COT 2000
Foundations of Computing

Spring 2024

Lecture 22 – part 1

Exam 4 – 08/02/24

Lecture 22 – part 2

Review

Review

- Graph Theory
- Tsetlin Machines

What are Tsetlin Machines

Tsetlin Automata

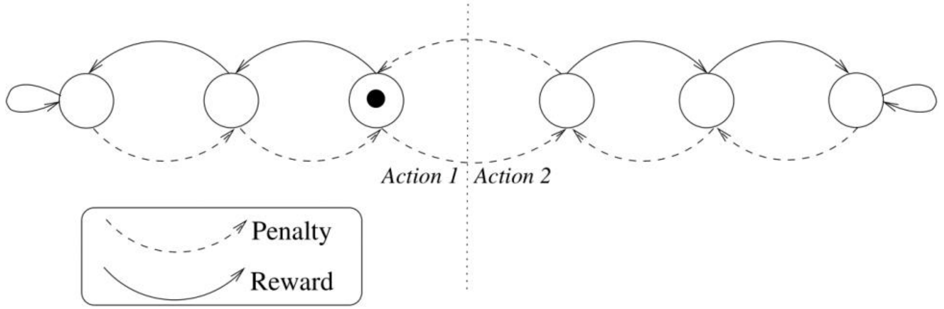


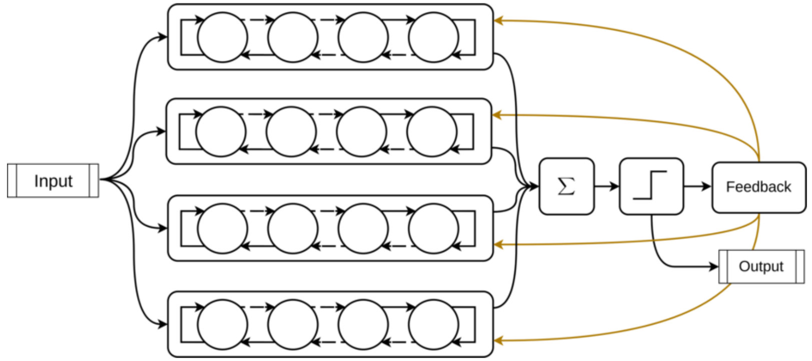
Image reproduced from https://arxiv.org/pdf/1804.01508.pdf

+

Propositional Logic

If condition then class

= Tsetlin Machine



Binarization

Literals: Input features and their negations

Dataset											
#	Four W	T.People	Wings	Yellow	Blue	**not** Four W	**not** T. People	**not** wings	**not** yellow	**not** blue	Class
1	1	1	0	0	1	0	0	1	1	0	car
2	1	1	0	1	0	0	0	1	0	1	car
3	1	1	0	1	0	0	0	1	0	1	car
4	1	1	1	0	1	0	0	0	1	0	plane
5	1	0	1	1	0	0	1	0	0	1	plane
6	0	1	1	0	1	1	0	0	1	0	plane

Rule Memory

A Tsetlin machine simulates forgetting and memorization.
Literals under position five(5) **do not participate** on the rule condition.

Maximally Memorized	10										
	9										
	8										
	7	Four W									
Memorized	6		T. People						**not** wings		
Forgotten	5				Yellow						
	4					Blue					**not** blue
	3						**not** Four W	**not** T. People		**not** yellow	
	2			Wings							
Maximally Forgotten	1										

Tsetlin Automata (columns)

Updated rule: If (Four W and T. People and not wings) **then** car

Rule Learning Algorithm

Initialization:

Set all literals to position 5

Step 1: Rule Evaluation:

Observe object and evaluate condition by assessing object literals.

Step 2: Recognize Feedback (Type 1a)

<u>If rule condition is true:</u>

Memorize the object true literals by incrementing position for p<Memorization_value.

Forget the object false literals by decrementing position for p<Forget_value.

Step 3: Erase Feedback (Type 1b)

<u>If rule condition is false</u>

Forget all literals (True or False) by decrementing position for p<Forget_value.

