Multistep Synthesis Tool for Organic Chemistry Education

Raelyn Brooks

October 23, 2025

1 Introduction

Organic synthesis—the design of reaction sequences to build complex molecules from simpler precursors—remains one of the most intellectually demanding areas of chemistry. Unlike single-step reactions that can be memorized and applied in isolation, multistep synthesis requires strategic planning, backward reasoning, and the coordination of many interacting chemical transformations. For students, this often feels like solving a "black box" puzzle: they understand individual reactions but struggle to combine them into coherent pathways. National failure rates for organic chemistry courses frequently exceed 50 percent [5], underscoring the widespread difficulty of mastering this skill.

In both research and education, the traditional static depiction of reactions in textbooks fails to reflect the underlying algorithmic logic of multistep synthesis. Designing a synthetic route involves recursive problem solving (retrosynthesis), constraint management (functional group compatibility and selectivity), and temporary state control (protecting groups and stereochemistry)—all of which resemble computational planning problems. However, existing instructional tools rarely make this algorithmic structure explicit, limiting learners' ability to see synthesis as a structured, rule-driven process.

This project addresses that gap by developing an interactive visualization pipeline that encodes reactions using the Smiles Reaction Knowledge Specification (SMIRKS) formalism and renders them using the RDKit cheminformatics library. SMIRKS provides a symbolic language for reaction rules, enabling computational parsing, pattern recognition, and database integration. This is based on Smiles Arbitrary Target Specification(SMARTS) notation, which encodes molecular structures as linear strings—facilitating algorithmic manipulation. RDKit then transforms these symbolic definitions into visual molecular diagrams, allowing entire multistep pathways to be represented as connected, interpretable reaction networks.

By leveraging these technologies, the project contributes to educational cheminformatics through a tool that:

1. Automates the rendering of multistep synthesis pathways from reaction datasets.

2. isualizes reaction logic in a way that emphasizes decision trees and constraint satisfaction, mirroring algorithmic reasoning.

3. Bridges chemical and computational representations, making synthesis more accessible for both chemists and computer scientists.

Ultimately, this work positions multistep synthesis not just as a chemical skill but as a computationally tractable process, paving the way for more effective teaching tools, database-driven synthesis planning, and algorithmic analysis in organic chemistry.

2 Background

In organic chemistry, multistep synthesis refers to the process of transforming simple molecular starting materials into complex target molecules through a series of discrete, logically ordered chemical reactions. Each reaction step modifies the structure of a molecule, typically by altering functional groups—specific arrangements of atoms that dictate chemical behavior. From a computer science perspective, these functional groups can be viewed as data types or states: just as an integer might be cast into a string or an array into a list, a hydroxyl group (–OH) might be converted into a halide (–Cl), or an alkene might be transformed into an alcohol. Each synthetic step is thus a state transition in a well-defined, though highly constrained, chemical state machine.

Designing a synthetic pathway is not unlike algorithm design, where chemists aim to find an efficient and feasible sequence of operations to reach a target state. Here, the "output" is a desired molecule, and the "input" is a set of commercially available or easily accessible starting materials. The multistep aspect arises because direct, one-step conversions are rarely possible; instead, chemists must plan a sequence of reactions, each altering the molecular structure in a controlled manner. This is analogous to chaining multiple functions in a program to transform an initial data structure into a final desired format.

A central algorithmic strategy in synthesis planning is retrosynthesis, introduced by E.J. Corey, organic chemist and 1990 Chemistry Nobel Prize Winner. Retrosynthesis involves working backward from the target molecule to break it down recursively into simpler precursors. This is strikingly similar to goal decomposition in AI planning or backtracking algorithms in search problems. Starting from the target, chemists iteratively identify strategic bonds to "disconnect," yielding simpler molecules that could plausibly be converted into the target in one step. This continues until reaching molecules that are known or easily obtainable starting materials. In computational terms, retrosynthesis resembles a reverse search through a tree or graph, where each node represents a molecular state and each edge represents a known chemical transformation.

