attempted in laboratories.

However, the tool still suffers from performance limitations, rendering many chemical queries intractable. The synthesiser requires further effort, particularly during the reaction synthesis stage, to make it more useful to researchers.

6.3 LLM Usage

Agentic LLMs were used to help identify issues in the code, create code for generating plots and CSVs after benchmark runs, and enhance the structure of benchmark code for greater legibility. All outputs were manually verified, and the author takes full responsibility for the code's quality.

Furthermore, NotebookLM was used to suggest outlines for the paper's drafts, generate code for the included LaTeX figures, and support adherence to the academic writing style. Given its citation features, all information was manually verified, further reviewed against the papers' actual contents, and properly referenced. Grammarly was used to improve the text's grammar, spelling and legibility. Finally, none of the models declares the use of shared data for training and all appropriate toggles specifying this use were switched off.

7 Conclusions and Future Work

This study examined the effect of adding high-level building blocks from known molecules to Wijers' [5] grammar-driven program synthesis for automated discovery of chemical reaction networks. It also evaluated how introducing molecular similarity scoring affects the number of candidate reactions and networks generated prior to target discovery.

[3] github.com/Herb-AI/CRNSynthesizer/tree/substructures

[4] https://zenodo.org/records/20751522

The results gathered from the benchmark evaluating the synthesiser's performance on the reaction rebalancing tasks from the curated SynRXN [2] test sets show that, thanks to the introduction of BRICS [6] fragments from known molecules, the synthesiser is able to discover complex missing fragments more quickly than the baseline atom-by-atom approach. Large substructures are efficiently represented using just a single grammar rule. However, due to the increased branching factor in the AST search space caused by the larger number of possible rules, combined with a heuristic that prioritises molecules containing fragments, the synthesiser with enhanced grammar fails to find small missing molecules that do not share substructures early on.

The results suggest that a version of the molecule synthesiser that uses the fragment-enhanced and baseline grammars separately, then combines the two pools of candidate species, could yield better results than using only one grammar. Running the two synthesisers in parallel could also be more efficient in terms of runtime.

In addition, the use of BRICS fragments in the molecular grammar establishes a framework for integrating other fragmentation algorithms, such as r-BRICS [15]. It should be noted that BRICS does not fragment ring structures, so future work will need to address cycle recombination.

Sorting candidate molecules and reactions by maximum Tanimoto similarity based on Morgan2 fingerprints for each known molecule yielded promising improvements in an example esterification network synthesis task. In the reaction synthesis step, the heuristic prioritises the inclusion of the missing formic acid and methanol, which are structurally similar to the known molecules. The improvement from similarity guidance is even more substantial at the network synthesis stage, where reactions involving observed molecules and similar species are considered as candidates for networks first.

On the other hand, the results show that molecular similarity slightly increases the number of reactions synthesised, up to the target for the reaction rebalancing tasks in the SynRXN [2] datasets. This suggests that, in general, incomplete reactions miss molecules that differ structurally from known ones, so a better approach is needed for this task. For example, the reaction synthesiser could use the remaining atom balance during reaction synthesis to guide the molecule synthesiser. Prioritising the search for molecules containing the missing atoms required to balance the reaction could enable earlier exploration of target structures.

The evidence for the impact of molecule-similarity guidance on the synthesiser is mixed. This suggests the heuristic may be valuable for retrosynthesis, as noted by Coley et al. [7], and potentially useful for CRN synthesis when information is limited, such as in less-complete networks compared to reactions in SynRXN datasets [2]. However, more data is needed to fully quantify the effect of molecular similarity on CRN synthesis efficiency.

Due to limitations of the reaction synthesiser, synthesising networks remains intractable, necessitating early halting of molecule and reaction synthesis once the respective targets are found. The reaction synthesiser should be further improved to reduce the number of candidates it produces. For example, an extra stage could be added before the network synthesiser to filter candidate reactions based on chemical plausibility.

A preliminary analysis of an alternative molecule synthesiser employing a bottom-up approach with observational equivalence has been conducted and is detailed in Appendix C.

A Reaction Rebalancing Runtime

The runtime distributions of molecule and reaction synthesis for respective successful problems are closely related to how many candidates are explored before reaching the goal, as shown in Figures 10 and 11. Most molecule synthesis subproblems were solved under 2 seconds, but the reaction synthesis step could take up to 50 seconds. These distributions show only runtimes for solved problems. Unsuccessful reaction synthesis runs often take even longer than 50 seconds, since they fully exhaust the 30,000-candidate limit. For the full benchmark, only the first 500 problems from each SynRXN dataset were selected, except for the `mbs` dataset, which was analysed using all 491 problems. The full benchmark on the aforementioned laptop took over 22 hours to complete. These results further underline the need to optimise the synthesiser for the reaction rebalancing step.

