# The Chemputer and Chemputation:

# A Universal Chemical Compound Synthesis Machine

Leroy Cronin\*

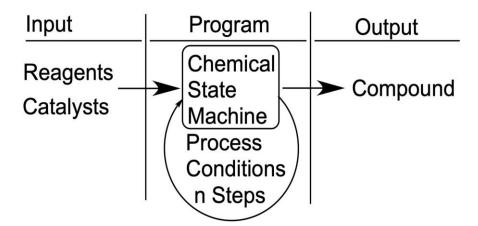
\*School of Chemistry, Advanced Research Centre, University of Glasgow, Glasgow, G11 6EW UK, www.croninlab.com email: lee.cronin@glasgow.ac.uk

#### **Abstract**

This work establishes a rigorous proof for the universality of the chemputer as a chemical synthesis machine, capable of constructing any stable and isolable molecule through a finite, expressible process. This process is governed by three key parameters: reagents, process conditions, and catalysts. Additionally, the study introduces dynamic error correction mechanisms integrated into each step of the synthesis pathway, ensuring real-time accuracy and reliability. The role of universally configurable hardware is also highlighted, with the introduction of a "chempiling" function that translates synthesis pathways into executable hardware configurations. These advancements collectively demonstrate the chemputer's capability to perform any feasible chemical synthesis, thereby establishing it as a universal tool in chemical manufacturing and synthesis. I show that every finitely realizable chemical synthesis process that can exist within the bounds of physical laws can be perfectly instantiated and executed by a universal chemputer, provided that the process can be completed within the finite number of reagent input vessels, reaction vessels, and product output vessels available, and that the error correction mechanisms are sufficiently robust to maintain the accuracy of the synthesis within these constraints. Finally, I show that chemical reactions are not implicit functions, but are an emergent property coming from the combination of the reagents, process conditions, and catalysts.

#### Introduction

Turing completeness is a concept from theoretical computer science that defines the ability of a computational system to perform any computation that can be done by a Turing machine [1, 2, 3]. For a system to be Turing complete, it must have the capability to simulate a Turing machine. This means it can execute any algorithm, given sufficient time and memory, and solve any problem that is computationally solvable. Turing completeness is a foundational concept in understanding the limits of what can be computed. In essence, if a programming language or computational system is Turing complete, it can, in theory, perform any computation that a computer can, assuming no constraints on resources like time and memory.



**Figure 1:** A schematic of the Chemical State Machine (CSM). The inputs are the Reagents (R), Catalysts(K) and the chemical program or  $\chi$ DL file[12] contains details of the process conditions and code to run the hardware. The output are the pure target compounds (C) as the chemical state machine includes a reactor, workup, isolate and purify system.

Expanding this concept to the realm of chemistry involves envisioning chemical systems that can perform computations in a way analogous to a Turing machine. Here we explore this idea where chemical reactions are used to undergo programmable transformations in a device we call a chemputer[4, 5, 6, 7]. The chemputer is designed to automate and control chemical reactions with high precision[8]. It uses a combination of hardware and software to carry out complex sequences of chemical processes[9]. By programming these sequences, the

chemputer can perform tasks that require conditional logic, loops, and the manipulation of data—key components of Turing completeness.

The concept of a chemputer as a universal chemical synthesis machine posits that it can instantiate any feasible chemical synthesis, see Figure 1. This document outlines the proof for the universality of the chemputer, demonstrating that it can synthesize any target compound within the chemical space defined by the provided parameters. To prove the universality of the chemputer, we need to demonstrate that it can conduct any feasible chemical synthesis. This involves showing that the transformation function  $\tau$  can account for all chemical reactions possible under the defined reagents, process conditions, and catalysts (it has been suggested that catalysts might themselves be viewed as a type of constructor [10]). Furthermore, we incorporate the mechanisms of dynamic error correction[11] during synthesis and the use of universally configurable hardware to support complex chemical processes through a chempiling function.

