Automated inference of production rules for glycans

To be presented at CMSB'21

Ansuman Biswas[2], Ashutosh Gupta[1],

Meghana Missula[1] and Mukund Thattai[2]

[1]IITB

[2]NCBS

Formal methods in Biology

- Computationally hard problems in Biology
  - Examples: Determining future mutations of cancer, brain models

Formal methods in Biology

- Computationally hard problems in Biology
  - Examples: Determining future mutations of cancer, brain models

- Formal methods may help

Formal methods in Biology

- Computationally hard problems in Biology
  - Examples: Determining future mutations of cancer, brain models

- Formal methods may help

- We present a novel application of formal methods in biology namely,

INFERENCE OF GLYCAN PRODUCTION RULES

Why are glycans important?

Why are glycans important?

- Glycans are molecules that are used to identify cell types like IP addresses

Why are glycans important?

- Glycans are molecules that are used to identify cell types like IP addresses

- Any drug given must not interfere with the glycan processes

Why are glycans important?

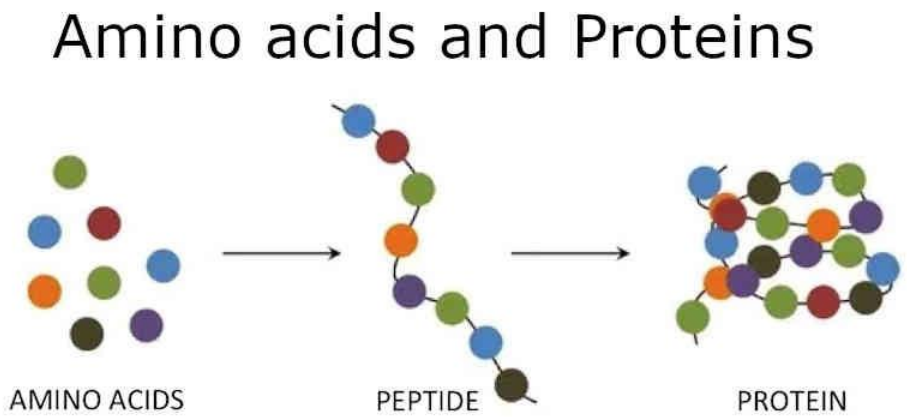
- Glycans are molecules that are used to identify cell types like IP addresses
- Any drug given must not interfere with the glycan processes
- Even though they are so important, we do not know how they are produced
  - Limited and expensive research

Why are glycans important?

- Glycans are molecules that are used to identify cell types like IP addresses
- Any drug given must not interfere with the glycan processes
- Even though they are so important, we do not know how they are produced
  - Limited and expensive research
- Maybe formal methods can find the production processes

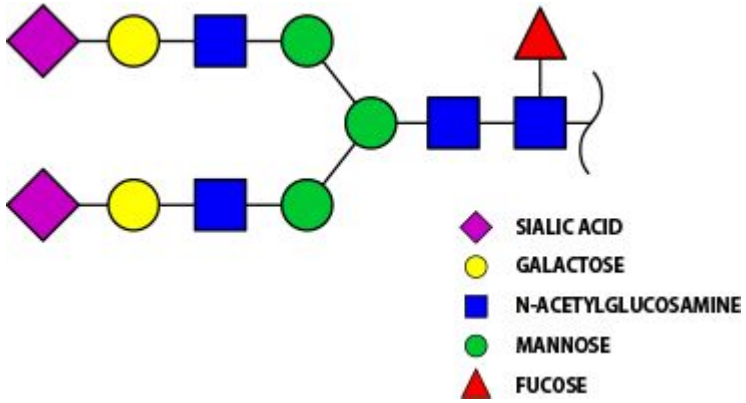
Assembling complex molecules

- Well known examples:
  - synthesis of linear DNA from nucleotide building blocks
  - synthesis of linear proteins from amino-acid building blocks



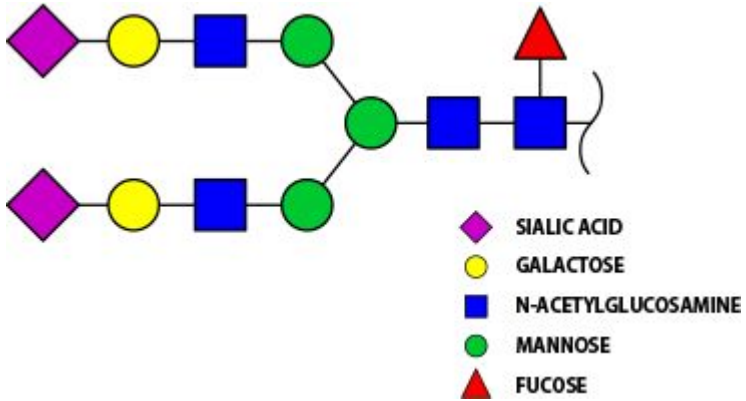
Complex sugars: Glycans

- Glycans



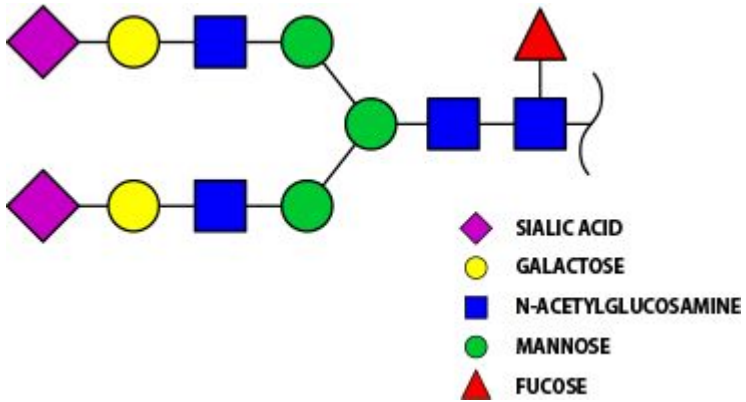
Complex sugars: Glycans

- Glycans
  - Tree-like polymers made up of sugar monomers



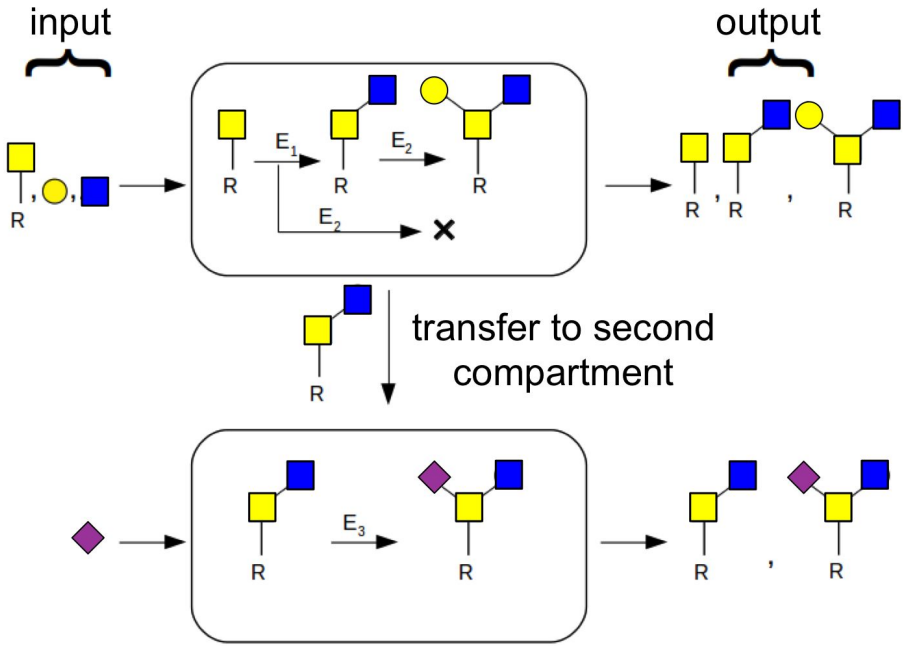
Complex sugars: Glycans

- Glycans
  - Tree-like polymers made up of sugar monomers
  - Set of which are found on the surface of all living cells; the set identifies the cell type



Assembling glycans

- Enzymes
  - Proteins which assemble glycans via successive additions, process known as **glycosylation**



Properties of Glycosylation I

- Enzymes have to work with limited resources
  - **Specificity:** Attachment of a monomer happens at a specific point on the tree
  - **Intra-cell variability**: Produce a given set of glycans using a few enzymes
  - **Inter-cell variability**: Different cell types have different types of glycans

Properties of Glycosylation II

- Other issues
  - **Microheterogeneity** and **stochastic** operation of enzymes

Properties of Glycosylation II

- Other issues
  - **Microheterogeneity** and **stochastic** operation of enzymes
  - Nevertheless, the glycan profiles of individual proteins are typically narrow and reproducible

Properties of Glycosylation II

- Other issues
  - **Microheterogeneity** and **stochastic** operation of enzymes
  - Nevertheless, the glycan profiles of individual proteins are typically narrow and reproducible

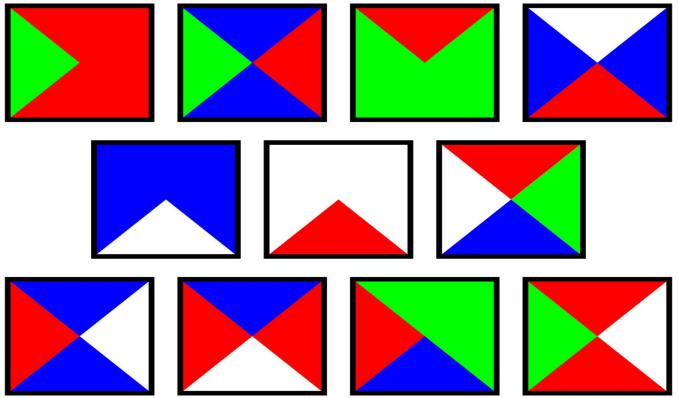
How can this stochastic and heterogeneous biosynthetic process generate narrow and reproducible glycan profiles?

