Assignment #2

CS 228 (Logic in Computer Science)
TAs: Dion Reji & Thomas Biju Cheeramvelil
Due Date: ~~September 30, 2025~~ October 2, 2025

General Instructions

- This is a graded *Theory* assignment. There is NO coding involved. This assignment contains 6 questions. Question 5 is a bonus question.
- **Academic Integrity:** The institute's plagiarism policy applies strictly. Any instance of plagiarism will result atleast in zero marks for the assignment and may lead to harsher penalties, including grade drops and escalation to the DADAC. This applies equally to both giver and taker. Both teammates will be penalized equally, regardless of who plagiarised.
- **Do not use Large Language Models (LLMs) such as ChatGPT, Gemini, or Copilot to complete this assignment.** Any use will be considered a violation of academic integrity.

Submission Instructions

- You should submit a single pdf file named `<rollnumber1>_<rollnumber2>_assignment2.pdf` on Moodle before the deadline. Name your file exactly as mentioned. Any deviations from the submission format will invite atleast 50% penalty. If you are the only member in your team, the file should be named as `<rollnumber>_assignment2.pdf`.
- The deadline is **~~September 30, 2025 EOD~~ October 2, 2025 EOD**. Submissions beyond this deadline will not be considered for grading.
- Only one person from the team should submit the assignment.
- Your solution pdf must be properly typeset in LaTeX. Handwritten solutions will not be evaluated.

Other Instructions

- For every question, you should give *proper, clear* and *rigorous* explanations. Marks will be deducted for gaps in solutions and *incomplete* or *informal* reasoning.
- Make sure to include a Table of Contents at the beginning of your document, with page numbers for each question. Any question not listed in the Table of Contents may not be evaluated. (In LaTeX, you can automatically generate this by using the `\tableofcontents` command.)
- Marks will also be awarded for the presentation of your work. Please ensure your document is well-formatted, with neatly typeset mathematics and clear, readable reasoning. A well-structured and aesthetically pleasing submission might be graded more favorably :)
- The marks are not disclosed for any question. It will be decided relative to the number of correct submissions. Different questions can have different marks. Focus on the correctness and completeness of your solution.

Hogwarts Trial: The Chamber of Exact Magic

In the last assignment, we tackled puzzles. Now, it's time to step into the magical world of Hogwarts and play an enchanted game where magic will meet logic.

Hidden deep beneath the castle lies the **Chamber of Exact Magic**, a place where wizards must not only cast spells but also think with perfect precision. Legends whisper that even the bravest wizards have faltered here, undone not by lack of power, but by the smallest flaw in reasoning. Every step in the chamber demands not only courage, but a mind sharpened like a sword, for logic itself is the strongest spell. Hogwarts is hosting the *Quintalogic Quest* where Harry is selected to compete.

Professor Krishnerva McShankara has devised the secret test for this challenge. The trial is framed as a sequence of challenges, and only those who can balance logic and magic can survive.

The entire trial is described by a pair $\varphi = (\Omega, \Theta)$ where Ω is a sequence of *challenges* and Θ is a natural number representing the *potentia*, that is, the magnitude of supernatural power which must be inherited. More precisely, $\Theta \in \mathbb{N}$ [1] is the **Target Power**—the exact amount of magical energy the student must accumulate and inherit by the end of the game, while Ω represents a sequence of enchanted trials:

$$\Omega = (\mathcal{R}_1, \mathcal{R}_2, \ldots, \mathcal{R}_n),$$

where each $\mathcal{R}_i$ is the i-th challenge hidden within the chamber. Each challenge $\mathcal{R}_i$ is elaborated below.
Each challenge $\mathcal{R}_i$ involves a clash of two magical forces:

- The **Keeper of Doors** (an agent of fate) chooses one of two enchanted doors which are inscribed with *runes*[2] $\{a_i, b_i\}$. Thus, the Keeper's choice at step i contributes either a_i or b_i to the energy accumulated so far.
- The **Spellcaster** (the student) must then respond by casting one of two *spells*, whose respective magical potency are denoted respectively by $\{c_i, d_i\}$, both $c_i, d_i \in \mathbb{N}$. The chosen spell adds its corresponding potency c_i or d_i to the energy accumulated.