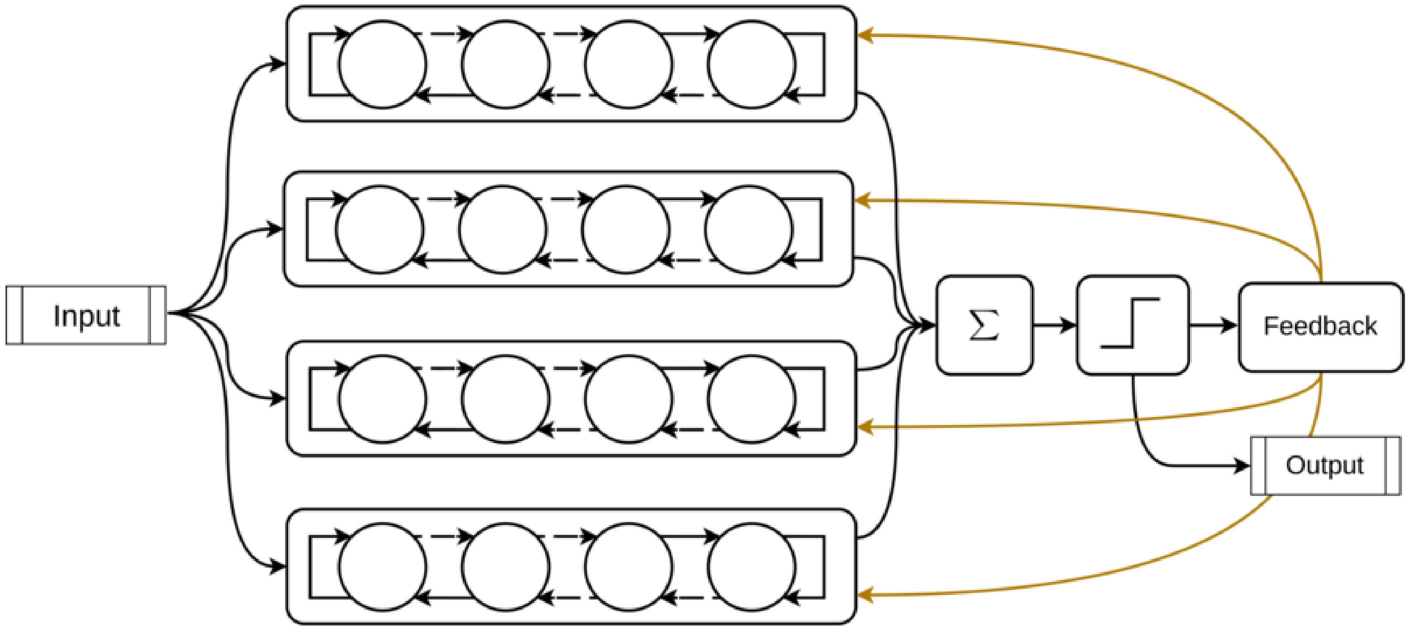
Step 4: Reject Feedback (Type 2) – (For a different class)

<u>If rule condition is true:</u>

Increment all forgotten literals (pos <=5) that are false for the object (no randomization).

<u>Else:</u> do nothing

Rule coordination



Resources

- Professor Ole-Christoffer Granmo publications: Google Scholar Page
- Book (WIP): https://tsetlinmachine.org/
- Code and Datasets on GitHub: https://github.com/cair/TsetlinMachine
- International Symposium on the Tsetlin Machine (ISTM): https://istm.no/
- Python Package: https://pypi.org/project/pyTsetlinMachine/

Lecture 22 – part 3

Course Review

Introduction

- Introduction to the **mathematical topics** that are useful in computer science.
- Foundational knowledge in **discrete mathematics** and its application in computing.

Course Review

Set Theory
symbols: “belongs to”, “subset of”
Identifying equal sets
Set-Roster Notation
Set-Builder Notation
Ordered pairs
Cartesian Product, Cartesian Plane
Relations
Functions
Arrow diagrams of relations and functions

Compound statements
Statements or propositions
Converting to symbolic form
Notation for Inequalities
Truth tables
Not
Conjunction (and)
Disjunction (or)
Exclusive (or)
Logical equivalence
Verifying logical equivalence by truth tables
De Morgan’s Laws
Negation of inequalities
Tautologies and Contradictions
Logical Equivalences Laws
Commutative, Associative, Distributive, Identity, Negation, Double Negative
Idempotent Laws, Universal bound laws, Absorption laws
Verifying logical equivalences by simplifying statement forms.
Simplifying Statement Forms

Course Review

Conditionals
Truth table of a conditional statement
Conditional Identity
The negation of a conditional statement
Contrapositive
Converse
Inverse
Biconditional

Arguments
Test a valid argument
Valid argument forms
Modus Ponens
Modus Tollen
Generalization
Specialization
Elimination
Transitivity
Proof by Division into Cases
Invalid arguments (Fallacies)
Sound argument
Converse error
Inverse error
Contradiction rule