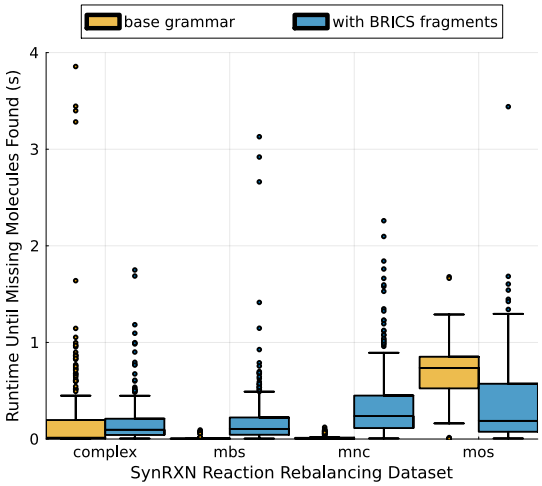

Figure 10: Runtime distributions for the missing molecule synthesis subproblem from SynRXN reaction rebalancing datasets across successful runs.

B Avoiding structural symmetries with a "special bond"

Because `Herb.jl` iterates over structurally equivalent rules within a `UniformTree` [12], Rule 4 in Algorithm 1 utilises a "special bond" that evaluates to the fixed bond type used by that fragment. This ensures Rules 4 and 5 from Algorithm 1 do not share a uniform shape. Without this distinction, replacing a Rule 4 node with Rule 5 would introduce a new ring bond without the `Uniform Ringbond` constraint registering its location, as it only tracks the location of base grammar ring bonds in the AST when a new `UniformTree` is formed. By restricting the shape, the `Uniform Ringbond` constraint is posted only when necessary, maintaining efficient constraint

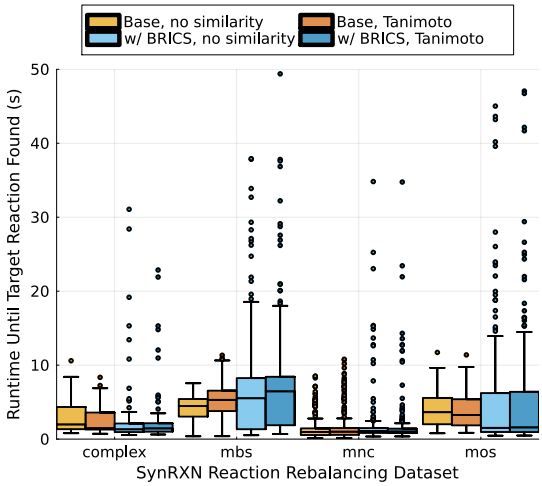

Figure 11: Runtime distributions of reaction synthesis for SynRXN reaction rebalancing datasets among successful runs.

propagation. The recorded locations are then only required to be updated whenever a fragment rule is replaced, as two fragments might possess the same number of connections but different internal chunks.

C Bottom-up Molecule Synthesis

A basic demo of a molecule bottom-up synthesiser was created using a tweaked base grammar (without BRICS fragments) that allows the representation of a molecule substructure in which only the root atom does not meet its valency constraints. In this system, the root atom and ring bonds are connected to dummy nodes. As a result, this representation of fragments can be parsed with RDKit, which can canonicalise their SMILES strings. This canonicalisation step enables observational equivalence pruning. However, initial tests showed worse performance than the base top-down approach, most likely due to the lack of equivalent structural constraints. For example, the remaining valence of root atoms of fragments could be embedded directly into the grammar to prevent synthesis of molecules violating valency limits. Currently, the demo uses only RDKit as the validator to determine whether the synthesised molecules are valid. The code is available on GitHub: github.com/Herb-AI/CRNSynthesizer/tree/bottom-up

References

[1] Jan P. Unsleber and Markus Reiher. 'The Exploration of Chemical Reaction Networks'. In: *Annual Review of Physical Chemistry* 71.Volume 71, 2020 (2020), pp. 121–142. ISSN: 1545-1593. DOI: https://doi.org/10.1146/annurev-physchem-071119-040123. URL: https://www.annualreviews.org/content/journals/10.1146/annurev-physchem-071119-040123.

[2] Tieu-Long Phan, Nhu-Ngoc Nguyen Song and Peter F. Stadler. 'SynRXN: An Open Benchmark and Curated Dataset for Computational Reaction Modeling'. en. In: *Scientific Data* 13.1 (Apr. 2026), p. 625. ISSN: 2052-4463. DOI: 10.1038/s41597-026-07260-w. URL: https://www.nature.com/articles/s41597-026-07260-w (visited on 31/05/2026).

[3] Mingjian Wen et al. 'Chemical reaction networks and opportunities for machine learning'. en. In: *Nature Computational Science* 3.1 (Jan. 2023), pp. 12–24. ISSN: 2662-8457. DOI: 10.1038/s43588-022-00369-z. URL: https://www.nature.com/articles/s43588-022-00369-z (visited on 23/04/2026).