In each challenge, the Keeper of the Doors goes first, followed by the Spellcaster.

Therefore formally, each challenge $\mathcal{R}_i$ is represented as $(\{a_i, b_i\}, \{c_i, d_i\})$, as a tuple of two sets, each containing two natural numbers $(a_i, b_i, c_i, d_i \in \mathbb{N})$.

This process repeats for all n challenges. After the final challenge, a *syzygy* of the magical energies from the enchanted doors and the spells is formed. Since each rune and each spell have a numeric annotation, this syzygy is their sum. Formally, a journey through the chamber can be seen as an element of the set,

$$\mathcal{J} = \prod_{1 \leq j \leq n} \{a_j, b_j\} \times \{c_j, d_j\}$$

[1] Throughout this assignment $\mathbb{N}$ denotes the set of all natural numbers including *zero*
[2] A *rune* is a magical inscription, numeric in nature. More precisely, a rune is a member of $\mathbb{N}$

A journey $P \in \mathcal{J}$ is **successful** if

$$\sum_{i=1}^{|P|} P_i = \Theta$$

that is, the accumulated magical power is *exactly equal* to the Target Power Θ. Any deviation, whether surplus or shortage, awakens the chamber's ancient curses.

An **incantation schema** is a sequence of rules

$$s = (s_1, s_2, \ldots, s_n),$$

where each s_i tells the student which spell to cast in rounds $1, \ldots, i$ depending on the sequence of runes previously chosen by the Keeper, i.e.,

$$s_i : \prod_{1 \le j \le i} \{a_j, b_j\} \;\to\; \{c_i, d_i\}$$

For example, for $n = 2$, $s = (s_1, s_2)$ is an incantation schema where $s_1 = \{(a_1, c_1), (b_1, d_1)\}$ is the function representing what spell is casted depending on the first rune chosen by the Keeper, and $s_2 = \{(a_1a_2, c_2), (a_1b_2, c_2), (b_1a_2, c_2), (b_1b_2, d_2)\}$ is the function representing what spell is casted depending on the runes selected by the Keeper in the first and second challenges.

A journey conforms to an incantation schema if the spells chosen align with the rules. The set of such conforming journeys is written as Journeys(s), with

$$|\text{Journeys}(s)| = 2^n.$$

A schema is **perfect** if every conforming journey yields the Target Power. A schema is **doomed** if some conforming journey does not yield the Target Power.

Problem 1. *Given φ, decide whether the student has a* ***perfect incantation schema*** *to always achieve the Target Power, no matter which doors the Keeper opens.*

Question 1. In this part, you will have to play this game with one team member being the Keeper and the other being the Spellcaster.

There are $n = 4$ challenges. The first two challenges $\mathcal{R}_1$ and $\mathcal{R}_2$ are determined from your roll numbers. Take $a_1b_1c_1d_1$ and $a_2b_2c_2d_2$ to be the last four digits of you and your teammate's roll numbers [3].
Choose target powers $\Theta_1, \Theta_2 \in \mathbb{N}$ and design a perfect schema (for Θ_1) as well as a doomed schema (for Θ_2) by choosing $a_3, b_3, c_3, d_3 \in \mathbb{N}$ and $a_4, b_4, c_4, d_4 \in \mathbb{N}$ appropriately for each case. Justify why your perfect schema is perfect and why your doomed schema is doomed.

---

[3] If you are the single member in your team, talk to the TAs

Harry Potter thought his ordeal in the Chamber of Exact Magic was over. But alas, Professor Krishnerva had other plans. With a sly smile, she declared that the next stage would not involve potions, charms or spells, but something far more subtle: *logic itself.*

Hogwarts, of course, has a mysterious department known only to a select few: the **Department of Enchanted Logics**. Its halls are filled not with cauldrons and flying brooms, but with enchanted parchments and self-writing quills. Here, logic is not just philosophy or mathematics—it is spell-craft of the purest kind.