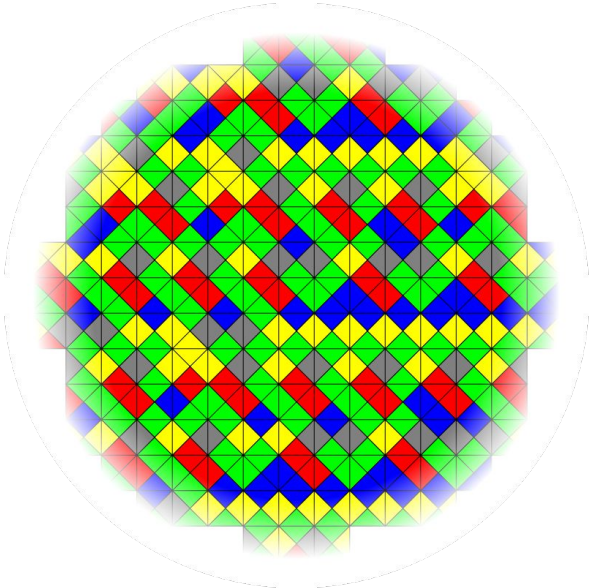
Wang tiles: An analogy

- A central inverse problem in self-assembly is to design building blocks that assemble into a target shape

Wang tiles: An analogy

- A central inverse problem in self-assembly is to design building blocks that assemble into a target shape
- Glycans may be considered a natural realization of the Wang construct, with monomers acting like tiles whose stickiness is encoded by GTase enzymes.



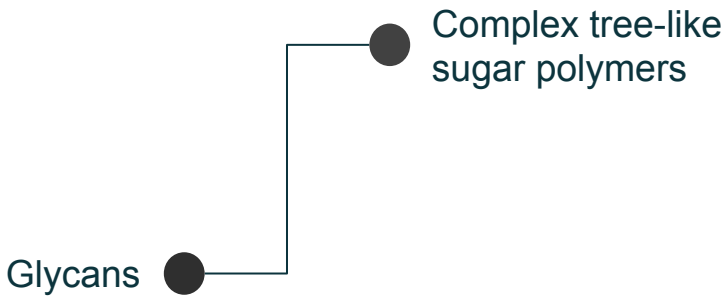


The CS problem

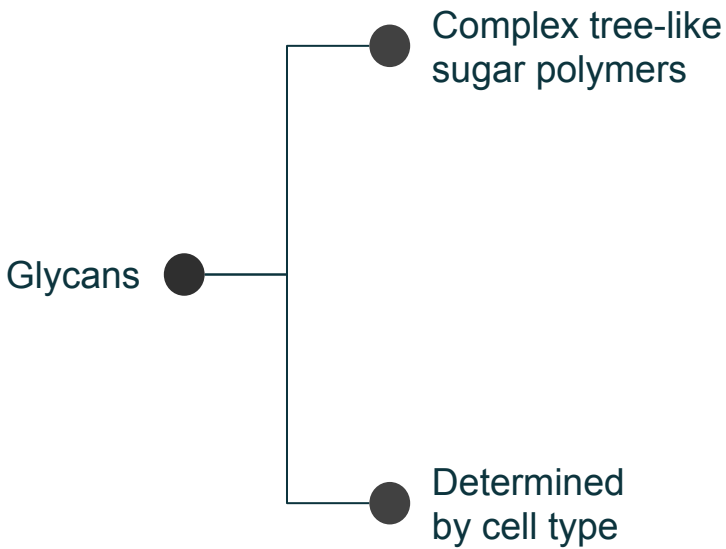
The CS problem

- Glycans

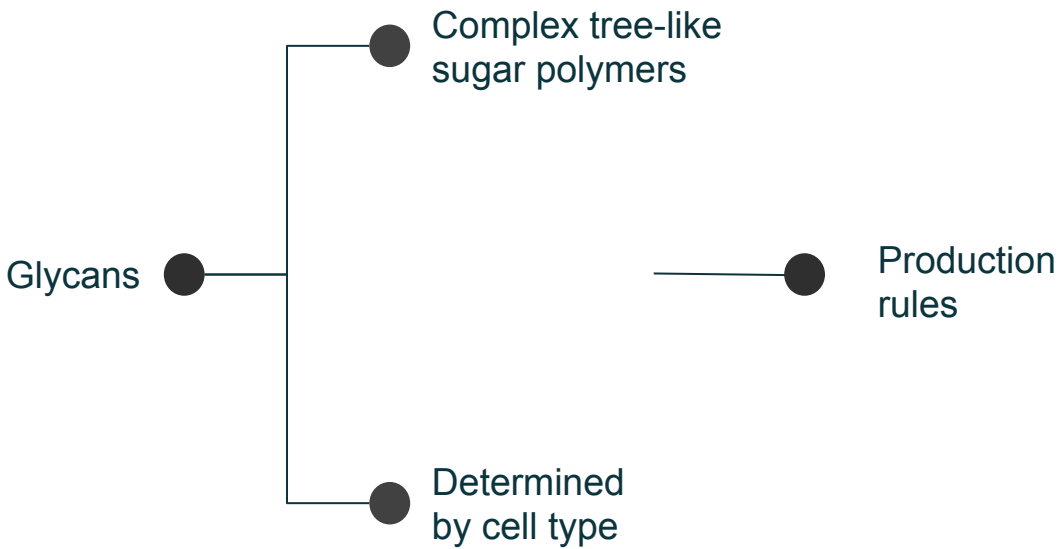
The CS problem



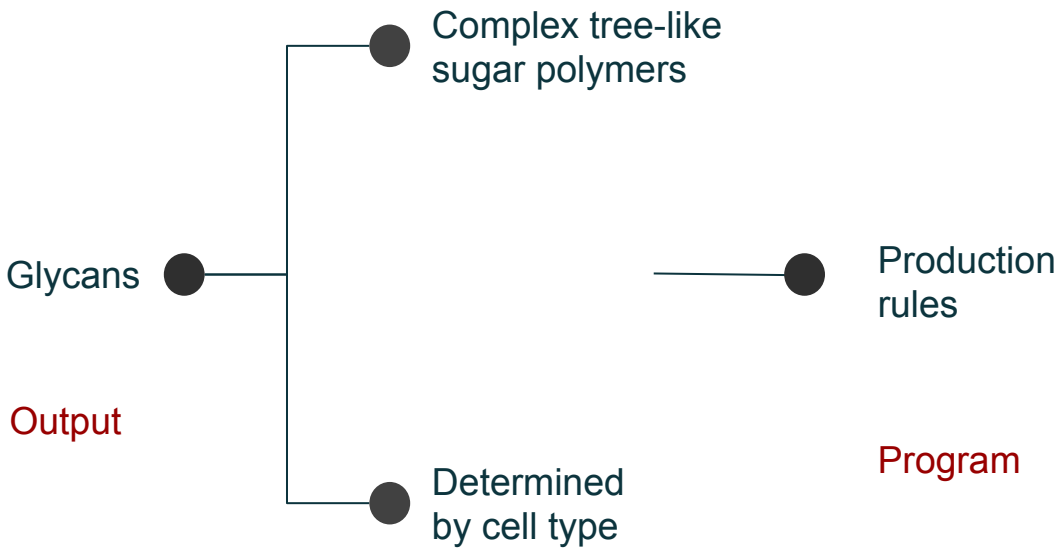
The CS problem



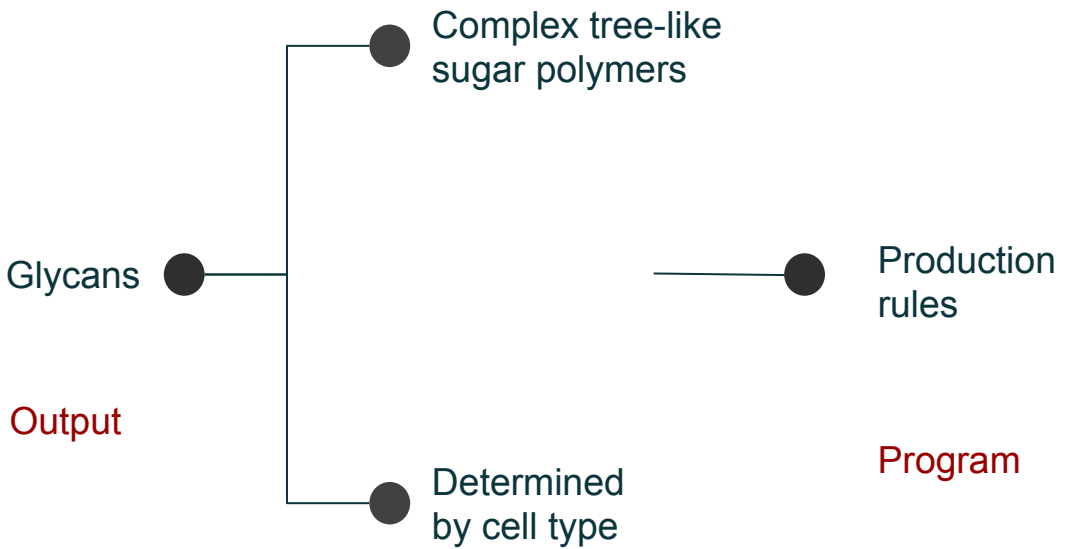
The CS problem



The CS problem



The CS problem



Find a program that produces the output i.e program synthesis

State-of-the-art for finding production rules

- Biologists' identification of production rules
  - Manual
  - Uses prior knowledge
  - Previous work: Algorithmic construct in special situations without stochasticity
  - Can we do better?

State-of-the-art for finding production rules

- Biologists' identification of production rules
  - Manual
  - Uses prior knowledge
  - Previous work: Algorithmic construct in special situations without stochasticity
  - Can we do better?

- Search space is large: 10^{70} rules sets possible !

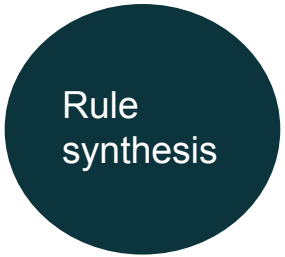
State-of-the-art for finding production rules

- Biologists' identification of production rules
  - Manual
  - Uses prior knowledge
  - Previous work: Algorithmic construct in special situations without stochasticity
  - Can we do better?

- Search space is large: 10^{70} rules sets possible !