During synthesis planning, selectivity and functional group compatibility act as constraints, analogous to conditional logic in code or constraint satisfaction problems (CSPs) in AI. [6] For example, a reaction designed to modify an alcohol group might also inadvertently react with a nearby amine group, producing unwanted side products. Chemists must therefore choose reaction sequences that respect these constraints, ensuring that each transformation occurs at the correct site and does not disrupt other parts of the molecule. This is akin to managing function side effects in programming—ensuring that an operation modifies only what it is intended to, without corrupting unrelated data.

An additional layer of complexity involves stereochemistry (the 3D arrangement of atoms) and protecting groups (temporary modifications used to “mask” reactive sites). These elements function like temporary state variables or conditional flags that preserve critical information during intermediate steps. For example, protecting groups are akin to placing certain variables in a “read-only” or “inactive” state to prevent unwanted changes until the program reaches the correct point in execution to “unprotect” them. Stereochemical relationships act like metadata that must be preserved across transformations, ensuring that the final molecule has the correct 3D orientation—just as a compiler must preserve type information across optimizations.

To manage these complex transformations systematically, chemists often model reactions in a linear data flow format:

$$\text{Reactants} + \text{Reagents} \rightarrow \text{Products}$$

This format mirrors data flow in functional programming or pipeline architectures in software engineering. Each step is a transformation function that takes molecular “inputs” and produces “outputs,” which can then feed into the next step. From a computational perspective, this linear structure offers several advantages:

1. **Traceability:** Each transformation can be logged and indexed, enabling the reconstruction of synthetic routes and error checking, similar to how logs are used in debugging.

2. **Database Integration:** Standardized formats like SMILES and SMIRKS allow reactions to be stored as structured data, facilitating search, retrieval, and machine learning. This mirrors how normalized database schemas support efficient querying.

3. **Modularity:** Each reaction step is a modular operation that can be reused across different pathways, analogous to reusable functions or classes in code.

4. **Visualization:** Linear reaction flows can be easily represented as directed acyclic graphs (DAGs), where nodes are molecular states and edges are transformations—making them ideal for cheminformatics pipelines and algorithmic reasoning.

In short, multistep synthesis can be understood as a sophisticated algorithmic problem: molecules represent structured data, functional groups represent types and states, reactions are transformation functions, and retrosynthesis is a backward search strategy under complex constraints. For computer scientists, this framing highlights why cheminformatics libraries like RDKit, symbolic languages like SMIRKS, and visualization pipelines are powerful—they translate the chemical logic into computational structures that can be analyzed, optimized, and taught using algorithmic principles.

3 Related Works

3.1 Reaction Representation and SMIRKS Formalism

The foundational step in computational reaction modeling is the translation of chemical transformations into a symbolic language interpretable by algorithms. The SMIRKS format, derived from SMILES, provides a flexible mechanism to encode both specific reactions and generalized transformation rules. Literature on chemical data representation emphasizes that these formats enable reaction pattern recognition, rule-based synthesis prediction, and large-scale database integration. Works in this domain highlight that robust SMIRKS parsing supports not only mechanistic modeling but also serves as a data layer for visualization systems and generative synthesis algorithms [8].

Works in this domain highlight that robust SMIRKS parsing supports not only mechanistic modeling but also serves as a data layer for visualization systems and generative synthesis algorithms. The annotated literature identifies several efforts to standardize chemical notation and link structural encoding with reaction prediction models, a crucial link to the project's educational synthesis visualization goals.

3.2 RDKit and Cheminformatics Frameworks

Among open-source cheminformatics platforms, RDKit is one of the most widely adopted due to its robust handling of molecular representations and transformations. Numerous studies in cheminformatics emphasize RDKit's versatility for tasks including substructure searching, molecular fingerprinting, and reaction enumeration. Researchers have particularly noted its extensibility for educational and visualization purposes, allowing developers to generate 2D molecular depictions, reaction schemes, and datasets for computational learning systems.
In this project, RDKit's molecule parsing (MolFromSmiles) and coordinate generation (Compute2DCoords) functions serve as the computational backbone for converting textual reaction definitions into visual chemical diagrams. By leveraging RDKit's drawing modules (Draw.MolToImage), the project extends the toolkit into a visual pedagogy tool—making chemical synthesis pathways interpretable even for non-specialist learners. This positions the work within a subfield of cheminformatics focused on interpretable visual analytics, bridging chemical informatics and visual communication.