The only wizard still willing to guide Harry in this ordeal is our own Thomas, who recently uncovered a new kind of magical reasoning. He named it the **Thomasian Logic** (abbreviated as $\mathsf{T-logic}$ and called so henceforth), in his own honour. This new logic introduces a fragment of FOL where signature comes with structure!

Formally, the setup begins with a structure, ζ [4] which is a tuple $\langle [n], \mathcal{P}, \Lambda \rangle$ such that

- $[n]$ defines an underlying universe of elements, $U_n = \{0, \ldots, n-1\}$, i.e, the set of natural numbers (including *zero*) until and including $n-1$.
- $\mathcal{P} = \{P_i\}_{i=1}^m$ is a set of m unary predicate symbols that is available to be used in any formula.
- $\Lambda : \mathcal{P} \mapsto 2^{U_n}$ is an interpretation function. $\Lambda(P_i)$ returns a subset of the universe where the predicate P_i is true. For any i and $a \in U_n$, we say $P_i(a)$ is true iff $a \in \Lambda(P_i)$

Notice that structures in $\mathsf{T-logic}$ are not the same as they are in FOL. Unlike in FOL, a ζ structure defines the underlying universe, the available predicate symbols and also their interpretation. Just like how we need a signature to write formulae in FOL, we need to specify a ζ structure to define the syntax and semantics of formulae in $\mathsf{T-logic}$. Given a ζ structure, it corresponds to a set of formula in $\mathsf{T-logic}$ whose universe and the interpretation of predicates is fixed. We call the set of all such formulas $\mathsf{T}_\zeta - \mathsf{logic}$ if we fix a ζ structure.

Fix a ζ structure. We define the syntax of a formula in $\mathsf{T}_\zeta\mathsf{-logic}$ as follows:

- **Terms.** Terms are variables $x, y, z, \ldots$ There are no constants in ζ structures.
- **Atomic formulae.** If $P \in \mathcal{P}$ and t is a term, then $P(t)$ is an atomic formula. Additionally, $\top$ and $\bot$ are also atomic formulae.
- **Elementary sub-formulae.** Every atomic formula is an elementary sub-formula. If ψ is an elementary sub-formula, then $\neg\psi$ is also an elementary sub-formula. Likewise, if ψ_1, ψ_2 are elementary sub-formulae, $\psi_1 \wedge \psi_2$, $\psi_1 \vee \psi_2, \psi_1 \rightarrow \psi_2$ are also elementary sub-formulae.
- **Formulae.** Any formula in $\mathsf{T}_\zeta\mathsf{-logic}$ is of the form

$$Q_1 x_1\, Q_2 x_2 \cdots Q_k x_k \;\; \psi$$

  where $Q_j \in \{\forall, \exists\}$, ψ is an elementary sub-formula, $x_1, \ldots, x_k$ are all the distinct variables such that all variables in ψ should be quantified and every formula is in Prenex form.

[4]This symbol in LaTeXis `\zeta`

For example, consider the structure $\zeta = \langle [n], \mathcal{P}, \Lambda \rangle$ with $n = 2, \mathcal{P} = \{P_1, P_2\}$ and $\Lambda : \mathcal{P} \mapsto 2^{U_n}$ defined as $\Lambda(P_1) = \{0, 1\}, \Lambda(P_2) = \{1\}$.

- $\forall x \exists y \forall z\ (P_1(x) \wedge P_2(y))$ is a formula in which x and y are terms, $P_1(x)$ and $P_2(y)$ are atomic formulae, and $(P_1(x) \wedge P_2(y))$ is an elementary sub-formula, and all the variables in it are quantified. (There is no problem in quantifying more variables than those occurring in ψ.)
- $\forall x \exists y\ (x \wedge P_2(y))$ is not a formula, since x is not an atomic formula.
- $\exists y\ (P_1(x) \to P_2(y))$ is not a formula, since x is not quantified.
- $\forall x \exists y\ (P_1(x) \vee P_2(y) \vee \forall z (P_1(z)))$ is not a formula, since it is not in prenex form.