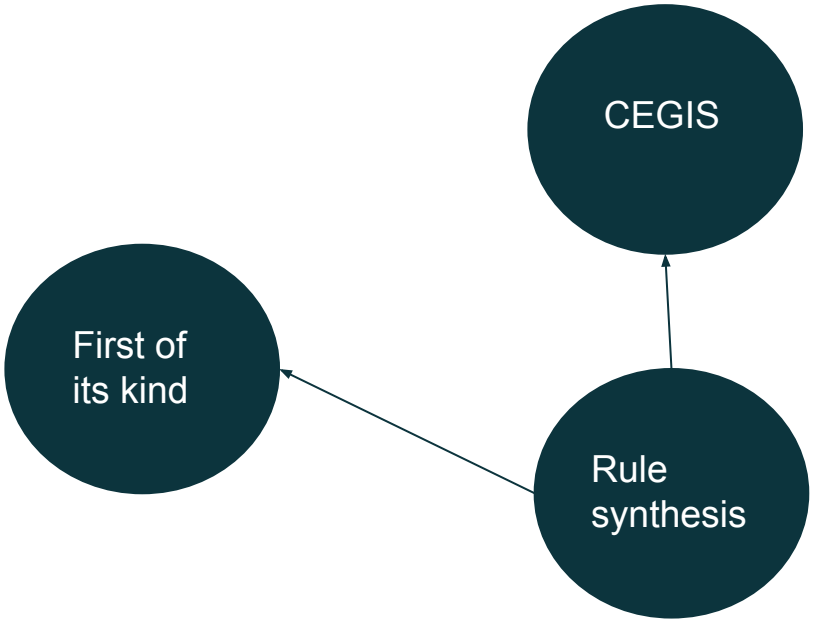
- Need for automated synthesis

Our method for production rule synthesis

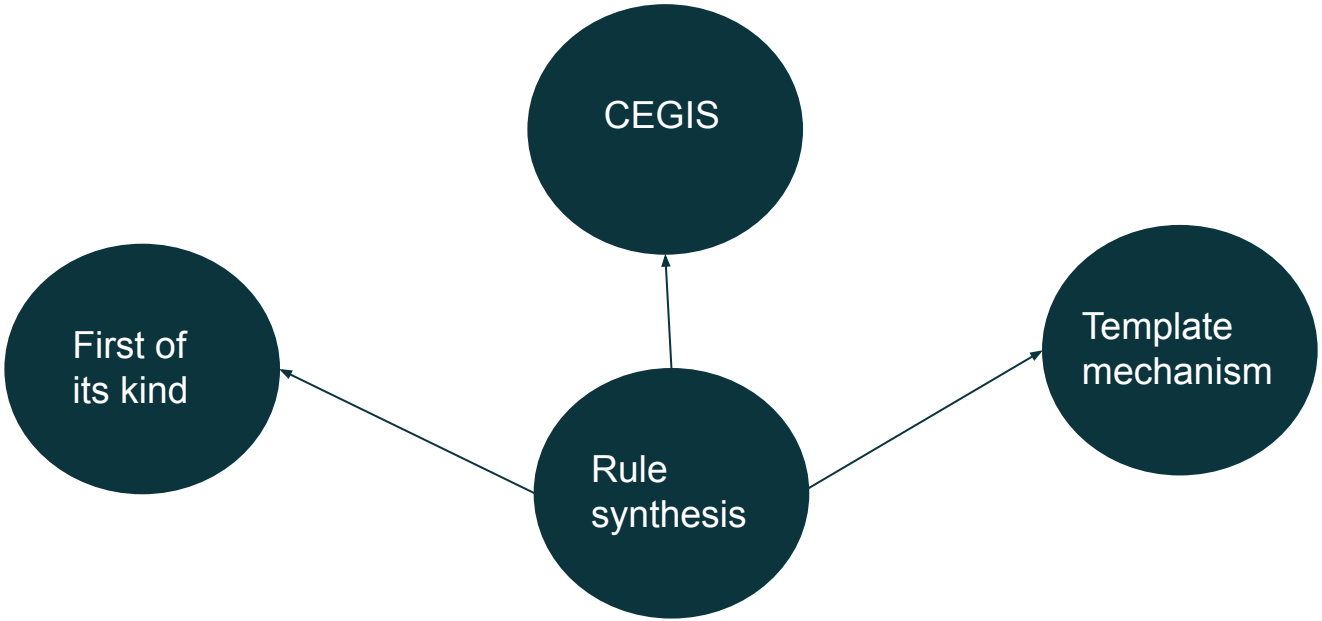


Our method for production rule synthesis

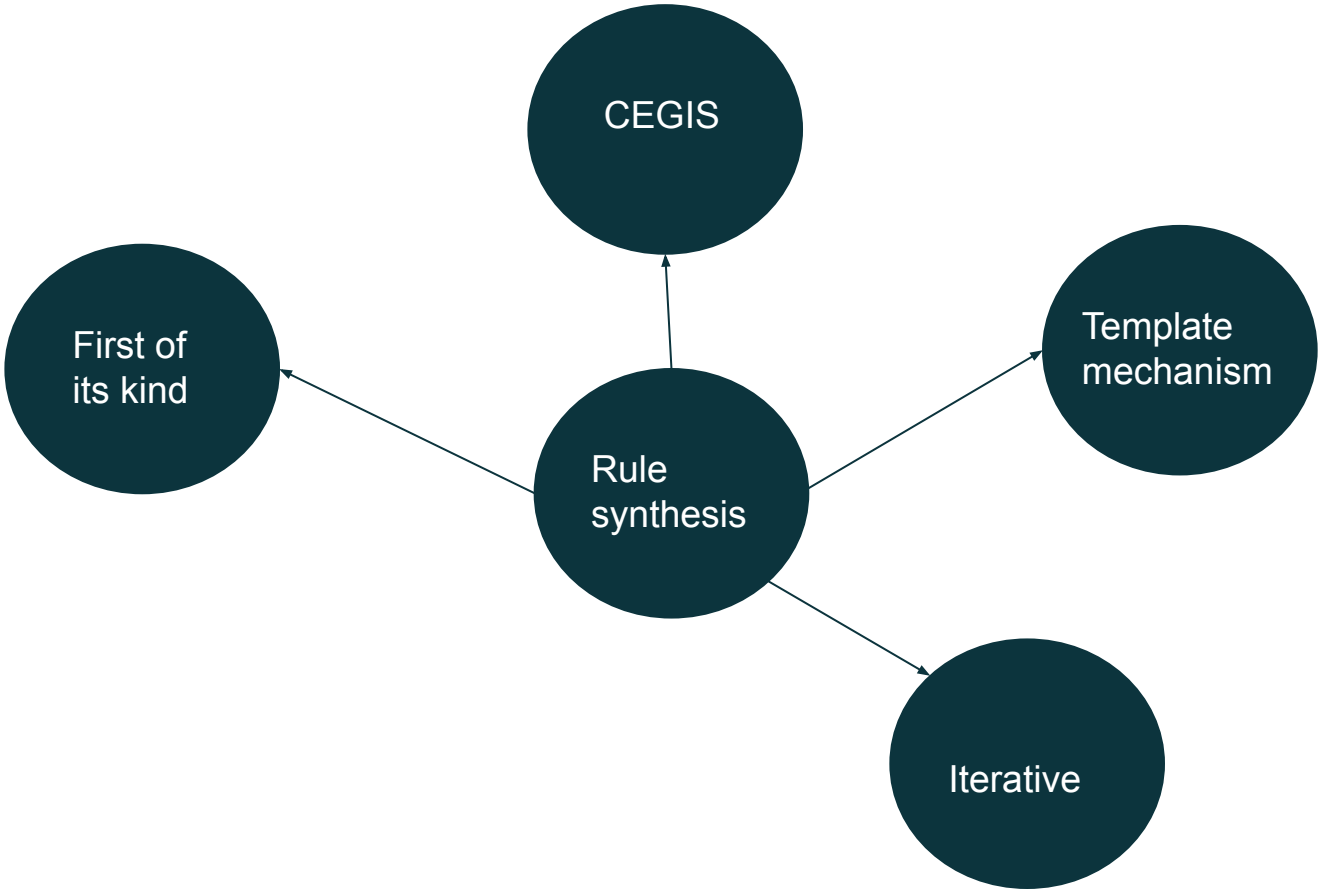
Our method for production rule synthesis



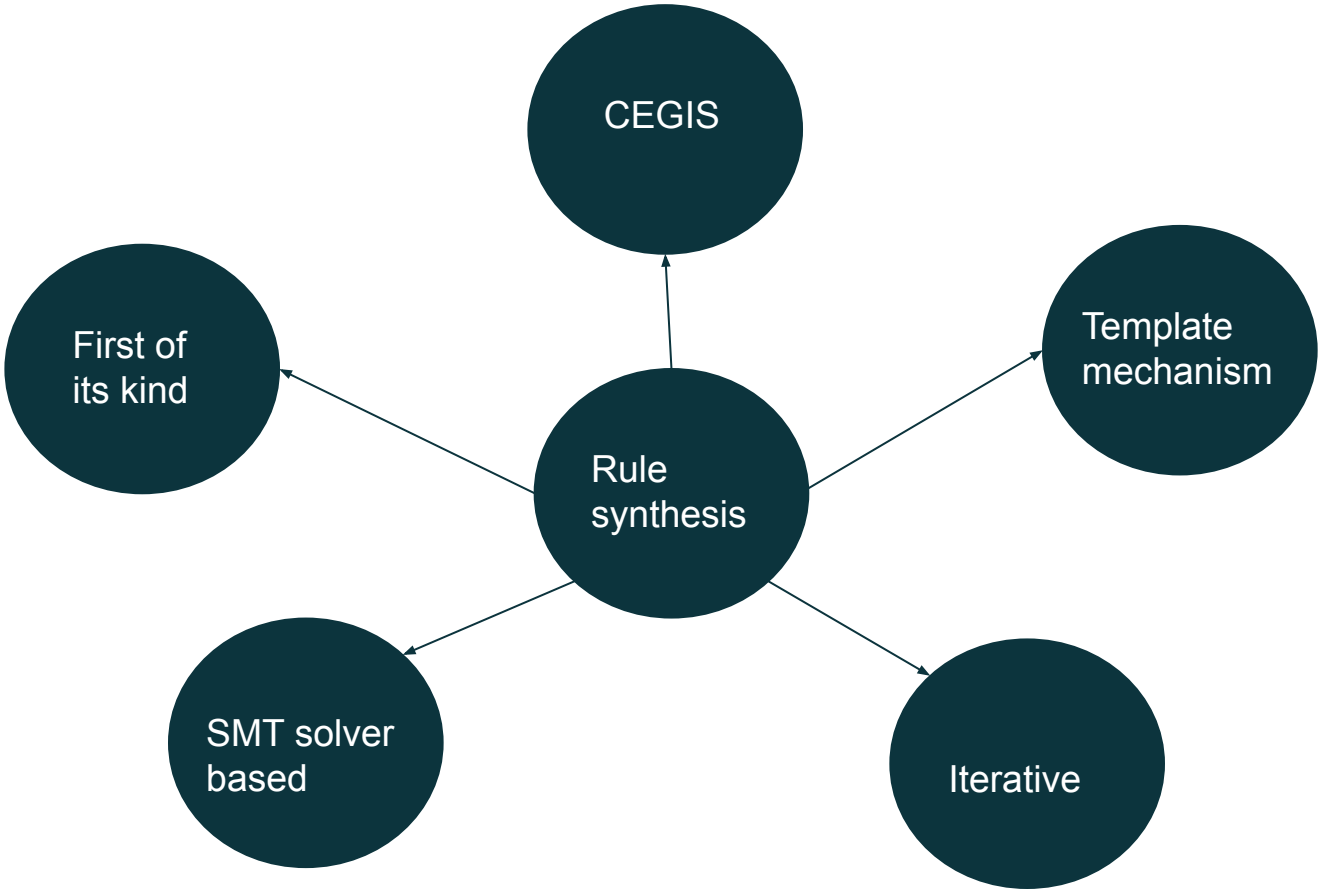
Our method for production rule synthesis