#### **Definitions**

- **Reagent Space** (*R*): A finite set of all possible chemical reagents, including all chemical elements and basic compounds.
- **Process Conditions** (*P*): A set of environmental parameters (e.g., temperature, pressure, solvent / gas conditions, energy input type) that influence the outcome of reactions.
- Catalysts (K): A set of substances that alter the reaction pathways or rates without being consumed in the process.
- Target Compounds (C): The set of desired products or output compounds.
- Universally Configurable Hardware (H): A hardware platform that can be dynamically reconfigured to execute various chemical synthesis processes. In the chemputer, the system is constrained by a finite number of reagent input vessels, reaction vessels, and product output vessels represent the number of available reagent input vessels, reaction vessels, and product output vessels,  $V_R$ ,  $V_P$ ,  $V_O$  respectively. This means that any chemical synthesis is realizable if it can be completed within these finite resources. The

configuration is represented as a graph G = (V, E), where: V is a set of nodes representing

hardware components (e.g., reactors, mixers, sensors) and E is a set of edges

representing connections between components, defining the flow of reagents, products,

energy, or information.

**Dynamic Error Correction (\delta)**: A mechanism embedded within each step of the synthesis

process, enabling real-time detection and correction of errors, ensuring the accuracy of each

transformation before proceeding to the next step.

**Chempiling Function** ( $\chi$ ): The process of translating a synthesis pathway  $\sigma$  into a corresponding

hardware configuration G(H) that can execute the synthesis process.

**Axioms** 

I introduce three axioms:

A1: Conservation of matter.

A2: Finite reaction time.

A3: Stability of elements found in *R* under standard conditions.

Lemmas

L1: For any  $c \in C$  there exists a finite sequence of transformations from Reagents R to c. Proof:

By the definition of *C* and finite reaction time axiom.

L2: For any transformation function  $t \in \tau$  can be decomposed into a finite sequence of

elementary reactions. Proof: By the nature of chemical reactions and the conservation of

matter.

4

**Assumptions** 

1. Existence of a Universal Set-Up: This demonstrates that the chemputer can implement

any feasible chemical synthesis, showing that the function  $\tau$  is sufficiently general to

account for all chemical reactions possible under the reagents given, process conditions,

and catalysts.

2. Construction of Synthesis Pathway: For each target compound c, a sequence  $\sigma$  of

transformations from initial reagents  $R_0$  to c can be constructed. This construction must

account for all intermediate transformations and ensure that  $\sigma$  is valid under P, and K.

3. **Verification of Stability**: This verifies that for the resulting compound *c*, the stability

condition S(c) is satisfied.

4. **Dynamic Error Detection and Correction**: The chemputer can detect errors in real-time

during each step of the synthesis by continuously monitoring the reaction progress and

comparing the actual outcome with the expected result. Upon detecting an error during

any synthesis step, the chemputer applies corrective steps immediately, either reverting

to a previous state or adjusting the process to ensure the synthesis remains on track.

5. **Universality and Completeness**: This proves that for any  $c \in C$ , there exists a pathway  $\sigma$ 

and a stable outcome, demonstrating the universality of the chemputer as a synthesis

device, including error detection and correction at each synthesis step.

**Formalization** 

**Stability Condition** 

S(c): Stability  $c \in C$  such that c is isolable and stable

The stability condition S(c) ensures that the resulting compound c is stable and can be isolated,

i.e., S(c) must hold true for the synthesis to be considered successful. However, the synthesis

may or may not utilize unstable reaction intermediates that could be isolated for some period

of time.

5

## Transformation Function $(\tau)$

$$\tau: R \times P \times K \rightarrow C$$

The transformation function  $\tau$  defines the emergent property we conventionally call the reaction rule which is the resultant outcome when reagents R are added under the process conditions P, in the presence of catalysts K to give the output compounds C. The transformation function can be used to predict how the reagent graphs R can be transformed into the product graphs C as graph transformations between the reagents R.

## Construction of Synthesis Pathways ( $\sigma$ )

For any target compound  $c \in C$ , we construct a pathway  $\sigma$  such that:

$$\sigma: (R_0, R_1, ..., R_n) \rightarrow C$$

A synthesis pathway  $\sigma$  is a sequence of transformations leading from an initial set of reagents  $R_0$  through intermediate sets  $R_1,...,R_n$  to the final product c. The chemputer is said to be universal if, for any target compound c in the set of desired compounds c, there exists a sequence of transformations  $\sigma$  that leads from an initial set of reagents  $R_0$  to c.

## **Existence of Synthesis Pathway (Universal Synthesis Theorem)**

$$\forall c \in C, \exists R_0 \subseteq R, P, K \text{ such that } \sigma(R_0,...,R_n) = c$$

This theorem asserts that for every target compound c in C, there exists a set of initial reagents  $R_0 \subseteq R$ , a set of process conditions P, and catalysts K such that a synthesis pathway  $\sigma$  exists, leading from  $R_0$  to c.