Let $a \in U_n$ and let ψ be an elementary subformula. We define $\psi[a/x]$ as follows:

1. Every occurrence of the variable x in ψ is replaced by the element a.
2. For each atomic formula $P(x)$ occurring in ψ with $P \in \mathcal{P}$, we evaluate $P(a)$ and substitute
$$P(a) \ \mapsto \begin{cases} \top & \text{if } a \in \Lambda(P), \\ \bot & \text{otherwise.} \end{cases}$$
3. After substitution, $\psi[a/x]$ is simplified using standard semantic equivalences (e.g. elimination of redundant connectives).

If x does not occur in ψ, then we set $\psi[a/x] = \psi$.

For example, let $\psi = (P_1(x) \vee P_2(y)) \to (P_2(x))$. Under a ζ structure where $P_1(0)$ and $P_2(0)$ are both true, we can see that $\psi[0/x] = \top$.

The semantic question related to a formula in $\mathsf{T}_\zeta-\mathsf{logic}$ is asking whether the formula evaluates to true. If a formula φ evaluates to true under a fixed ζ structure, we denote it as $\models_\zeta \varphi$. With this in hand, given a ζ structure, we can define the semantics of $\mathsf{T}_\zeta-\mathsf{logic}$ formulae φ as follows:

- $\models_\zeta \top$ and $\not\models_\zeta \bot$
- If $\varphi = Q_1 x_1\, Q_2 x_2 \cdots Q_k x_k\ \ \psi$ and $Q_1 = \exists$ then
  $\models_\zeta \varphi$ if there exist $a \in U_n$ such that $\models_\zeta Q_2 x_2 \ldots Q_k x_k\ \psi[a/x_1]$
- If $\varphi = Q_1 x_1\, Q_2 x_2 \cdots Q_k x_k\ \ \psi$ and $Q_1 = \forall$ then
  $\models_\zeta \varphi$ if for all $a \in U_n$, $\models_\zeta Q_2 x_2 \ldots Q_k x_k\ \psi[a/x_1]$

Notice that in the semantics above, we simplify an expression once we substitute a term with an element from the universe. We simplify it using semantic equivalences as we do in propositional logic.

For example, consider again the structure $\zeta = \langle [n], \mathcal{P}, \Lambda \rangle$ with $n = 2, \mathcal{P} = \{P_1, P_2\}$ and $\Lambda : \mathcal{P} \mapsto 2^{U_n}$ defined as $\Lambda(P_1) = \{0, 1\}, \Lambda(P_2) = \{1\}$, and the $\mathsf{T}_\zeta-\mathsf{logic}$ formula $\varphi = \forall x \exists y \forall z\ (P_1(x) \wedge P_2(y))$.

$$\begin{aligned}
\models_\zeta \varphi &\iff \text{For all } a \in U_2 = \{0,1\}, \models_\zeta \exists y \forall z (P_1(a) \wedge P_2(y)) \qquad [\because \Lambda(P_1) = \{0,1\}] \\
&\iff \models_\zeta \exists y \forall z (\top \wedge P_2(y)) \\
&\Longleftarrow \text{For } b = 1, \models_\zeta \forall z\ P_2(b) \\
&\iff \models_\zeta \forall z\ \top \\
&\iff \models_\zeta \top
\end{aligned}$$

Problem 2 (T-Satisfiablilty Problem)**.** *Given a ζ structure and a formula, φ in* $\mathsf{T}_\zeta-\mathsf{logic}$*, is φ true, i.e, $\models_\zeta \varphi$?*

Question 2. Let $\zeta = \langle [n], \mathcal{P} = \{P_i\}_{i=1}^m, \Lambda \rangle$ be a fixed ζ structure and φ be a formula in $\mathsf{T}_\zeta-\mathsf{logic}$. Give an algorithm to check satisfiability of φ in $\mathcal{O}(n^k \cdot \mathsf{length}(\varphi))$ time where k is the number of variables in φ and $\mathsf{length}(\varphi)$ denotes the length of the formula φ [5]. Your algorithm must take only space polynomial in $n, k, \mathsf{length}(\varphi)$. Clearly derive the time and space complexities for your algorithm. You should also argue the correctness of your algorithm.