3.3 Computational Visualization and Educational Tools

The use of visualization in computational chemistry extends beyond aesthetics; it plays a vital cognitive role in supporting chemical reasoning. Prior work on reaction visualization tools—ranging from commercial platforms like ChemDraw to research-oriented frameworks like Indigo and Open Babel—demonstrates how symbolic chemistry can be rendered graphically to assist comprehension and verification. However, many of these systems rely on manual input or lack automated support for multi-step reaction sequences.

By contrast, the system developed in this project automates pathway rendering from large reaction datasets. Each reaction, formatted as Reactants → Reagents → Products, is parsed and visually represented as a network of chemical structures connected by arrows annotated with reagent labels. This approach aligns with modern pedagogical chemistry tools that emphasize interactivity and visual logic to improve understanding of synthesis design. The use of PIL (Python Imaging Library) to dynamically compose RDKit-rendered molecules onto a unified canvas represents a practical integration of informatics and visual communication principles highlighted in the literature.

3.4 Data-Driven Synthesis Prediction and Multistep Modeling

Recent studies in reaction prediction, retrosynthesis, and synthesis planning have leveraged large datasets encoded in SMILES/SMIRKS to train machine learning models capable of generating plausible synthetic routes [2]. Although this project does not directly implement predictive modeling, it shares methodological continuity with such research by structuring reactions in a format compatible with machine learning frameworks [4]. Literature examining data-driven synthesis design emphasizes that visualization and explainability remain key challenges; thus, tools that render reaction logic interpretable, like this project's visualization pipeline, contribute to advancing both human understanding and computational transparency in chemistry.

Moreover, by including thousands of curated textbook and large-scale reactions, the dataset architecture supports potential future expansion into reaction pathway decision trees or knowledge graph representations, which several annotated studies identify as emerging directions in cheminformatics research [9].

4 Methodology

4.1 Introduction to Methodology

The design and implementation of the Multistep Synthesis Tool were guided by a core objective: to translate the complex, constraint-driven logic of organic synthesis into a computationally tractable and visually intuitive format. This section details the data architecture, the graph-based framework for synthesis planning, and the cheminformatics visualization pipeline developed to achieve this goal.

4.2 Data Acquisition and Structural Formatting

The foundation of the system is a comprehensive dataset of chemical transformations. The primary data source is a large SMILES dataset compiled by Rik van der Lingen, which encompasses 1.37 million reaction SMILES entries [10]. These entries, which include reactants, reagents, solvents, and products, were sourced predominantly from United States Patent and Trademark Office USPTO patent literature spanning from 1976 to 2024, with an additional 2.5 percent from academic literature. This extensive dataset was generated through custom parsing and conversion from an existing USPTO dataset, employing OSCAR for semantic parsing and ChatGPT LLM-based extraction for data extraction. Molecular entity identification was achieved using OPSIN and a custom synonym list. All SMILES strings were

verified to be RDKit-safe, and duplicates were removed to ensure data integrity.
To complement this large-scale data, a custom collection of textbook organic chemistry reactions was curated and formatted explicitly in the SMIRKS notation. The SMIRKS format is crucial for encoding chemical transformation rules, utilizing the convention: Reactants → Reagents → Products. Within this structure, the ">" symbol separates the reaction components, while the "." symbol is used to separate individual molecules within a component list. This symbolic language enables the computational system to parse reaction logic, a necessary step for the visualization pipeline and future expansion into prediction models.

4.3 Algorithmic Architecture for Synthesis Planning

The synthesis problem—finding a pathway from a starting material to a target molecule—was framed as a graph search problem in an effort to model the decision-tree logic inherent in chemical planning [3]. While the system's eventual goal aligns with a retrosynthesis approach, the forward synthesis pathway is initially constructed as a directed graph.

1. **Nodes:** Represent every molecules (compound) involved in the multistep synthesis scheme, including the initial starting material, all intermediate compounds, and the final target product.

2. **Edges:** Represent the application of reagents and/or reaction conditions that transform a molecule (Node A) into a subsequent molecule (Node B)

3. **Edge Weights:** The primary edge weight is the number of synthetic steps. Assigning a unit weight to each successful transformation allows the system, when fully implemented, to leverage algorithms like Dijkstra's algorithm to identify the shortest, or most efficient, patyway in terms of step count [1].

