Formal models are magic

Davide Catta[1], Wojciech Penczek[2] and Laure Petrucci[1]

[1]LIPN, CNRS UMR 7030, Université Sorbonne Paris Nord, Villetaneuse, FRANCE
[2]Institute of Computer Science, Polish Academy of Sciences, Warsaw, POLAND

Abstract
In this work, we investigate a variant of the Four Glasses Puzzle (FGP) through the lens of formal verification. Although the puzzle appears to be unsolvable at first glance, formal verification enables the discovery of a correct solution. We demonstrate how the model can be constructed and reduced through progressive abstractions, ultimately yielding a concise and comprehensible model that captures all possible puzzle configurations. This last model is easy to understand, even for non-specialists. Finally, by analysing the strategic capabilities of the player, we present a solution to the puzzle.

1. Introduction

As software and hardware systems grow in complexity, ensuring that they behave as intended has become a central concern. Formal verification addresses this by providing precise tools to describe what a system should do, and to check—exhaustively—whether it does it. Unlike testing, which considers selected scenarios, verification explores all possible behaviours based on a model of the system.

The rise of autonomous and distributed systems—such as robots, trading agents, and self-driving cars—has made this need even more urgent. These systems act independently, interact with one another, and adapt to changing environments. Verifying them requires reasoning not only about sequences of events, but also about goals, knowledge, and strategies. The tools involved draw from logic, game theory, and automata theory, adapted to the challenges of interaction and decentralization.

In this paper, we use a variant of the Four Glasses Puzzle (FGP) [1] — also known as the Blind Bartender Problem — to illustrate some core techniques in the formal verification of multi-agent systems [2, 3, 4]. The puzzle provides a minimal yet expressive setting to present three typical steps of the method: modelling, abstraction, and strategy synthesis. Modelling involves giving a precise structure to the system, specifying states, actions, and their transitions. Abstraction reduces the system to what is relevant for verification, omitting unnecessary details. Strategy synthesis is the automatic construction of a plan that ensures a given goal is achieved, when possible. Our aim is to make these concepts tangible through a simple yet non-trivial example.

The Four Glasses Problem The *Four Glasses Puzzle*, also known as the *Blind Bartender's Problem*, gained widespread attention when it was popularized by Martin Gardner in his "Mathematical Games" column in the February 1979 issue of *Scientific American*. This deceptively simple logic puzzle challenges participants to devise a foolproof strategy under strict constraints, making it a classic example of recreational mathematics.

The problem can be presented as follows: four glasses are placed at the corners of a square tray. Each glass is either upright or upside down. A blindfolded person must manipulate the glasses to make them all face the same direction (either all up or all down), subject to the following rules:

- On each turn, the person may inspect two adjacent glasses and choose to flip either, both, or neither.

PeNGE'25: Petri Net Games, Examples and Quizzes for Education, Contest and Fun. June 24, 2025, Paris, France
davide.catta@lipn.univ-paris13.fr (D. Catta); penczek@ipipan.waw.pl (W. Penczek); laure.petrucci@lipn.univ-paris13.fr (L. Petrucci)
0000-0001-8656-3274 (D. Catta); 0000-0001-6477-4863 (W. Penczek); 0000-0003-3154-5268 (L. Petrucci)

- After each turn, the tray is randomly rotated, eliminating any positional reference.
- The solution must guarantee success in a finite number of steps, without relying on luck.

The puzzle can be analyzed through the lens of game theory and theoretical computer science. The problem can be reinterpreted as a two-player game where Player 1 (the agent) attempts to align the glasses using limited actions, and Player 2 (the environment) introduces uncertainty by randomly rotating the tray after each move. The agent's lack of positional knowledge creates a scenario of *imperfect information*, akin to games like poker or robotics under sensor noise.

The puzzle's elegance has inspired numerous generalizations. In this paper, we focus on a particularly devilish variant of the original one in which the blindfolded person additionnally wears boxing gloves and cannot sense glass orientations—only flip them blindly. This removes tactile feedback, transforming the problem into a purely combinatorial game.

Purpose of this work In this paper, we model the aforementioned devilish variant of the Four Glasses Puzzle through the lens of *formal verification of multi-agent systems*.