Harry, still shaken from the Chamber of Exact Magic, hurried into the dimly lit quarters of the **Department of Enchanted Logics**. Rows of parchments hovered mid-air, their ink constantly rearranging into new proofs and counterexamples.

The Rise and the Fall of Thomasian Logic

At the very end of the hall stood Thomas, robes slightly disheveled, quill tucked behind his ear, eyes glowing with the thrill of discovery.

"Ah, Potter," Thomas began, voice steady and deliberate. "Krishnerva is clever, but not clever enough. What she's given you is not a mere riddle—it is a problem in my own creation, the *Thomasian Logic*, $\mathsf{T}-\mathsf{logic}$."

Harry frowned. "But… what does that mean? I thought logic was just truth and falsehood."

Thomas smirked. "Not in these halls. In $\mathsf{T}-\mathsf{logic}$, every formula carries not only its symbols but also the *structure* of the world it lives in. The question you face is to identify the truth living inside a given *structure*!"

He leaned closer, lowering his voice like he was revealing a forbidden spell.

"Krishnerva wants you to stumble here. But I, Thomas, have already coded up an enchanted solver for $\mathsf{T}-\mathsf{logic}$. All you must do, Harry, is to pose the challenge correctly: reduce your ordeal to a T-satisfiability problem. Feed it to the solver, and let the enchanted machine do its work."

Harry blinked. "So I don't duel with spells this time?"

"No," Thomas said with a wispish grin. "You duel with *structures and logics*. Now—show me if you can frame it, or Hogwarts itself may never know the answer."

Question 3. Can you help Harry do it? More precisely, given an instance of the trial, (Ω, Θ), construct a formula φ in $\mathsf{T}_\zeta-\mathsf{logic}$ for some fixed ζ such that the student (Harry) has a *perfect incantation schema* if and only if φ is *T-satisfiable*. In other words, reduce *Problem 1* to *Problem 2* for some fixed ζ. Also prove that the reduction is correct, that is, the formula you have created

[5] Length of a formula in $\mathsf{T}-\mathsf{logic}$ is the length of the matrix, i.e, the quantifier-less part of the formula.

is *T-satisfiable* iff there exist a *perfect incantation schema.*

Hint (from Prof. Dumbledore): Try reducing the trial instance to a structure ζ with $n = 2$ before attempting structures with universes of more elements.

Following the discussion on Piazza, we have decided to consider the following options for this solution:

- **Option 1:** You can carry out a reduction by choosing ζ depending on the trial instance (Ω, Θ).
- **Option 2:** You can carry out the reduction by fixing a particular ζ beforehand. In this case, your reduction should work for any (Ω, Θ).

Option 2 will be given *higher* marks as compared to **Option 1**. You **SHOULD** only write one out of the two options in your solution.

Additional instructions:

- Your formula φ should be such that $\mathsf{length}(\varphi)$ is polynomial in number of challenges (n) and the values $a_1, b_1, c_1, d_1 \ldots a_n, b_n, c_n, d_n$. Solutions that achieve better formula length complexity will be awarded more marks.
- φ should be a syntactically valid formula, i.e, it should clearly respect the syntax rules of $\mathsf{T}_\zeta-\mathsf{logic}$. Any invalid formula will be given *zero* marks. Some common mistakes are as follows (this is not an exhaustive list):
  - Taking the universe to be $\mathbb{N}$. As per the definition a ζ structure only admits universe of the form $\{0, 1, \ldots, n-1\}$ for some $n \in \mathbb{N}$. Thus, it is not possible to consider an infinite universe.
  - Using equality predicate. The only predicates allowed in a ζ structure are unary predicates (given as part of the structure).