Quantified statements
Predicates, Predicate Variables, Statements, Predicate Variables Domain
Truth Set of a Predicate
Universal Quantifier: $\forall$
Existential Quantifier: $\exists$
Universal Conditional Statement
Negation of Quantified Statements
Negation of Conditional Statements
Arguments with Quantified Statements
The rule of Universal Instantiation
Universal Modus Ponens
Universal Modus Tollens
Converse Error (Quantified form)
Inverse Error (Quantified form)

Course Review

Foundational Tools
Bash Terminal
Commands: ls, cd, pwd, cp, mv, rm,
Commands: touch, cat, man, find, grep…
Introduction to LaTeX
Concepts in Python Labs

Sequences
Sum notation
Product notation
Transforming a Sum by a Change in Variable
Upper limit Change
Factorial Notation
(n choose r) Notation
Mathematical Induction
Closed Form Expressions
Sum of Geometric Sequences
Defining sequences recursively
Fibonacci Numbers
Compound Interest
Hanoi Tower

Course Review

Functions revisited.
Function extended definition
Equality of functions
The identity function
Functions to define sequences.
Boolean Functions
One-to-one Functions
Onto Functions
One to One correspondence
Inverse functions
Composition of functions

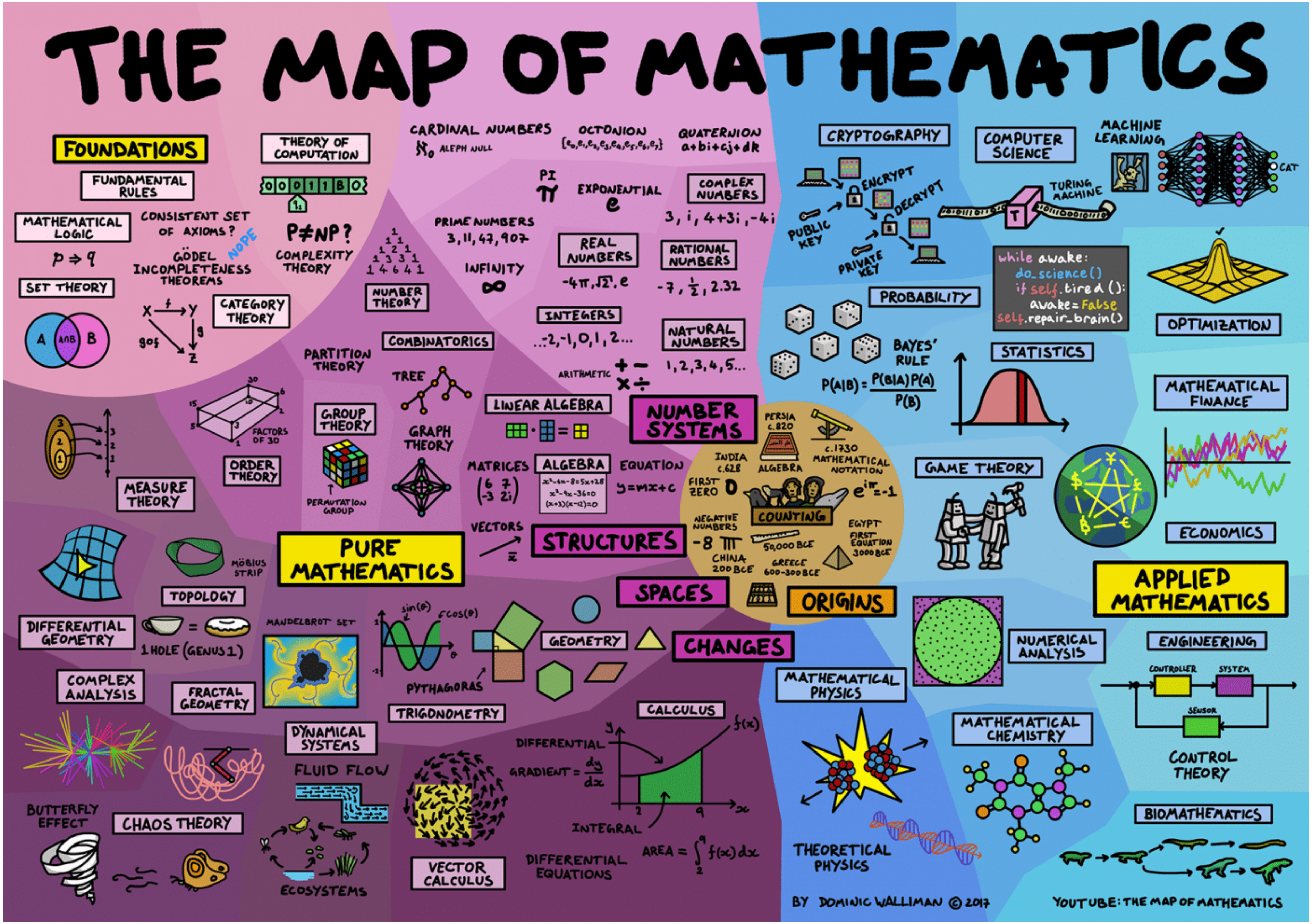
Counting and Probability
Random Process
Sample Space
Event Space
Probability
Possibility Trees
The Rule of Products
Permutation –P(n,k)
Combinations –C(n,k)