This approach highlights how abstract logical frameworks can be grounded in tangible, intuitive problems. By distilling complex ideas like strategy invariance and partial observability into the bartender's challenge, we bridge recreational mathematics and formal verification, offering a concrete entry point to explore theoretical computer science principles. What makes the bartender problem interesting from a verification perspective is that the bartender has no information about the current state of the system. Whatever action they choose must work regardless of how the glasses are actually arranged. This means the strategy cannot depend on the state—it must be the same for all situations that are indistinguishable from the agent's point of view. In technical terms, the strategy must be uniform.

This idea has a clear analogue in distributed and autonomous systems. In many real-world cases, agents operate with limited or no access to the global state. A robot, a sensor, or a protocol—all must follow a fixed plan that works in all the situations they cannot tell apart. In this sense, finding a uniform winning strategy is like designing a local, deterministic algorithm that guarantees the right outcome under uncertainty.

The bartender puzzle, in this version, becomes a small but sharp illustration of this challenge. It shows how strategy synthesis under partial information connects logic, games, and system design.

2. Formal Models at Work

We start with introducing a variant of of the Four Glasses Problem.

2.1. A Devilish Variant of the Four Glasses Problem

We consider a variant of the Blind Bartender with Boxing Gloves problem [1]. The setup is as follows: four glasses are placed at the corners of a square tray, and each glass is either upright or upside down. A blindfolded person wearing boxing gloves aims to make all the glasses face the same direction.

On each turn, the person may choose one of three actions: flip a single random glass, flip two adjacent random glasses, or flip two random glasses located on opposite corners (i.e. along a diagonal). After declaring the desired action, the corresponding glasses are flipped—but the person receives no feedback and does not know which glasses were affected. If the goal has not been achieved after the move, the tray is rotated by an unknown angle.

2.2. Formal modelling

Given this problem description, we can provide a precise mathematical model in the form of a directed graph with labelled edges. The states of the graph will represent one of the 16 possible configurations of the glasses (each glass can be either up or down, yielding 2^4 possible combinations). Each edge between two states will be labelled with an action. The input state represents the state from which a given action can be performed, while the output state represents one possible result of applying that action to the given input.

Definition 1 (Game)**.** *Given a finite set $\mathcal{A}$ of actions, we define a game $\mathcal{G} = (S, \{\xrightarrow{a}\}_{a\in\mathcal{A}}, S_I, S_F)$ as a directed graph, where S is a set of nodes (states), $\{\xrightarrow{a}\}_{a\in\mathcal{A}}$ is a set of edges (arcs) labelled by actions, and $S_I \subseteq S$ is a set of initial states and $S_F \subseteq S$ is a set of final (winning) states. We say that a game is a* turn-based two player game *if the set S is the union of two disjoints set S_1 and S_2 and for each $s, s' \in S$ if $s \xrightarrow{a} s'$, then $s \in S_1 \wedge s' \in S_2$ or $s \in S_2 \wedge s' \in S_1$.*

A **play** π in a game is any non-empty alternated sequence $s_0, a_0, s_1, \ldots$ of states and actions such that $s_0 \in S_I$, and $s_i \xrightarrow{a_i} s_{i+1}$ for each index $i \geq 0$ of the play. Moreover, we require that any play ends in a state if the play is finite, and we call histories finite plays.

A **strategy** for a player i is a function $\mathcal{S}$ that takes as input a history ending in a state $s \in S_i$ and outputs an action a that labels an edge leaving s. A play $\pi = s_0, a_0, s_1, \ldots$ is **compatible with a strategy** $\mathcal{S}$ iff for every index k of the play, if $s_k \in S_i$ then $\mathcal{S}(s_0, a_0, \ldots, a_{k-1}, s_k) = a_k$. For each state s, by $Out(s, \mathcal{S})$ we denote the set of all maximal (w.r.t. the prefix order) plays starting in s that are compatible with the strategy $\mathcal{S}$.

We say that a strategy for a player is **uniform** if it returns the same output for plays that are indistinguishable to the given player. We say that a uniform strategy is **winning** if the paths compatible with it, starting from any initial state, are all finite.