Harry, after hours of scribbling on enchanted parchment, finally managed the reduction into a crisp *T-satisfiability problem.* His heart soared—surely, at last, victory was near.

He rushed back to Thomas, who tapped his wand against a stack of runic stones. The solver stirred awake with a low hum, glyphs spinning in the air like constellations.

"Now watch, Potter," Thomas declared proudly. "My solver will crack this in seconds."

The glyphs glowed, lines of proof forming, twisting—then collapsed into smoke. The parchment burst into laughter, mocking their effort. Thomas' solver has failed [6]. Thomas paled. "Impossible… I… I must have overlooked… something in the construction."

[6] Sorry Thomas :(

Harry's shoulders sank. The Chamber, the ordeal, the sleepless toil—all for nothing. With a heavy sigh, he turned and left the halls of Enchanted Logics, parchment crumpled in his hand.

Midway through the endless corridor of shifting staircases, a short figure intercepted him. His robes shimmered like flowing ink, and his eyes carried both sternness and quiet fire.

The Emergence of Dionian Machines

It was Dion, master of a secretive domain even more arcane than logic itself: the **Department of Transcendental Automagica**. Within those halls, automata were not machines of cogs or parchment, but living constructs of magic, tracing infinite languages through infinite skies.

Dion studied Harry's forlorn face. "Thomas's solver has failed you, hasn't it?" he said gravely. Then, with a hint of pride, he continued:

"Do not despair. For where logics stumble, automata endure. I have been forging a new model, one the wizarding world has yet to name." Then with a flick of his quill-wand, he inscribed letters of fire in the air: **Dionian Automata** (abbreviated as D−Automata and called so henceforth).

"These," he whispered, "are not bound by old rules. They traverse words and worlds alike. Perhaps your challenge belongs here, not in Thomas's fragile solver."

Harry's eyes widened. For the first time since leaving the Chamber, a spark of hope rekindled in him.

We will now introduce this new model of automata called D−Automata. In class you have seen DFA and NFA which are machines with finite control and accepts input strings by moving between states. But a key limitation of such models is that they only have a finite number of states to remember information about the input. This limits what it can recognize. For instance, there is no DFA or NFA recognizing the language $L = \{a^n b^n \mid n \in \mathbb{N}\}$[7]. A natural way to extend finite automata is to give them some extra memory. One easy way to do this is to equip the automata with *registers*. A register is a form of memory that can store a natural number starting from *zero*. Now we can do more with the automata. The automata can increase the value in the *register*, check if the *register* has a particular value or reset the *register* back to *zero*. So you can think of an automaton with *registers* as a normal finite automaton, but with some additional "memory slots" that hold numbers.

A D−Automata is a particular kind of automaton with several *registers*, say $r_1, \ldots, r_k$ each storing a non-negative natural number. But unlike independent *registers*, these *registers* are arranged in a stack-like hierarchy:

- All the *registers* can be increased freely.
- To test a *register* (say r_i), i.e to check if the *register* has a particular value, the machine must already know the values of all registers, $r_{i+1}, \ldots r_k$.
- To reset a *register*, r_i, the machine must already know the values of all registers, $r_i, \ldots r_k$.

When the automaton increases a register, it adds a specified number $n \in \mathbb{N}$ to that register. The automaton has the ability to perform equality tests for a register against a fixed value, but the

[7]NFAs can do only simple kinds of counting

stack-based restrictions must be respected. For example, testing the equality $r_k = 2 \wedge r_{k-1} = 10$ is valid because r_{k-1} can be tested once the value of r_k is fixed. On the other hand we cannot directly test the equality $r_{k-1} = 10$ because the value of r_k is not known.

The automaton may also reset a register, but it should abide by the stack-based restrictions. So, a transition can reset a register r_i only when that transition tests equality against the values of r_i through r_k. For example, r_{k-2} may only be reset if the transition also does a test of the form $r_{k-2} = n_1 \wedge r_{k-1} = n_2 \wedge r_k = n_3$.