Our method for production rule synthesis



Our method for production rule synthesis



Our method

- *Synthesis query*: constraints to the solver
  - Unsatisfiable: No production rules in the current template's search space

Our method

- *Synthesis query*: constraints to the solver
  - Unsatisfiable: No production rules in the current template's search space
  - Satisfiable
    - *Counterexample query*: check that no extra molecules are produced than the input set
    - Add constraints for the same

Our method

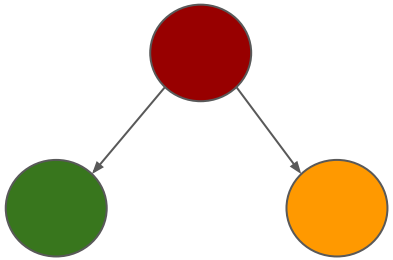
- *Synthesis query*: constraints to the solver
  - Unsatisfiable: No production rules in the current template's search space
  - Satisfiable
    - *Counterexample query*: check that no extra molecules are produced than the input set
    - Add constraints for the same
  - Variations: Coming up!

Our method

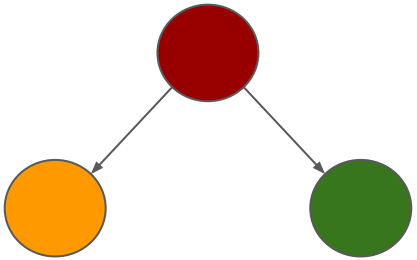
- *Synthesis query*: constraints to the solver
  - Unsatisfiable: No production rules in the current template's search space
  - Satisfiable
    - *Counterexample query*: check that no extra molecules are produced than the input set
    - Add constraints for the same
  - Variations: Coming up!

- Implemented in our tool, **`GlySynth`**!

Formal model: input glycan molecules



Molecule 1



Molecule 2

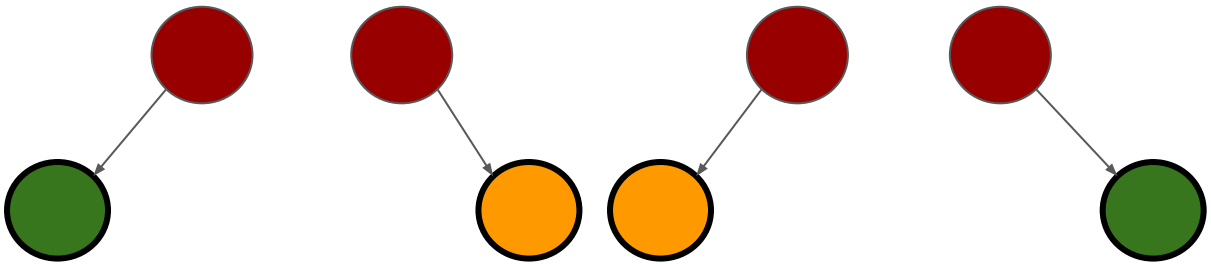
Input glycan molecules

Formal model: Monomers

Set of monomers = {  }

Formal model: candidate production rules

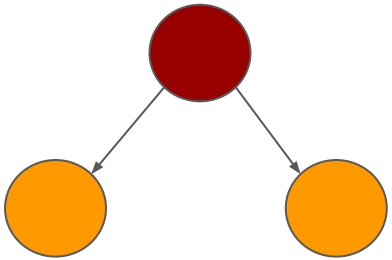
- Intuitively, we feel that there are 4 rules which can make this set of molecules



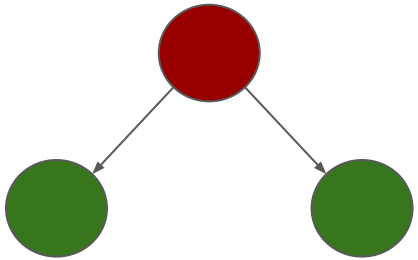
*Thick border on the circles represent the monomer being added

Formal model: counterexample molecules

- However, can these rules produce a molecule which is not in the set?



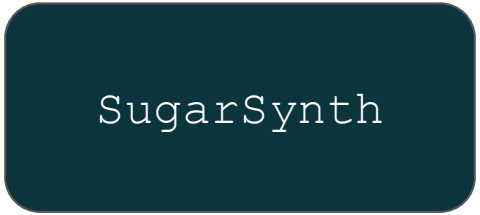
Extra Molecule 1



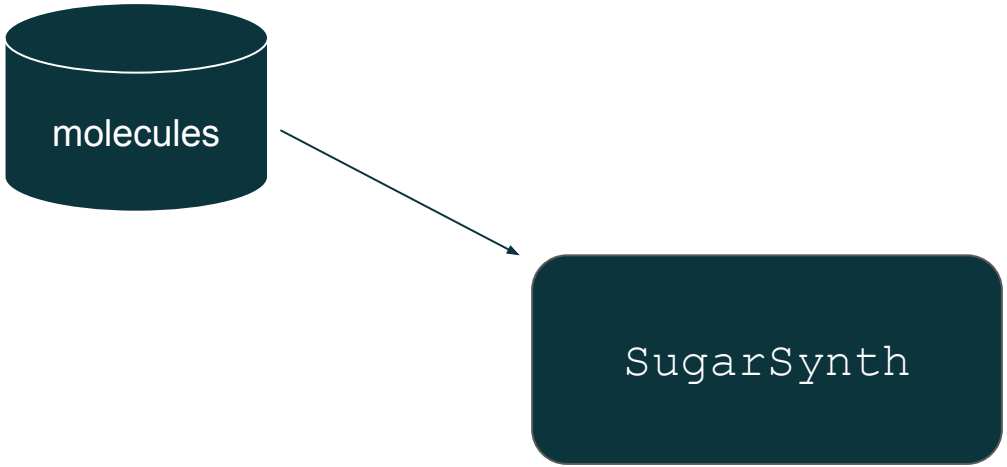
Extra Molecule 2

- Correct rules produced by **`GlySynth`**, coming up shortly!