Our bartender acts without having any information about the current state of the game. Therefore, his strategy should not depend on the current configuration of the game: in other words, the strategy must return the same output for all plays that it cannot distinguish. Remark that in our context two plays are indistinguishable for the bartender if they have the same length, and that all actions are enabled in all three indistinguishable states (see the final attempt), and no action is enabled in the winning states.

First attempt. Now that we have defined the mathematical model for representing the game, let us proceed to its concrete representation. Given that the bartender problem involves two players, it is natural to model it as a turn-based two-player game. To do so, we need to define two disjoint sets of states, each representing the possible configurations of the glasses on the tray. A natural choice is to consider for each of the two sets, the 2×2 matrices over a two-elements set.

Thus we define the first version of the bartender game, $\mathcal{G}_B^1$, as follows. The set of states S is defined as the union of the set S_1 of 2×2 matrices over $\{0, 1\}$ and the set S_2 of 2×2 matrices over $\{u, v\}$. The set of actions $\mathcal{A}$ consists of two main types:

$$\mathcal{A} = \{\mathrm{t1}, \mathrm{t2a}, \mathrm{t2d}\} \cup \{\mathrm{r_i} \mid i \in \{0, 90, 180, 270\}\},$$

where t1 corresponds to flipping a single glass, t2a to flipping two adjacent glasses, and t2d to flipping two diagonal glasses. Each $\mathrm{r_i}$ element represents the rotation of the tray by i degrees, with $i \in \{0, 90, 180, 270\}$.

To model the transitions between these configurations, we define the edges, which describe how the game state changes due to the actions of the bartender and of the other player. Consider a map f that maps 0 to u and 1 to v, we define the following rules for the transitions, for any matrix M different from the unit and the zero one:

- $M \xrightarrow{\mathrm{t1}} M'$ with $M \in S_1$ iff there exists exactly one position (i, j) such that $f(M_{ij}) \neq M'_{ij}$, and for all other positions $(k, l) \neq (i, j)$, we have $f(M_{kl}) = M'_{kl}$. In other words, a single glass is flipped, and all other glasses remain untouched.

- $M \xrightarrow{\mathrm{t2d}} M'$ with $M \in S_1$ iff there exists a diagonal $D \in \{\{(1, 1), (2, 2)\}, \{(1, 2), (2, 1)\}\}$ such that $f(M_{ij}) \neq M'_{ij}$ for all $(i, j) \in D$, and for all other positions $(k, l) \notin D$, we have $f(M_{kl}) = M'_{kl}$. This means that two glasses on a diagonal are flipped.

- $M \xrightarrow{\text{t2a}} M'$ with $M \in S_1$ iff there exists a pair of adjacent positions $A \in \{\{(1,1),(1,2)\},\{(2,1),(2,2)\},\{(1,1),(2,1)\},\{(1,2),(2,2)\}\}$ such that $f(M_{ij}) \neq M'_{ij}$ for all $(i,j) \in A$, and for all other positions $(k,l) \notin A$, we have $f(M_{kl}) = M'_{kl}$. This corresponds to flipping two adjacent glasses, either horizontally or vertically.

- $M \xrightarrow{\text{r}_\text{i}} M'$ with $M \in S_2$ iff M' is the result of rotating the matrix M by i degrees, modulo the function f.

For example: $\begin{pmatrix} 0 & 0 \\ 1 & 1 \end{pmatrix} \xrightarrow{\text{t1}} \begin{pmatrix} u & v \\ v & v \end{pmatrix}$; $\begin{pmatrix} u & v \\ v & v \end{pmatrix} \xrightarrow{\text{r}_{180}} \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$.

The set of initial states of $\mathcal{G}_B^1$ is S_1 without the zero and the unit matrices. The set of final states contains exactly the zero and the unit matrices.

Remark: The only states with no outgoing transitions are the final states, corresponding to the bartender's tray with all glasses upside-down (the zero matrix) and the one with all glasses upright (the unit matrix), i.e. those states where the bartender wins. Given this, asking for a solution to the blind bartender problem with boxing gloves is equivalent to asking the following:

> **Question**: is there a uniform strategy $\mathcal{S}$ for player 1 such that for any initial state s of $\mathcal{G}_B^1$ we have that each $\pi \in Out(s, \mathcal{S})$ is finite?

Note that this solution has lots of states and transitions, therefore, we do not visualize it.