We will now define this model formally and it will help you formalize the details mentioned above. A D−Automata is a tuple (S, R, Δ, s_0), where:

- S is a finite set of states,
- $R = \{1, \ldots, k\}$ is a set of register indices,
- $s_0 \in S$ is the initial state,
- Δ denotes the transition relation such that each transition in Δ has the form (s, E, I, X, s'), where:
  - $s, s' \in S$,
  - E is a partial function $E : R \rightharpoonup \mathbb{N}$ to denote the equality tests such that if $E(i)$ is defined for some i, then $E(j)$ must be defined for all $j \in R$ with $j > i$,
  - $I \in \mathbb{N}^k$ specifies how the registers are increased,
  - $X \subseteq R$ specifies the set of registers to be reset. It is required that $E(x)$ is defined for every $x \in X$.

A configuration of a D−Automata is a state along with values for each of the k registers. Thus, the set of all possible configurations is $S \times \mathbb{N}^k$. A configuration $(s, r_1, r_2, \ldots, r_k)$ can transition to a configuration $(s', r'_1, r'_2, \ldots, r'_k)$ iff there exists a transition $(s, E, I, X, s') \in \Delta$ such that the following conditions hold:

- For every i such that $E(i)$ is defined, we must have $r_i = E(i)$.
- For every $i \in X$, we must have $r'_i = 0$.
- For every $i \notin X$, we must have $r'_i = r_i + I_i$.

A run of the D−Automata is a sequence of configurations $c_0, c_1, \ldots, c_n$, where each c_i can transition to c_{i+1}.

Problem 3 (Reachability Problem of D−Automata)**.** *Given a* D−Automata $\mathcal{D}$*, does there exist a run from* $(s_0, 0, 0, \ldots, 0)$ *to some target configuration* $(s_t, t_1, t_2, \ldots, t_k)$*.*

We say that a D−Automata is γ-*bounded*, if there exist some $\gamma \in \mathbb{N}$, such that for every configuration $(s, r_1, r_2, \ldots, r_k)$ that can be reached by a run from $(s_0, 0, 0, \ldots, 0)$, $r_i \leq \gamma$ for all i. Additionally, a bounded D−Automata is one which is γ-bounded for some $\gamma \in \mathbb{N}$.

To concretize the idea, consider a simple D−Automata as shown in Fig 1. For this D−Automata we have $S = \{s_0, s_1, s_t\}$ with an initial state s_0 and a target state s_t (for checking reachability). We

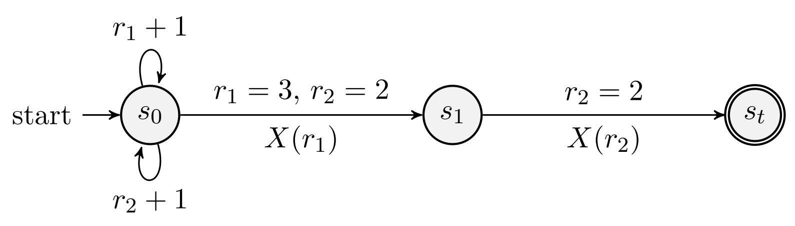

Figure 1: A simple D−Automata

have 2 registers, r_1 and r_2. Look at the transition from s_0 back to s_0 labelled with $r_1 + 1$. Any run of the automata taking this transition only increments the register r_1. Formally this transition can be denoted as $(s_0, \emptyset, (1,0), \emptyset, s_0) \in \Delta$ where $I = (1,0)$ as the register r_1 is increased by 1. For this transition, no equality tests or resets happens, thus $E = X = \emptyset$ for this. In the figure, in the transition from s_1 to s_2, two things happen: equality test for r_2 and resetting r_2, Formally this represents $(s_1, \{2 \mapsto 2\}, \emptyset, \{2\}, s_2) \in \Delta$. Notice that this figure denotes transitions in a way different from the way it is formally defined. You are free you use this notation while constructing your own automata to keep things simple for you.