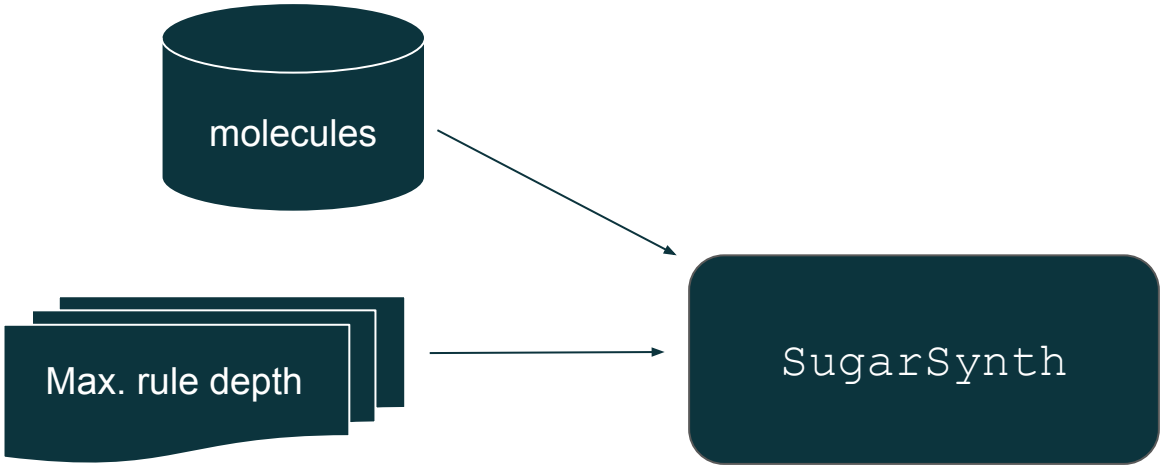
Input and Output



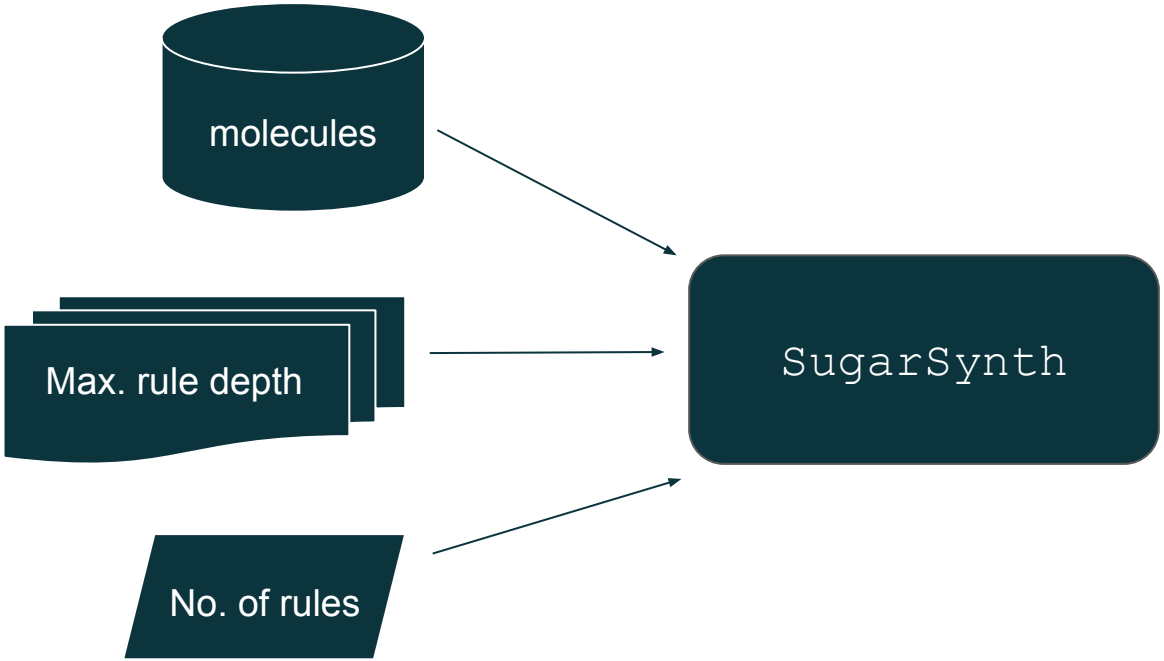
Input and Output



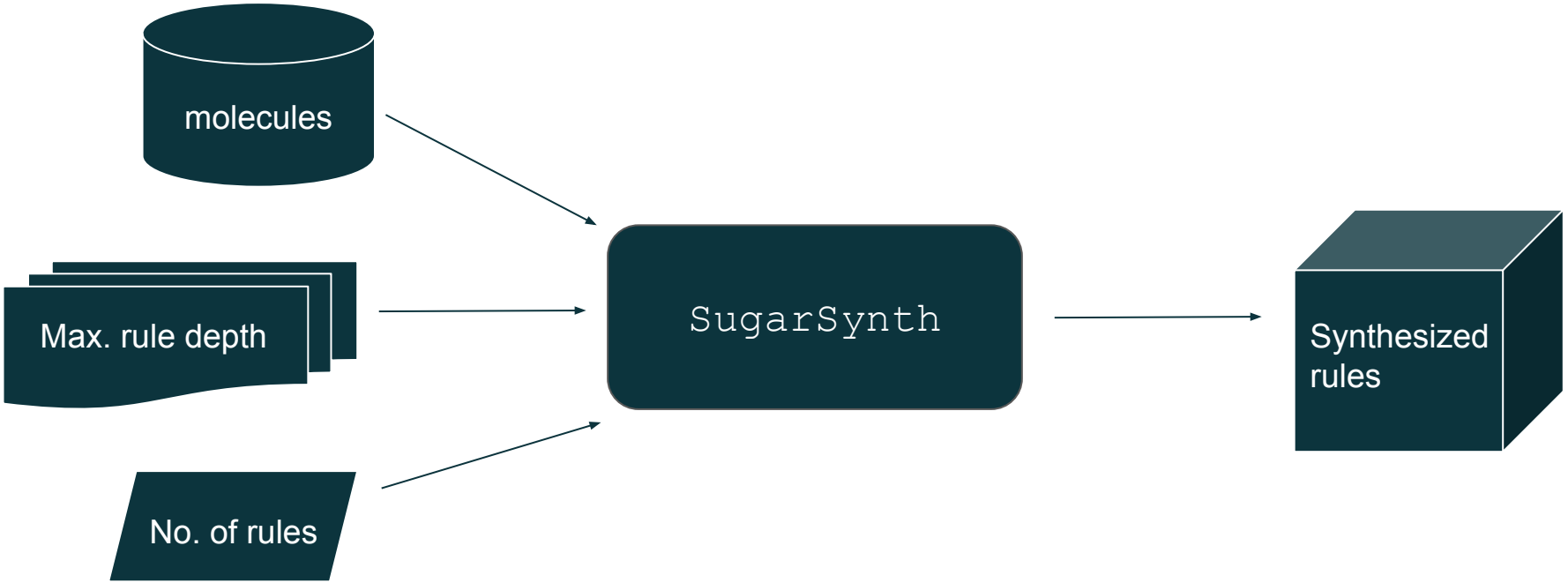
Input and Output



Input and Output



Input and Output

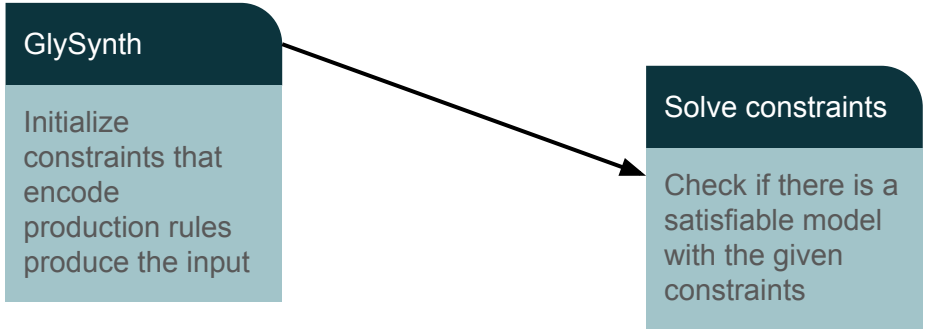


High-level Algorithm

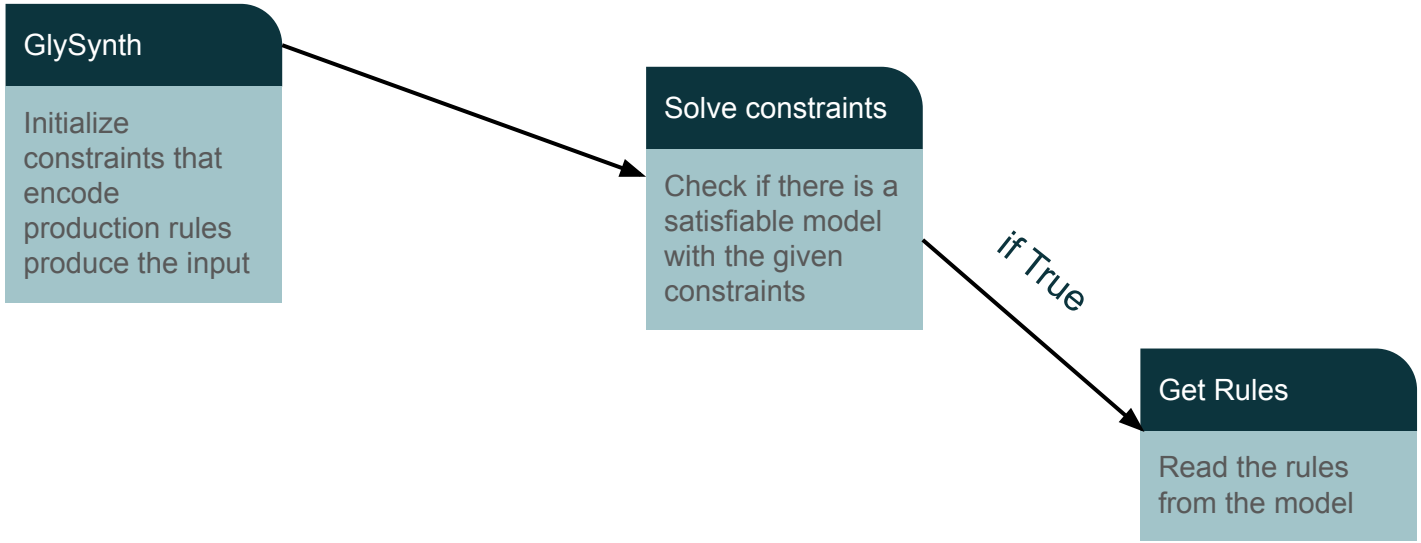
GlySynth

Initialize constraints that encode production rules produce the input

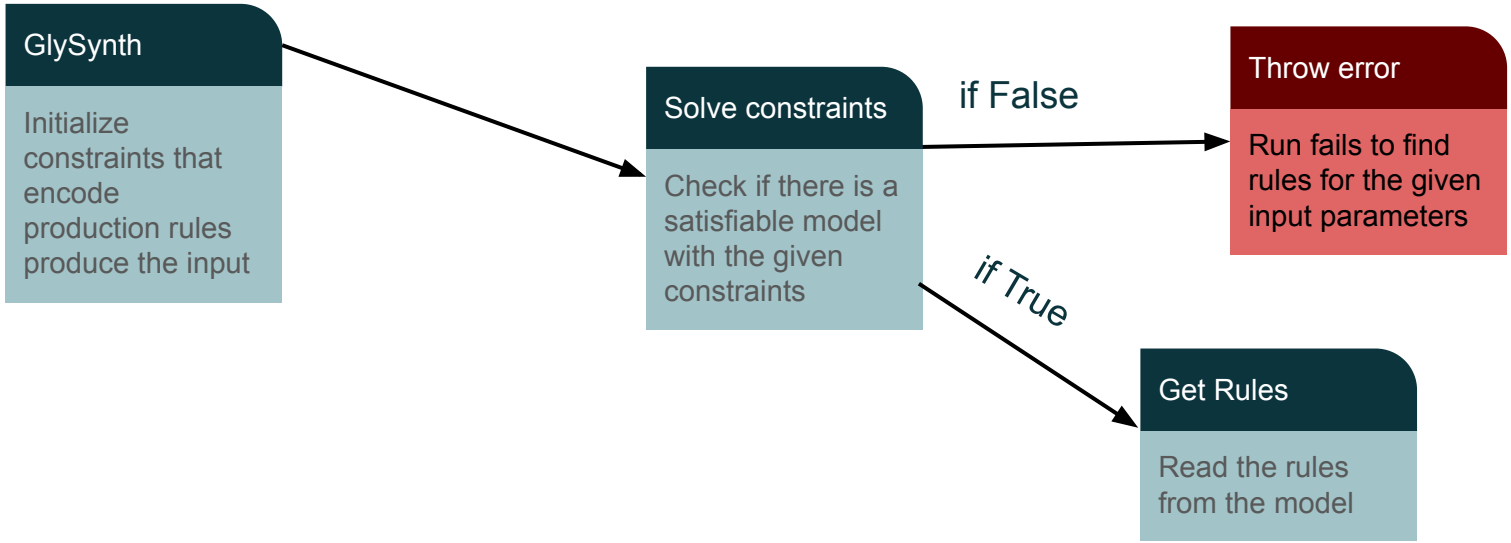
High-level Algorithm



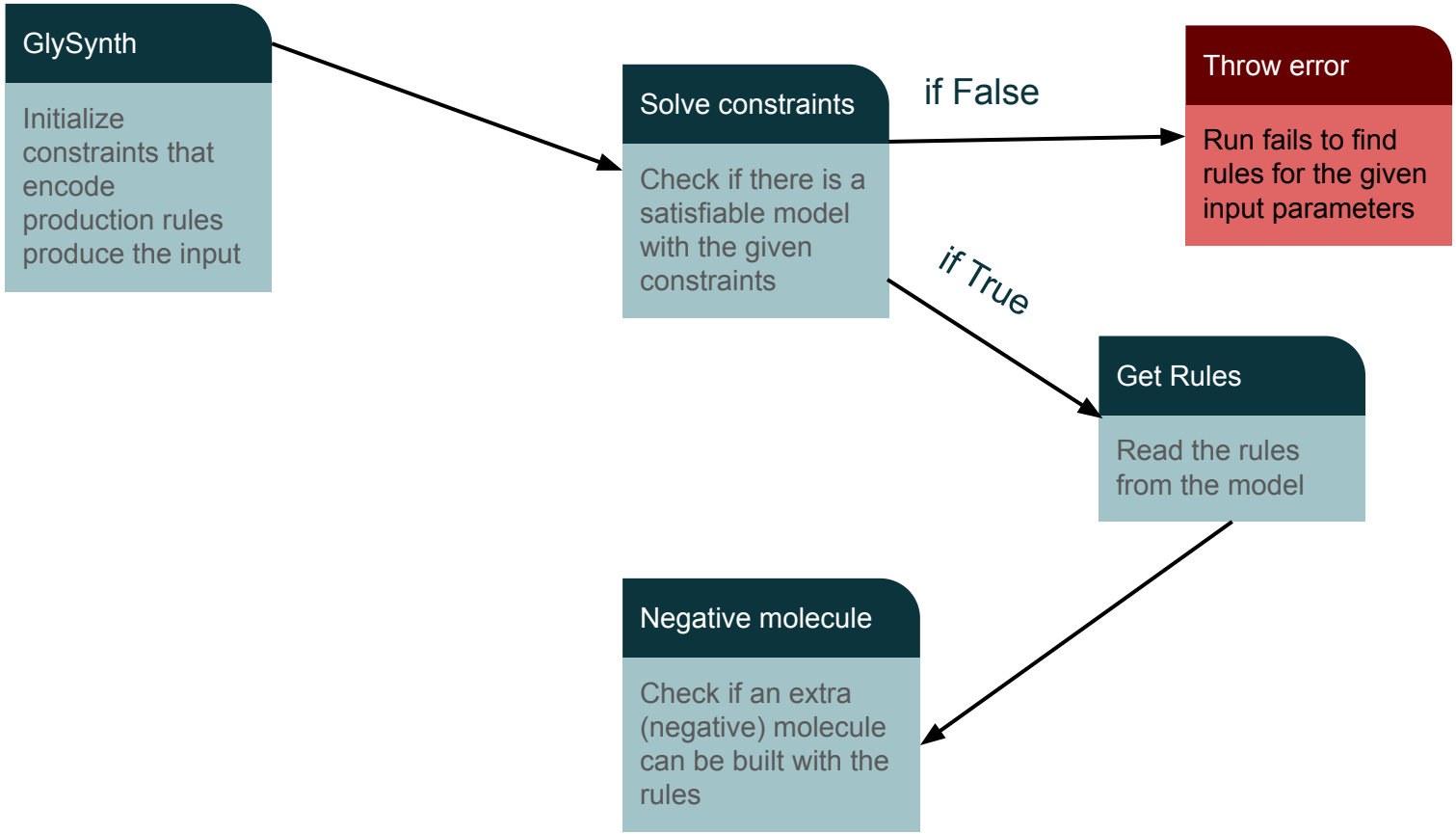
High-level Algorithm



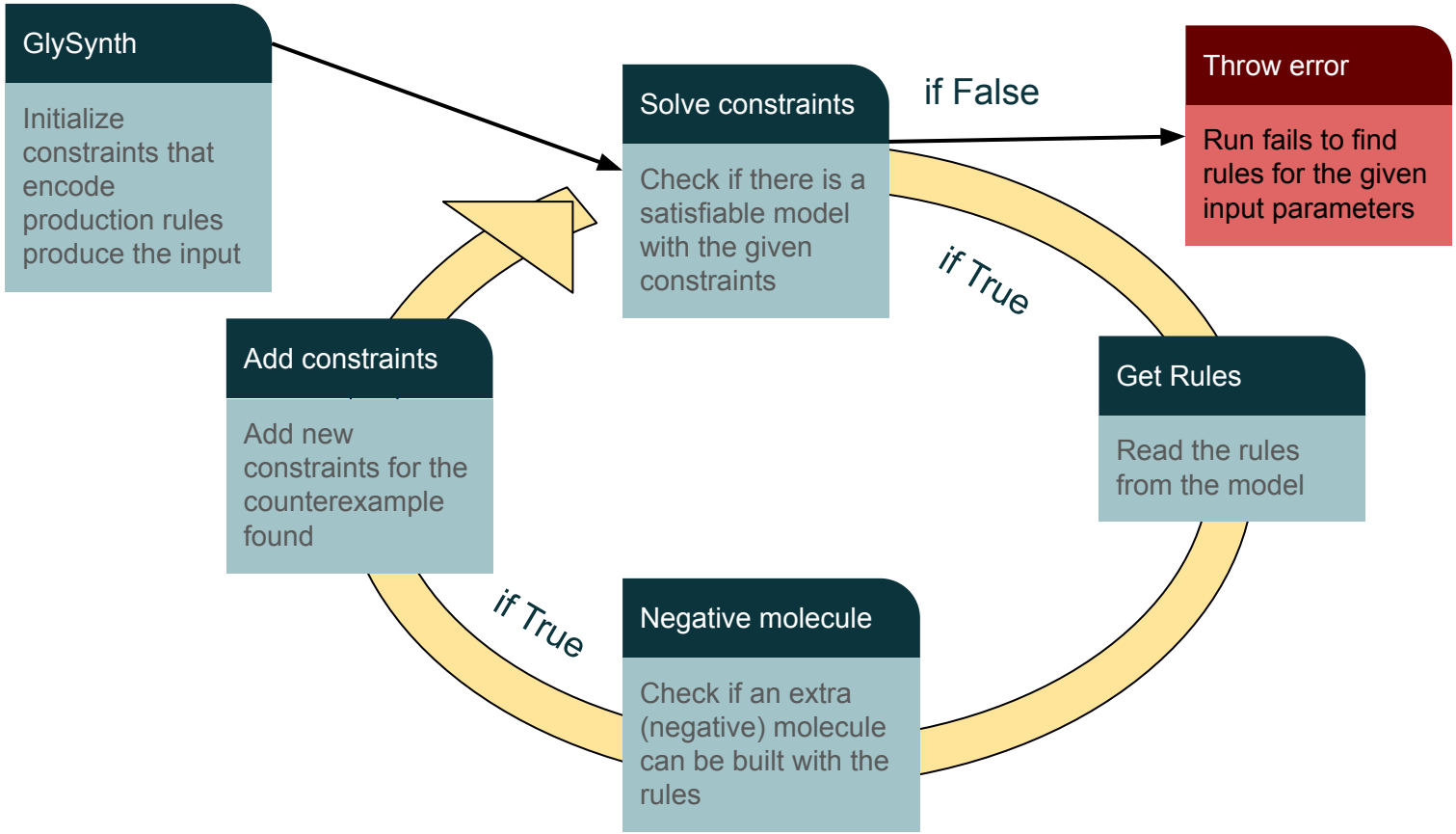
High-level Algorithm



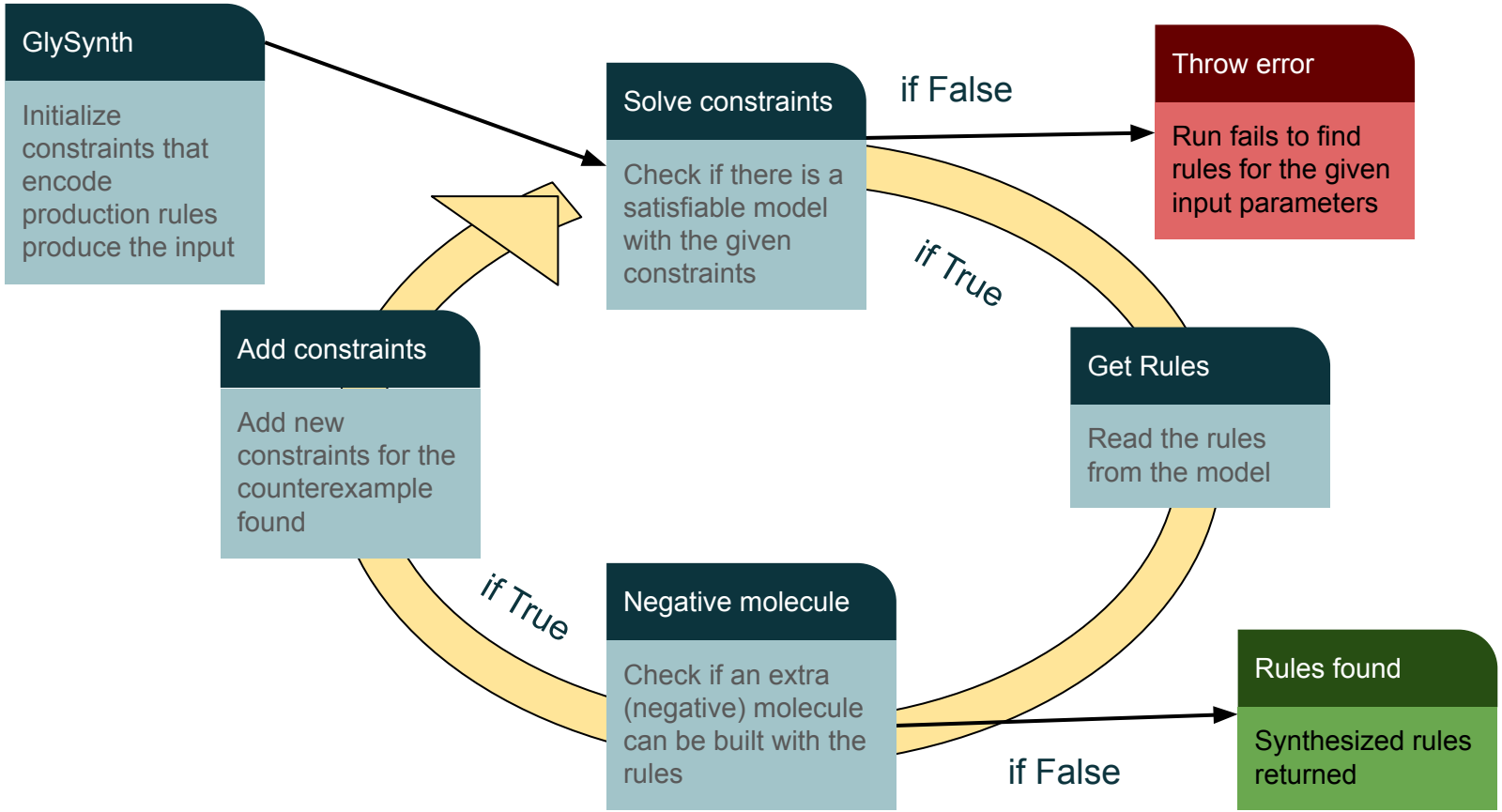
High-level Algorithm



High-level Algorithm

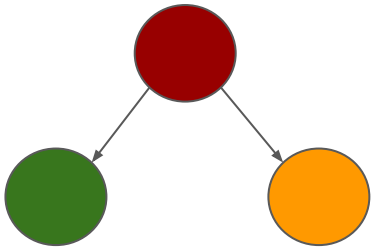


High-level Algorithm

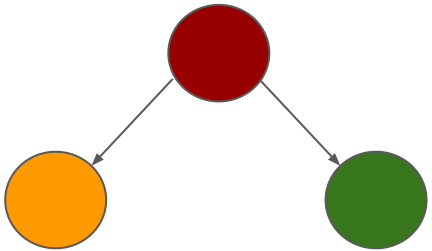


Revisiting our previous example

- Input molecules