Second attempt Although $\mathcal{G}_B^1$ faithfully represents the turn-based structure of the original game, it is somewhat forced. We had to artificially duplicate the number of states in order to have one set of states from which the bartender can make a move and another set of states from which the other player can act. Fortunately, we can easily address this modeling issue. For any triple of states in the model described above, x, y, and z, if $x \in S_1$ and $x \xrightarrow{a} y \xrightarrow{b} z$, then $z \in S_1$, and if $x \in S_2$ and $x \xrightarrow{a} y \xrightarrow{b} z$, then $z \in S_2$, since the game is turn-based.

Thus, we can consider a simpler model, which we will call $\mathcal{G}_B^2$, where the set of states is simply S_1, the set of actions consists of the three actions of the bartender, the set of initial states is equal to the set of initial states in $\mathcal{G}_B^1$ and $x \xrightarrow{a} z$ in $\mathcal{G}_B^2$ if and only if $x \in S_1$ and $x \xrightarrow{a} y \xrightarrow{b} z$ in $\mathcal{G}_B^1$ for some $y \in S_2$ and some action b.

Given a play π on $\mathcal{G}_B^1$, let π^B denote its subsequence containing only states in S_1 and actions of the bartender. We have that for every play π in $\mathcal{G}_B^1$, π^B is a play in $\mathcal{G}_B^2$, and conversely, for any play ρ in $\mathcal{G}_B^2$, there exists a play π in $\mathcal{G}_B^1$ such that $\rho = \pi^B$. Furthermore, note that any strategy $\mathcal{S}$ for the bartender on $\mathcal{G}_B^1$ can be applied to $\mathcal{G}_B^2$. From the observations above, we immediately obtain the following:

Proposition 1. *There is a winning strategy on $\mathcal{G}_B^1$ if and only if there is a winning strategy on $\mathcal{G}_B^2$.*

Even if $\mathcal{G}_B^2$ is smaller than $\mathcal{G}_B^1$, as it shows only the bartender, it is still huge (16 states and many transitions), so we do not show it.

Third (and final) attempt The model just described is certainly more efficient than the first one. In this model, we did not have to unnecessarily multiply the number of states, and we simply modelled the second player by introducing greater non-determinism in the transition relation: given an action and a state, the set of possible outcomes consists of all possible rotations of the tray after the action is performed. However, this model is still not optimal, and we can do better. From the bartender's perspective, the two configurations where all glasses are either up or down are equivalent: in both cases, the bartender wins. Similarly, the two configurations with two glasses facing one way along a diagonal and two facing the other way are equivalent for the bartender, and so on. To be short, in the Four Glasses with Boxing Gloves Problem, the equivalence between configurations is determined by the symmetry of the 2x2 grid of glasses. This includes rotations (0°, 90°, 180°, 270°) and flipping (inverting the positions of all glasses). The bartender's goal is to move the glasses in a way that either all are facing up or all are facing down, regardless of their initial orientation. Rotating or flipping the grid does not fundamentally change the problem because the relative positions of the glasses and the actions needed to achieve the goal remain the same. As such, configurations that are rotations or reflections (flips) of each other are considered equivalent, meaning that they require the same strategy to solve.

To sum up:

> ($\equiv_B$) From the bartender's point of view, two tray configurations M and M' (with four glasses on the tray) are equivalent if M' can be obtained by rotating M by 0°, 90°, 180°, or 270°, and flipping all the glasses (changing each glass from up to down, or the other way around).

The equivalence classes generated by the relation defined above can be visualized in Table 1.

Class	**Representatives**
All glasses equal	●●/●● ○○/○○
Checkerboard pattern	●○/○● ○●/●○
Two adjacent equal glasses	●●/○○ ○○/●● ●○/●○ ○●/○●
Three equal glasses, one different	●●/●○ ●●/○● ●○/●● ○●/●● ●○/○○ ○●/○○ ○○/○● ○○/●○

Table 1
Equivalence Classes of Glasses on the Tray

Given this, we can define the third (and final) model of the blind bartender with boxing gloves. Let us call this model $\mathcal{G}_B^3$. The set of actions $\mathcal{A}$ for $\mathcal{G}_B^3$ consists of the three actions of the bartender, the set of states only counts four states ℓ_1, ℓ_2, ℓ_3, and ℓ_4. Each of these states represents an equivalence class of glasses on the tray modulo the above-defined relation. In particular,

1. ℓ_1 represents the class in which three glasses are equal and one different,
2. ℓ_2 represents the class of checkerboard patterns,
3. ℓ_3 represents the class of configurations with two adjacent equal glasses,
4. ℓ_4 represents the class of configurations in which all glasses are equal.