### **Dynamic Error Detection and Correction**

 $\delta: c' \rightarrow \text{Corrected State } c \text{ corrected}$ 

Dynamic error correction is applied at each step in the synthesis process. For each transformation, if the outcome c' deviates from the expected intermediate or final product  $c_n$ , the error detection function  $\epsilon$  flags the deviation ( $\epsilon$  = 1). The error correction function  $\delta$  is then applied to revert to a prior valid state or adjust the process dynamically to ensure that the synthesis remains accurate.

# Chempiling Function $(\chi)$

$$\chi: \sigma \to G(H)$$

The chempiling function  $\chi$  maps the synthesis pathway  $\sigma$  into a hardware configuration G(H) that can execute the synthesis process.

## **Proof of Universality**

**Base Case**: For simple compounds (e.g., elements or basic molecules), the chemputer can directly synthesize them from their constituent elements or simpler precursors. If an error occurs during the synthesis of these simple compounds, it is detected and corrected dynamically before proceeding.

**Inductive Step**: Assume the chemputer can synthesize all compounds of complexity k (i.e., requiring k steps), with dynamic error correction applied at each step. For a compound of complexity k+1, there exists a precursor compound requiring k steps and a transformation function  $\tau$  that can transform this precursor into the target compound under appropriate P, and K in the presence of the reagents R. The dynamic error correction function  $\delta$  ensures that

each intermediate step is accurate. Therefore, by induction, the chemputer can synthesize all compounds up to any finite complexity.

#### **Practical Limitations**

Implementing the concept of chemputation in practice presents a series of significant challenges that extend beyond this robust theoretical framework. One of the foremost challenges lies in the complexity and scalability of the chemputer's hardware. The concept of universally configurable hardware, which is central to the chemputer's ability to synthesize any chemical compound, demands a highly versatile and flexible system with a range of different modules for operations like filtration, extraction and so on. Designing hardware that can seamlessly switch between different configurations for a wide variety of chemical processes is an intricate task. Each module within the system must handle diverse reaction types, process conditions, and scales of operation while maintaining precision and reliability. Moreover, there is an inherent tension between the need for miniaturization, which allows for precision, and the requirement for scalability to manage larger volumes or more complex reactions. Achieving both in a single system, particularly one that remains flexible and configurable, is a significant engineering challenge. Furthermore, the integration of this hardware with the software responsible for the chempiling function—mapping synthesis pathways to specific hardware configurations—adds another layer of complexity. This software must dynamically adjust the hardware setup in real-time, requiring a level of synchronization and control that is difficult to achieve.

Another critical challenge is the implementation of dynamic error correction within the chemputer, which is essential for ensuring the accuracy and reliability of chemical syntheses. The system must be capable of real-time monitoring and adjustment, continuously tracking the progress of each reaction, detecting any deviations from the expected pathway, and applying corrective measures immediately. This demands advanced sensing technologies and real-time data processing capabilities that can operate effectively across a broad range of reaction conditions. In multi-step syntheses, errors can propagate through the system,

compounding and becoming more difficult to correct as the process continues. Developing mechanisms that can effectively manage and contain such errors, ensuring the robustness and redundancy of the system, is crucial. Achieving this balance between robustness, cost, space, and energy efficiency poses a significant challenge.

The theoretical framework also assumes a comprehensive understanding of the chemical space and the ability to encode all possible reactions into the chemputer. However, the reality of chemical synthesis is more complex. Our current knowledge of chemical reactions is not exhaustive, particularly in the fields of complex organic and biological chemistry, where many reactions remain poorly understood or unpredictable. This limitation restricts the chemputer's ability to reliably handle all potential syntheses. Moreover, as complex molecules are synthesized, emergent properties may arise that are not predicted by existing models, leading to unexpected reactions or products. The chemputer must be designed to manage and correct such deviations, even in the face of novel or poorly understood chemistry. Developing algorithms and hardware that can adapt to new chemical data in real-time is a significant hurdle that must be overcome.

#### **Conclusions**

Since the chemputer can implement any transformation function  $\tau$  and can control all relevant process conditions, and catalysts, it can instantiate any chemical synthesis process. The inclusion of dynamic error detection and correction at each step ensures the reliability and accuracy of the synthesis. Additionally, the use of universally configurable hardware and the chempiling function allows the chemputer to dynamically adapt its configuration for various synthesis pathways. Thus, the chemputer is universal for chemical synthesis, capable of generating any compound  $c \in C$  given the appropriate initial conditions, transformations, and error correction mechanisms.