Molecule 1

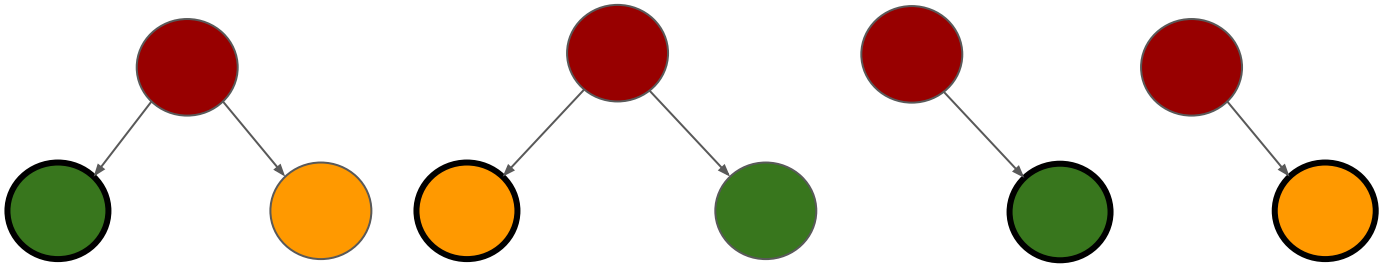


Molecule 2

- Number of rules to be synthesized = 4
- Maximum rule depth = 2

Revisiting our previous example

- Rules synthesized



Rule 1 Rule 2 Rule 3 Rule 4

- No extra molecule!

*Thick border on the circles represent the monomer being added

Variations: life is not that simple!

- Fast and slow reactions
  - Example coming up!

Variations: life is not that simple!

- Fast and slow reactions
  - Example coming up!

- Runaway reactions

Variations: life is not that simple!

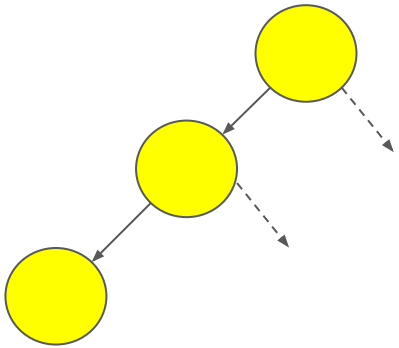
- Fast and slow reactions
  - Example coming up!
- Runaway reactions
- Compartments

Variations: life is not that simple!

- Fast and slow reactions
  - Example coming up!
- Runaway reactions
- Compartments
- Incomplete and noisy data

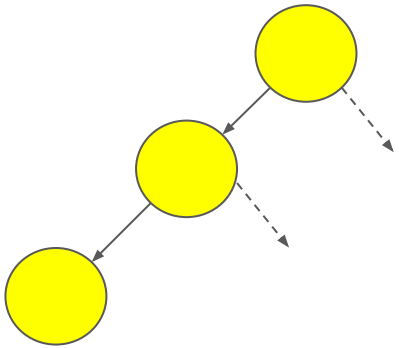