A key component of this architecture is the Tree Structure Implementation [6]. This structure uses nodes for molecules and edges for reagents/reaction conditions,effectively creating a binary decision framework (reagent present/absent). This design facilitates a highly efficient search, enabling an O(1) search time when the tree data is properly stored and queried. Once the target molecule is successfully reached through this forward search, the algorithm traverses backward (retrosynthetically) to construct and retrieve the complete pathway for the user.

4.4 Cheminformatics Visualization Pipeline

The entire system was built using Python, leveraging key libraries for cheminformatics and image generation. The development environment was maintained using Anaconda to ensure seamless integration with the RDKit environment [7].
The pipeline comprises two main operational scripts and a structured data organization system:

1. **multistepTemplate.py:** This core script is responsible for the actual rendering. It parses the SMIRKS notation by first splitting the reaction components using " > " and then individual molecules using ".". It uses the RDKit library for its robust handling

of molecular representations and transformations,specifically for interpreting SMIRKS notation and generating 2D molecular diagrams.

2. **generateFromDataset.py:** This script manages data access. It reads reaction SMILES and SMIRKS from the curated dataset folder, allowing for adjustable processing limits. It incorporates the tqdm library to provide runtime tracking and progress indication, crucial for managing the large data volume.

The RDKit output (molecular structure images) is then composed using the Python Imaging Library (PIL) to create a unified canvas that displays the reaction scheme in the familiar reaction format. All generated images (products, reactants, reactions, and complete multisteps) are stored in organized folders, facilitating rapid retrival for the interactive step-by-step user interface, which is designed with an input structure similar to RMechDB. The ultimate goal of this pipeline is to provide a transparent, visual logic for chemical processes, advancing cheminformatics through interpretable visual analytics.

5 Results

6 Conclusion

References

[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., and Stein, C. *Introduction to Algorithms, Third Edition*, 3rd ed. The MIT Press, 2009.

[2] Dai, H., Li, C., Coley, C. W., Dai, B., and Song, L. *Retrosynthesis prediction with conditional graph logic network.* Curran Associates Inc., Red Hook, NY, USA, 2019.

[3] Gale, E., Lobski, L., and Zanasi, F. A categorical model for organic chemistry. *Theor. Comput. Sci. 1032*, C (Apr. 2025).

[4] Hormazabal, R., Park, C., Lee, S., Han, S., Jo, Y., Lee, J., Jo, A., Kim, S., Choo, J., Lee, M., and Lee, H. Cede: a collection of expert-curated datasets with atom-level entity annotations for optical chemical structure recognition. In *Proceedings of the 36th International Conference on Neural Information Processing Systems* (Red Hook, NY, USA, 2022), NIPS '22, Curran Associates Inc.

[5] Mooring, S. R., Mitchell, C. E., and Burrows, N. L. Evaluation of a flipped, large-enrollment organic chemistry course on student attitude and achievement. *Journal of Chemical Education 93*, 12 (2016), 1972–1983. Published: December 13, 2016.

[6] O'Sullivan, B., Ferguson, A., and Freuder, E. C. Boosting constraint satisfaction using decision trees. In *Proceedings of the 16th IEEE International Conference on Tools with Artificial Intelligence* (USA, 2004), ICTAI '04, IEEE Computer Society, p. 646–651.

[7] RDKit Documentation Team. Rdkit overview. https://www.rdkit.org/docs/Overview.html, 2025. Accessed: 2025-10-22.

[8] Schilter, O. T., Laino, T., and Schwaller, P. Cmd+v for chemistry: Image to chemical structure conversion directly done in the clipboard. *Applied AI Letters 5*, 1 (Jan. 2024).

[9] Souza, V. F., Cicalese, F., Laber, E. S., and Molinaro, M. Decision trees with short explainable rules. *Theor. Comput. Sci. 1047*, C (Aug. 2025).

[10] van der Lingen, R. Reaction smiles crd 1.37m dataset. https://doi.org/10.6084/m9.figshare.28230053.v1, Jan. 2025. Dataset.