The formalization above establishes the concept of a chemputer as a universal chemical synthesis machine. The transformation function  $\tau$ , synthesis pathways  $\sigma$ , stability conditions S,

dynamic error correction  $\delta$ , chempiling function  $\chi$ , and configurable hardware H together define a universal model capable of synthesizing any target compound within the chemical space defined by R, P, and K.

The work presented here establishes the chemputer as a universal chemical synthesis machine, demonstrating its capability to synthesize any target compound within a defined chemical space. By formalizing the key components, such as the transformation function  $\tau$ , synthesis pathways  $\sigma$ , stability conditions S, dynamic error correction  $\delta$ , and the chempiling function  $\chi$ , we have constructed a robust theoretical framework that underpins this universality. The integration of universally configurable hardware further enhances the chemputer's adaptability, allowing it to dynamically reconfigure and execute a wide array of chemical processes with precision. This is universal considering finite constraints on the reaction hardware, reagents, reaction steps, and reaction time.

## **Acknowledgements**

I would like to thank David Deutsch, Muffy Calder, Sara Walker, Abhishek Sharma, S. Hessam Mehr, Keith Patarroyo, Emma Clarke, Dario Caramelli, and Edward Lee for comments and feedback. I acknowledge financial support from the John Templeton Foundation (grants 61184 and 62231), Sloan Foundation, Schmidt Futures, NIH, Google, and EPSRC (grant nos. EP/L023652/1, EP/R01308X/1, EP/S019472/1, and EP/P00153X/1 and ERC (project 670467 SMART-POM).

## References

- [1] Alonzo Church. A note on the Entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1):40–41, 1936.
- [2] Alan M. Turing. On Computable Numbers, with an Application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.
- [3] Alan M. Turing. Computing Machinery and Intelligence. *Mind*, 59:433–60, 1950.

- [4] Sebastian Steiner, Jakob Wolf, Stefan Glatzel, Anna Andreou, Jarosl aw M. Granda, Graham Keenan, Trevor Hinkley, Gerardo Aragon-Camarasa, Philip J. Kitson, Davide Angelone, and Leroy Cronin. Organic synthesis in a modular robotic system driven by a chemical programming language. *Science*, 363(6423):144–152, 2019.
- [5] Artem I. Leonov, Alexander JS Hammer, Slawomir Lach, S. Hessam M. Mehr, Dario Caramelli, Davide Angelone, Aamir Khan, Steven O'Sullivan, Matthew Craven, and Liam Wilbraham. An integrated self-optimizing programmable chemical synthesis and reaction engine. *Nature Communications*, 15(1):1240, 2024.
- [6] Piotr S. Gromski, Jarosl aw M. Granda, and Leroy Cronin. Universal chemical synthesis and discovery with 'The Chemputer'. *Trends in Chemistry*, 2(1):4–12, 2020.
- [7] J. Sebasti'an Manzano, Wenduan Hou, Sergey S. Zalesskiy, Przemyslaw Frei, Hsin Wang, Philip J. Kitson, and Leroy Cronin. An autonomous portable platform for universal chemical synthesis. *Nature Chemistry*, 14(11):1311–1318, 2022.
- [8] Liam Wilbraham, S. Hessam M. Mehr, and Leroy Cronin. Digitizing Chemistry Using the Chemical Processing Unit: From Synthesis to Discovery. *Acc. Chem. Res.*, 54(2):253–262, 2021.
- [9] Robert Rauschen, Mason Guy, Jason E. Hein, and Leroy Cronin. Universal chemical programming language for robotic synthesis repeatability. *Nature Synthesis*, (3):488–496, 2024.
- [10] David Deutsch. Constructor theory. Synthese, 190(18):4331–4359, 2013.
- [11] Jaroslaw M. Granda, Liva Donina, Vincenza Dragone, De-Liang Long, and Leroy Cronin. Controlling an organic synthesis robot with machine learning to search for new reactivity. *Nature*, 559(7714):377–381, 2018.
- [12] S. Hessam M. Mehr, Matthew Craven, Artem I. Leonov, Graham Keenan, and Leroy Cronin. A universal system for digitization and automatic execution of the chemical synthesis literature. *Science*, 370(6512):101–108, 2020.