Another example

- Inputs



Another example

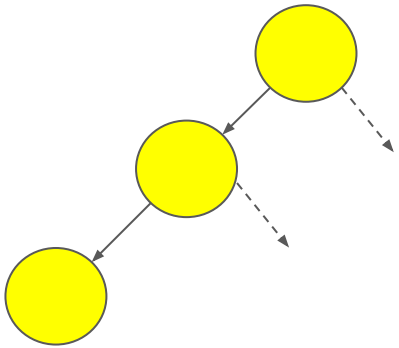
- Inputs



Max depth = 2

Another example

- Inputs

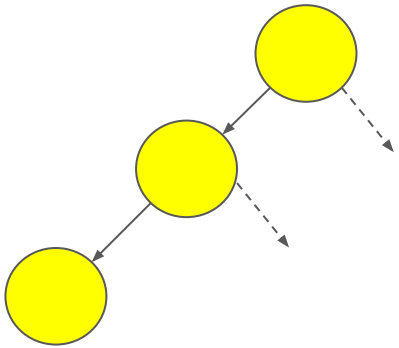


Max depth = 2

Number of rules to learn = 1

Another example

- Inputs



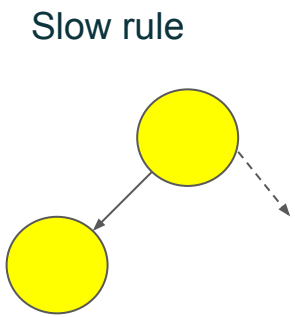
Max depth = 2

Number of rules to learn = 1

compartments = 1

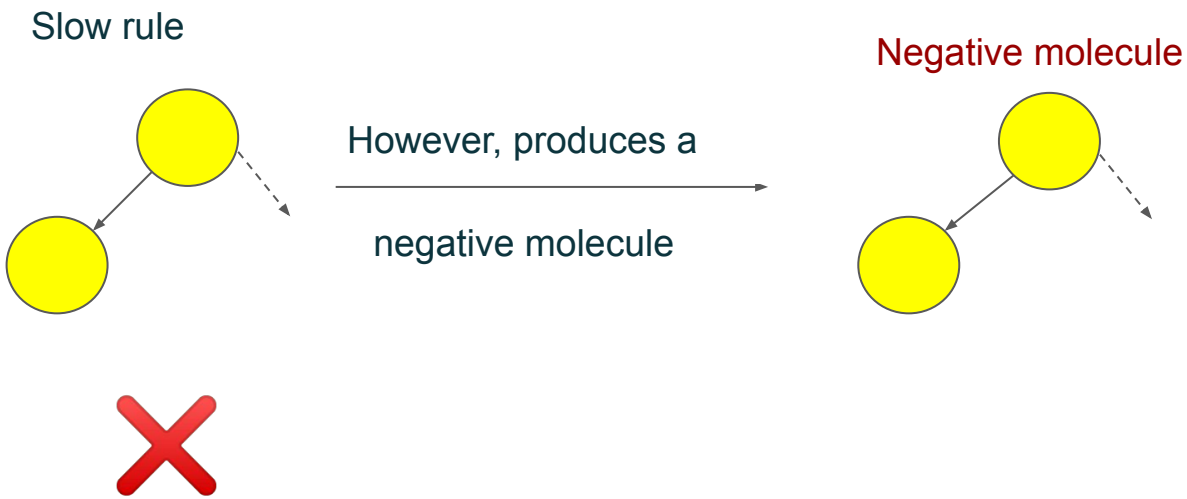
Another example

- First iteration



Another example

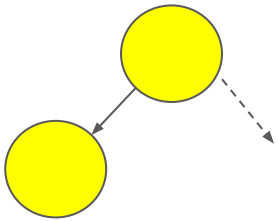
- First iteration



Another example

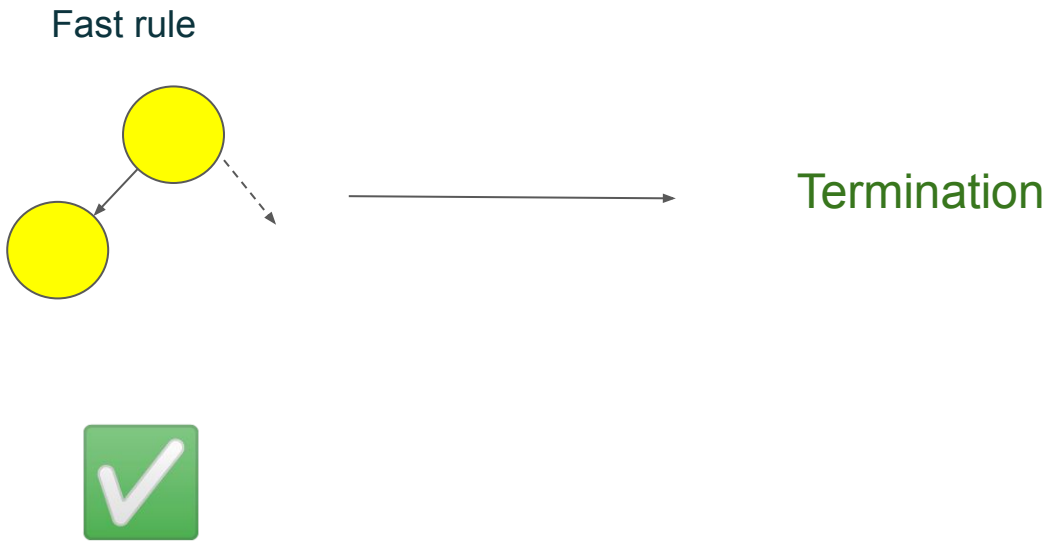
- Second iteration

Fast rule



Another example

- Second iteration



Our tool `GlySynth`

- Written in C++, uses z3
- Experiments data: skimming from literature
- Open Source
- Available on github: https://github.com/ashutosh0gupta/sugar-synth