Interestingly, Dion observed that the challenge that Harry is posed with can be reduced to the reachability problem of a bounded D−Automata and he has already made a tool to check the reachability of any bounded D−Automata. So, Dion asked Harry to work out this reduction so that he can go on with the trial using Dion's solver now!

Question 4. It seems like Harry is struggling with this reduction. So, let us help Harry build the automata step by step. Consider a trial (Ω, Θ) with only two challenges, i.e, $\Omega = (\mathcal{R}_1, \mathcal{R}_2)$ where $\mathcal{R}_1 = (\{a_1, b_1\}, \{c_1, d_1\})$ and $\mathcal{R}_2 = (\{a_2, b_2\}, \{c_2, d_2\})$. Construct a D−Automata and specify a target configuration such that Harry has a *perfect spellcasting plan* if and only if ~~the D−Automata reaches some target configuration~~ the target configuration is reachable from the initial configuration in the D−Automata. You should basically reduce this trial with two challenges to a Reachability problem of D−Automata. You should draw the automata, identify the right target configuration and explain your construction. Answers without *formal* explanation will get *zero* marks even if the automata is correct.

- Your construction should work for any choice of $a_1, b_1, c_1, d_1, a_2, b_2, c_2, d_2$.
- For this question, there is no specific bound on the number of states, registers and transitions. However, a more elegant and concise automata will get more marks. As a small hint, it is possible to construct an automata with less than 20 states, 12 registers and less than 4 transitions out of each states. These are just some rough numbers to help you with. We won't cut marks unless your automata is too messy and complicated.
- You are allowed to attach images of hand drawn automata for this question. But make sure to draw it neatly and show all transitions clearly. In case of ambiguity we will cut marks. Additionally, you can also describe the automaton mathematically using (S, R, Δ, s_0) notation, but make sure to describe it completely.

Question 5. **(Bonus Question)** Even though Harry can solve the previous reduction, it is not enough because, Professor Krishnerva is brilliant. She can give a trial with any number of challenges. So, can you help Harry with a general reduction? That is, given any instance of the trial, (Ω, Θ), construct a D−Automata and specify a target configuration such that Harry has a *perfect spellcasting plan* if and only if ~~the D−Automata reaches some target configuration~~ the target configuration is reachable from the initial configuration in the D−Automata.

- For this question the automata you construct should have number of states, registers and transitions polynomial in terms of n.
- Since it is a bonus question, it will be binary graded. Only completely correct solutions get full credits.
- Your solution should contain the following:
  - Complete construction of the automaton.
  - Proof of correctness of your construction.
  - Proof that the size of the automaton in terms of number of states, registers and transitions is polynomial in n.

At last, Harry pieced together the reduction, every step aligned like the gears of a perfect spell. With determination, he carried it into the vaulted chambers of the Department of Transcendental Automagica. Dion, with his calm but commanding presence, listened to Harry's explanation. Then, without hesitation, he lifted the parchment and tossed it into the heart of his creation — the gleaming, whirring construct known as the Dionian Automata.

The chamber lit up. Sigils blazed on the walls, runes rearranged themselves into flowing patterns, and the automaton began its dance across infinite languages. Harry held his breath.

Then—*viola!* The symbols converged into a single, shining truth. The problem was solved. The reduction held. The trial, once thought impossible, had yielded to the new automata. Dion gave the faintest of nods, a mark of rare approval. "You see, Potter," he said, "where brute force falters, structure prevails."

Harry, beaming, slid the enchanted tool gently into his bag. His footsteps echoed joyously down the ancient halls of Hogwarts, lighter than they had been in weeks. He had endured the Chamber of Exact Magic, bested the twisted puzzles of Thomasian Logic, and now triumphed with the aid of Dionian Automata. For the first time, he felt not just like a survivor of trials, but a wizard who had mastered the art of reason itself.

The trial was beaten. The halls of Hogwarts would never look the same again.

Question 6. Write down the contribution of each teammate clearly. Every team member *should* understand the solutions to all the problems. If your teammate did NO work, mention it clearly and we will grade accordingly. If you are the single member, you can skip this question.