[4] Luca Cardelli et al. 'Syntax-Guided Optimal Synthesis for Chemical Reaction Networks'. In: *Computer Aided Verification*. Ed. by Rupak Majumdar and Viktor Kunčak. Cham: Springer International Publishing, 2017, pp. 375–395. ISBN: 978-3-319-63390-9. DOI: 10.1007/978-3-319-63390-9_20.

[5] Richard Wijers. 'Automated Discovery of Chemical Reaction Networks using Program Synthesis'. MA thesis. Delft University of Technology, 2025. URL: https://resolver.tudelft.nl/uuid:f70ee49d-d0e6-4938-af11-7c6770a38502.

[6] Jörg Degen et al. 'On the Art of Compiling and Using 'Drug-Like' Chemical Fragment Spaces'. In: *ChemMedChem* 3.10 (2008), pp. 1503–1507. DOI: https://doi.org/10.1002/cmdc.200800178. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cmdc.200800178. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmdc.200800178.

[7] Connor W. Coley et al. 'Computer-Assisted Retrosynthesis Based on Molecular Similarity'. In: *ACS Central Science* 3.12 (2017). PMID: 29296663, pp. 1237–1245. DOI: 10.1021/acscentsci.7b00355. eprint: https://doi.org/10.1021/acscentsci.7b00355. URL: https://doi.org/10.1021/acscentsci.7b00355.

[8] Jeff Guo and Philippe Schwaller. 'TANGO: direct optimization of constrained synthesizability for generative molecular design'. en. In: *Nature Computational Science* 6.3 (Mar. 2026), pp. 260–270. ISSN: 2662-8457. DOI: 10.1038/s43588-026-00959-1. URL: https://www.nature.com/articles/s43588-026-00959-1 (visited on 21/05/2026).

[9] David Weininger. 'SMILES, a chemical language and information system. 1. Introduction to methodology and encoding rules'. In: *Journal of Chemical Information and Computer Sciences* 28.1 (1988), pp. 31–36. DOI: 10.1021/ci00057a005. eprint: https://doi.org/10.1021/ci00057a005. URL: https://doi.org/10.1021/ci00057a005.

[10] David Weininger, Arthur Weininger and Joseph L. Weininger. 'SMILES. 2. Algorithm for generation of unique SMILES notation'. In: *Journal of Chemical Information and Computer Sciences* 29.2 (1989), pp. 97–101. DOI: 10.1021/ci00062a008. eprint: https://doi.org/

10.1021/ci00062a008. URL: https://doi.org/10.1021/ci00062a008.

[11] Sumit Gulwani, Oleksandr Polozov and Rishabh Singh. 'Program Synthesis'. In: *Found. Trends Program. Lang.* 4.1–2 (July 2017), pp. 1–119. ISSN: 2325-1107. DOI: 10.1561/2500000010. URL: https://doi.org/10.1561/2500000010.

[12] Tilman Hinnerichs et al. *Herb.jl: A Unifying Program Synthesis Library*. 2026. arXiv: 2510.09726 [cs.PL]. URL: https://arxiv.org/abs/2510.09726.

[13] Tilman Hinnerichs et al. *Modelling Program Spaces in Program Synthesis with Constraints*. 2025. arXiv: 2508.00005 [cs.PL]. URL: https://arxiv.org/abs/2508.00005.

[14] Yu Shee et al. 'FragmentRetro: A Quadratic Retrosynthetic Method Based on Fragmentation Algorithms'. In: *Journal of Chemical Theory and Computation* 22.2 (2026). PMID: 41491661, pp. 972–980. DOI: 10.1021/acs.jctc.5c01632. eprint: https://doi.org/10.1021/acs.jctc.5c01632. URL: https://doi.org/10.1021/acs.jctc.5c01632.

[15] Leili Zhang, Vasumitra Rao and Wendy Cornell. 'r-BRICS – A Revised BRICS Module That Breaks Ring Structures and Carbon Chains'. In: *ChemMedChem* 19.4 (2024), e202300202. DOI: https://doi.org/10.1002/cmdc.202300202. eprint: https://chemistry-europe.onlinelibrary.wiley.com/doi/pdf/10.1002/cmdc.202300202. URL: https://chemistry-europe.onlinelibrary.wiley.com/doi/abs/10.1002/cmdc.202300202.

[16] *Daylight Theory: Fingerprints*. URL: https://www.daylight.com/dayhtml/doc/theory/theory.finger.html (visited on 06/06/2026).

[17] *RDKit: Open-source cheminformatics*. DOI: 10.5281/zenodo.16996017. URL: https://www.rdkit.org/.

[18] Greg Landrum. *A BRICS tutorial – RDKit blog*. en. Aug. 2025. URL: https://greglandrum.github.io/rdkit-blog/posts/2025-08-15-BRICS-tutorial.html#brics-basics (visited on 07/06/2026).