Results

	#mol-ecules	#Rules	Rule depth	#Comp-artments	success?	Time (in secs.)
D1	6	7	3	1	Yes	3.02
		7	4	2	Yes	**1.60**
		6	3	3	Yes	9.36
D2	3	7	3	2	Yes	14.37
		5	3	2	Yes	**7.97**
		5	3	3	Yes	13.42
D3	6	6	4	2	Yes	1.02
		5	2	1	Yes	**0.57**
		5	4	1	Yes	0.71
D4	3	8	4	1	Yes	4.35
		6	3	1	Yes	**0.85**
		6	2	2	No	1.17

D1: Respiratory mucins of a cystic fibrosis patient
D2: Horse chorionic gonadotropin
D3: SARS-CoV-2 spike protein T323/S325
D4: Human chorionic gonadotropin from a cancer cell line

Future Work

- Full end-to-end interpretation of the data from the wet experiments

Future Work

- Full end-to-end interpretation of the data from the wet experiments

- Check viability of the solutions found by our method

Future Work

- Full end-to-end interpretation of the data from the wet experiments
- Check viability of the solutions found by our method
- Few more variants of the synthesis problem
  - Flexible compartment boundaries
  - Other stay models

Future Work

- Full end-to-end interpretation of the data from the wet experiments
- Check viability of the solutions found by our method
- Few more variants of the synthesis problem
  - Flexible compartment boundaries
  - Other stay models
- Modeling of the synthesis problem as minimization of a modified version of tree automaton

Conclusion

- This talk
  - Glycans and glycosylation: Need for automated synthesis
  - Formal modeling of the glycan synthesis problem: Algorithm & example

Conclusion

- This talk
  - Glycans and glycosylation: Need for automated synthesis
  - Formal modeling of the glycan synthesis problem: Algorithm & example
- In the paper CMSB’21
  - Details on the related work and the algorithms
  - (Abstract) modeling and formal justifications
  - More experiments and results

Conclusion

- This talk
  - Glycans and glycosylation: Need for automated synthesis
  - Formal modeling of the glycan synthesis problem: Algorithm & example
- In the paper CMSB'21
  - Details on the related work and the algorithms
  - (Abstract) modeling and formal justifications
  - More experiments and results
- Impact
  - Data analysis allows us to infer the causes of microheterogeneity and species-specific diversity in real glycan datasets
  - Novel synthesis method for discovering the production rules of glycan molecules from the output of the rules
  - Identification of a new area of application for formal methods in biology

Appendix

Glycosylation

- Grown without templates - unlike DNAs

Glycosylation

- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes

Glycosylation

- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes
- Final glycan structure is determined by the behavior of the enzymes themselves

Glycosylation

- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes
- Final glycan structure is determined by the behavior of the enzymes themselves
- Stochastic

Glycosylation

- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes
- Final glycan structure is determined by the behavior of the enzymes themselves
- Stochastic
- Promiscuous

Glycosylation

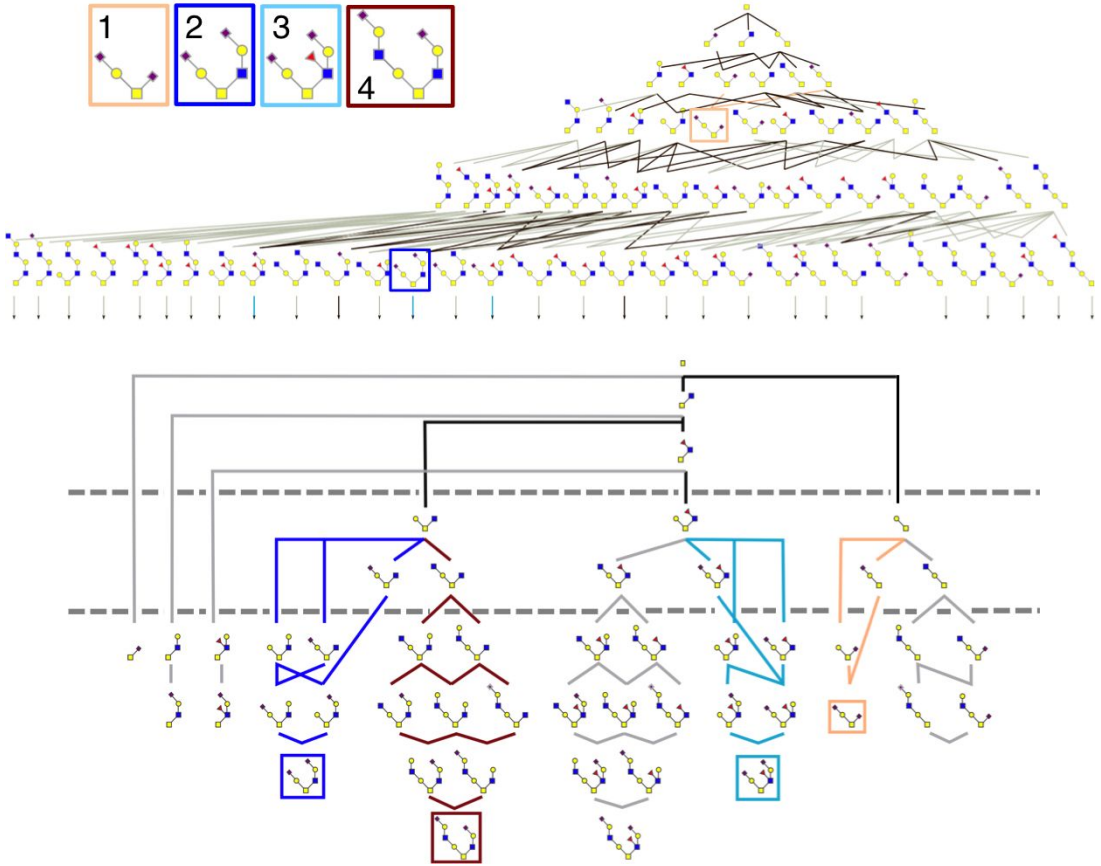
- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes
- Final glycan structure is determined by the behavior of the enzymes themselves
- Stochastic
- Promiscuous
- Biological experiment
    - produces a spectrum of glycan trees
    - previous work: a method to infer the production rules when a single glycan is produced

Glycosylation

- Grown without templates - unlike DNAs
- Carried out by large collection of GTase enzymes
- Final glycan structure is determined by the behavior of the enzymes themselves
- Stochastic
- Promiscuous
- Biological experiment
  - produces a spectrum of glycan trees
  - previous work: a method to infer the production rules when a single glycan is produced

Given a set of glycan trees produced by a cell, can we infer the set of enzymes that produce the glycans?

Single Vs Multiple compartments



Constraints

- Rules template correctness

Constraints

- Rules template correctness
- Molecule template correctness

Constraints

- Rules template correctness
- Molecule template correctness
- Constraints for the rules to produce the given molecule set

Constraints

- Rules template correctness
- Molecule template correctness
- Constraints for the rules to produce the given molecule set
- Constraints for the rules to not produce additional molecules

Properties of `SugarSynth`

- Soundness

Properties of `SugarSynth`

- Soundness
- Completeness

Properties of `SugarSynth`

- Soundness
- Completeness
- Generated rule set
  - Not unique
  - Not minimal
  - First set satisfying constraints is returned

How large is the search space?

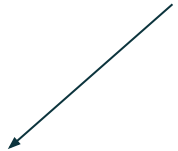
- Revisiting the earlier number, 10^{70}

How large is the search space?

- Revisiting the earlier number, 10^{70}
- For a problem having 10 monomers, 10 rules, 3 as rule size, 3 compartments and fast-slow reactions, the search space is ≈ 10^{74} rules $(2^{10} * {}^{(10+3-1)}C_{(3-1)} * 10^{(2\text{^}3 - 1)*10})$

How large is the search space?

- Revisiting the earlier number, 10^{70}
- For a problem having 10 monomers, 10 rules, 3 as rule size, 3 compartments and fast-slow reactions, the search space is ≈ 10^{74} rules $(2^{10} * {}^{(10+3-1)}C_{(3-1)} * 10^{(2\text{^}3 - 1)*10})$



fast / slow

How large is the search space?

- Revisiting the earlier number, 10^{70}
- For a problem having 10 monomers, 10 rules, 3 as rule size, 3 compartments and fast-slow reactions, the search space is ≈ 10^{74} rules $(2^{10} * {}^{(10+3-1)}C_{(3-1)} * 10^{(2\text{^}3 - 1)*10})$

fast / slow

Distribute 10 rules in 3 compartments

How large is the search space?

- Revisiting the earlier number, 10^{70}
- For a problem having 10 monomers, 10 rules, 3 as rule size, 3 compartments and fast-slow reactions, the search space is ≈ 10^{74} rules $(2^{10} * {}^{(10+3-1)}C_{(3-1)} * 10^{(2\text{^}3 - 1)*10})$