Graph Theory
Directed, Undirected,
Complete Undirected,
Multigraph
Notation and Terminology
Vertex, Edge
Initial Vertex, Terminal Vertex
Path, Vertex list
Path Length
Sub-path, proper, improper
Circuit, simple circuit
Subgraphs, Induced subgraph, Spanning subgraph.
Connected Component
Examples: Flowchart, Tournament Graph
Graph Isomorphisms

Special Topic
Tsetlin Machines

Lecture 24 – part 4

What is next ?



Domain of Science is produced by physicist Dominic Walliman

https://store.dftba.com/products/map-of-mathematics-poster

Technology/Trend	Description	Foundational Course Topics
Machine Learning and Deep Learning	Neural networks model complex functions for approximating mappings from inputs to outputs, using graph-based structures for data like social networks and recursive layers.	Mathematical Induction and Recursion, Functions and Relations, Graphs and Trees
Blockchain and Cryptocurrencies	Utilizes complex logical conditions in smart contracts and manages sets of transactions, analyzing transaction flow using graph theory.	Logic of Compound and Quantified Statements, Sets and Set Operations, Graphs
Internet of Things (IoT) and Network Security	Networks of devices and security protocols that involve complex functions and relationships, with probabilistic models for security risk analysis, and network graphs for optimizing data flow.	Functions and Relations, Graph Theory, Counting and Probability
Natural Language Processing (NLP)	Processes natural language using logical structures, sequences for text processing, and trees/graphs for parsing and semantic analysis.	Logic of Compound and Quantified Statements, Sequences, Mathematical Induction, and Recursion, Graphs and Trees
Quantum Computing	Involves quantum functions and relations like entanglements, set operations for quantum states, and uses graphs and probability for circuit analysis and interpretation.	Functions and Relations, Sets and Set Operations, Graphs and Probability
Bioinformatics	Analyzes DNA and protein sequences using recursive algorithms, constructs phylogenetic trees, and employs set operations and graphs for genomic data analysis.	Sequences, Mathematical Induction, and Recursion, Sets and Set Operations, Graphs and Trees

Subsequent Courses	Description	Extension of Foundations Topics
Algorithms and Data Structures	Focuses on efficient data handling and algorithm design, covering sorting, searching, and algorithm complexity analysis.	Builds on sequences, recursion, and mathematical induction; uses sets and graphs for representing complex data structures.
Software Engineering	Teaches principles of software development, project management, and design patterns, emphasizing the practical application of programming in large-scale projects.	Uses logic and functions to implement complex software systems; involves recursion and mathematical models in system design.
Operating Systems	Examines the underlying software that manages computer hardware and software resources, including process management, memory management, and file systems.	Relates to functions and relations in resource allocation; uses graphs and trees in the structure of file systems and processes.
Artificial Intelligence	Covers the theory and practice of building machines capable of performing tasks that typically require human intelligence, such as visual perception, decision-making, and language understanding.	Extends logic, functions, and sets to model intelligent behavior; uses graphs in learning algorithms and decision processes.
Computer Networks	Studies the protocols and architectures of computer networks, including the internet and distributed systems.	Expands on graphs and sets in the context of network topology and routing; applies functions and relations in protocol design.
Database Systems	Introduces database design and management, focusing on data models, query languages, and transaction processing.	Develops the concepts of sets, relations, and functions in the structure and querying of databases.
Theory of Computation	Explores computational theory including automata theory, computability theory, and complexity theory.	Based on mathematical logic, functions, and sets; involves graphs and trees in modeling computational processes and machines.
Cybersecurity	Focuses on the techniques for securing computer systems, including cryptography, network security, and application security.	Applies logical operations and sets in encryption and security protocols; uses probability in risk assessments and threat modeling.
Parallel and Distributed Computing	Investigates computing environments that use multiple processing elements simultaneously to solve large problems, focusing on architecture, algorithms, and networks.	Utilizes graphs and trees in the design of network architectures and algorithms; extends logic and functions in parallel processing.