The set of initial states is equal to $\{\ell_1, \ell_2, \ell_3\}$. Finally $\ell_i \xrightarrow{a} \ell_j$ iff there is a state s in the class represented by ℓ_i and a state s' in the class represented by ℓ_j such that $s \xrightarrow{a} s'$ in $\mathcal{G}_B^2$. A representation of $\mathcal{G}_B^3$ is shown in Figure 1. Given the definition of the model, it is fairly immediate to prove the following:

Proposition 2. *There is a winning strategy on $\mathcal{G}_B^1$ if and only if there is a winning strategy on $\mathcal{G}_B^3$.*

So, solving the Blind Bartender with Boxing Gloves Problem comes down to finding a uniform strategy—a fixed sequence of actions that does not rely on knowing the current state—guaranteeing that, no matter the starting configuration, the system eventually reaches the goal state ℓ_4.

This problem can be formalised in strategic logics (see [5] for an introduction) as $\langle\langle bartender \rangle\rangle F \ell_4$. This formula states that there exists a strategy (operator $\langle\langle\ \rangle\rangle$) of the $bartender$ model which ensures that ℓ_4 is eventually reached (operator F). We consider the following semantics:

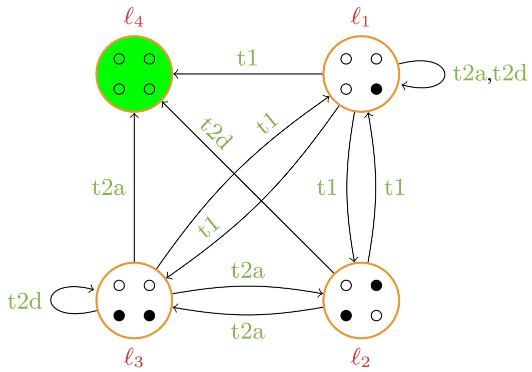


Figure 1: The blind bartender with boxing gloves problem

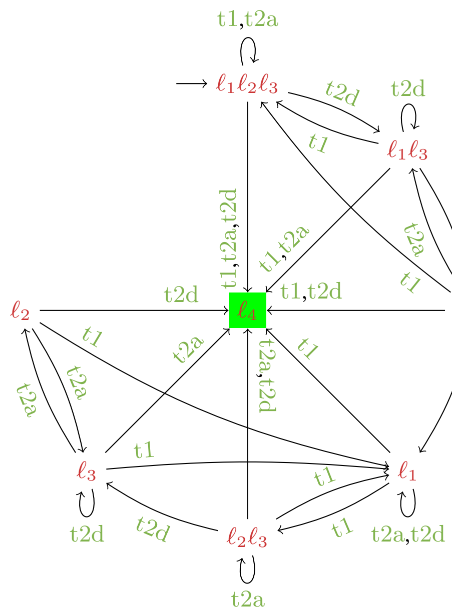


Figure 2: The vision of the bartender behaviour

- imperfect information: the bartender does not know the current state, but he is told if winning (hence ℓ_1, ℓ_2 and ℓ_3 are indistinguishable).
- perfect recall: the bartender remembers the sequence of actions he has played.

Let us analyse the behaviour of the system, starting from $\ell_1\ell_2\ell_3$ (remember the bartender does not know the actual starting point). It is pictured in Fig. 2. Starting from this state, all operations are possible from either ℓ_1, or ℓ_2, or ℓ_3. When taking t1, the bartender can win by getting directly in ℓ_4, which is possible from ℓ_1 only. Or it can lead to ℓ_1 from ℓ_2 and ℓ_3 ; or to ℓ_2 from ℓ_1 ; or to ℓ_3 from ℓ_1. To summarise, from the initial state t1 gets either to win of to any of the three states ℓ_1, ℓ_2 and ℓ_3, without yet any possibility to distinguish them, hence the self-loop on the initial state. Consider now t2d. If in ℓ_1, we stay there. From ℓ_2 it leads to the winning state ℓ_4. If in ℓ_3, we stay there. Hence, either we get to win, or we know that we are in ℓ_1 or ℓ_3, thus the state $\ell_1\ell_3$ in Fig. 2. Repeating such a reasoning from $\ell_1\ell_3$ (and so on) allows for constructing the graph in Fig. 2.

The idea here is to find a sequence of actions that necessarily goes to ℓ_4. Notice that in the graph of Fig. 2, the bartender may gain knowledge (e.g. can know that the current state is ℓ_1) but due to non-determinism, may again lose exact information (e.g. $\ell_2\ell_3$ is a successor of ℓ_1). Choosing an action that may get you to an already visited state in the graph does not help, so we will instead privilege progress. Thus, initially, the bartender will choose action t2d: either the game is immediately won, or the state is ℓ_1 or ℓ_3, but they cannot be distinguished at this stage. From there, choosing t2d does not help, choosing t1 may win (if the bartender is lucky) or get back to the initial situation. Hence the bartender chooses to progress with t2a. We continue in a similar manner and obtain the following strategy for winning:

$$\text{t2d}, \text{t2a}, \text{t2d}, \text{t1}, \text{t2d}, \text{t2a}, \text{t2d}$$

Note that this strategy is minimal, but other ones exist, including for example a cycle in the graph.

Several semantics exist for strategic abilities (see [5]), some of them being undecidable in general. Depending on the semantics, a winning strategy could be computed using a tool that supports reasoning under imperfect information, such as MCMAS [6] or STV [7, 8].

3. Modelling with Petri Nets

Figure 3 presents two Petri nets models for the Four glasses problem. The Petri net in Fig. 3a is a straightforward adaptation of the automaton in Fig. 1, using the same labels for places and transitions. The Coloured Petri Net in Fig. 3b features two places: one where the game is won (corresponding to ℓ_4 in the previous models), and one where the game is not won (corresponding to the other three states). There, tokens hold a value, indicating the state of the game, i.e. in $\{\ell_1, \ell_2, \ell_3, \ell_4\}$. Functions on the arcs are such that:

$$\begin{array}{ll} t1(\ell_1) \in \{\ell_2, \ell_3\} & t2a(\ell_1) = \ell_1 \\ t1(\ell_2) = \ell_1 & t2a(\ell_2) = \ell_3 \\ t1(\ell_3) = \ell_1 & t2a(\ell_3) = \ell_2 \end{array}$$

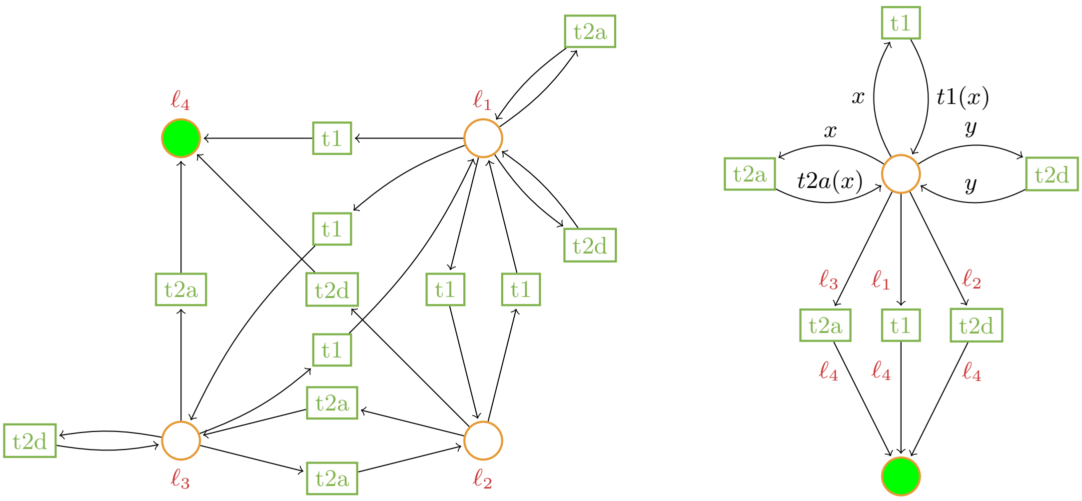
and variable domains are : $x \in \{\ell_1, \ell_2, \ell_3\}$, $y \in \{\ell_1, \ell_3\}$.

One could argue that the transition t2d which is not leading to the winning state is useless, but this is not the case: it represents a *permitted* action on ℓ_1 and ℓ_3 which has no effect.

Another possibility would have been to use a single place (with a slight modification of the functions used on arcs), and winning the game would resume to the reachability of marking ℓ_4 in the sole place.

4. Conclusions

In this paper, we have presented progressively abstracted models for the Four Glasses Puzzle and demonstrated how they can be used to derive a winning strategy for this seemingly unsolvable problem. The puzzle has been employed with the general public—first as a magic trick, and then as a teaching



(a) Petri net (b) Coloured Petri net

Figure 3: Petri net models for the bartender

tool to illustrate how formal modelling and verification can be leveraged to uncover a solution, thereby demystifying mathematical approaches.

As possible extensions, this methodology could be adapted to incorporate timing constraints [9, 10, 11], enabling the discovery of strategies that succeed within a specified time frame. Another interesting direction would be to generalize the puzzle to accommodate any number of glasses. In such a case, Coloured Petri Nets would provide a suitable framework for encoding tray configurations as tuples attached to tokens, similar to what was presented in Section 3.

Acknowledgments

This work was supported by: CNRS IRP "Le Trójkąt", the ANR-22-CE48-0012 project BISOUS, the PHC Polonium project MoCcA (BPN/BFR/2023/1/00045).

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] R. Ehrenborg, C. M. Skinner, The blind bartender's problem, Journal of Combinatorial Theory, Series A 70 (1995) 249–266.

[2] C. Baier, J.-P. Katoen, Principles of Model Checking, MIT Press, 2008.

[3] E. Clarke, B.-H. Schlingloff, Model checking, in: A. Robinson, A. Voronkov (Eds.), Handbook of Automated Reasoning, Elsevier, 2001, pp. 1635–1790.

[4] E. Clarke, O. Grumberg, D. Peled, Model Checking, MIT Press, 1999.

[5] É. André, W. Penczek, L. Petrucci, Model Checking Timed and Strategic Properties, Transactions on Petri Nets and Other Models of Concurrency (2025). To appear.

[6] A. Lomuscio, H. Qu, F. Raimondi, MCMAS: An open-source model checker for the verification of multi-agent systems, International Journal on Software Tools for Technology Transfer 24 (2015) 84–90.

[7] D. Kurpiewski, W. Jamroga, M. Knapik, STV: model checking for strategies under imperfect information, in: E. Elkind, M. Veloso, N. Agmon, M. E. Taylor (Eds.), Proceedings of the 18th International Conference on Autonomous Agents and MultiAgent Systems, AAMAS '19, Montreal, QC, Canada, May 13-17, 2019, International Foundation for Autonomous Agents and Multiagent Systems, 2019, pp. 2372–2374.

[8] D. Kurpiewski, M. Kaminski, W. Jamroga, ????

[9] W. Penczek, A. Pólrola, Advances in Verification of Time Petri Nets and Timed Automata: A Temporal Logic Approach, volume 20 of *Studies in Computational Intelligence*, Springer, 2006.

[10] W. Penczek, Model checking of strategic timed temporal logics (invited talk), in: M. Köhler-Bussmeier, D. Moldt, H. Rölke (Eds.), Proceedings of the 2023 International Workshop on Petri Nets and Software Engineering (PNSE 2023) co-located with the 44th International Conference on Application and Theory of Petri Nets and Concurrency (PETRI NETS 2023), June 27, 2023, Lisbon, Portugal, volume 3430 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023.

[11] J. Arias, W. Jamroga, W. Penczek, L. Petrucci, T. Sidoruk, Strategic (timed) computation tree logic, in: N. Agmon, B. An, A. Ricci, W. Yeoh (Eds.), Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems, AAMAS 2023, London, United Kingdom, 29 May 2023 - 2 June 2023, ACM, 2023, pp. 382–390.