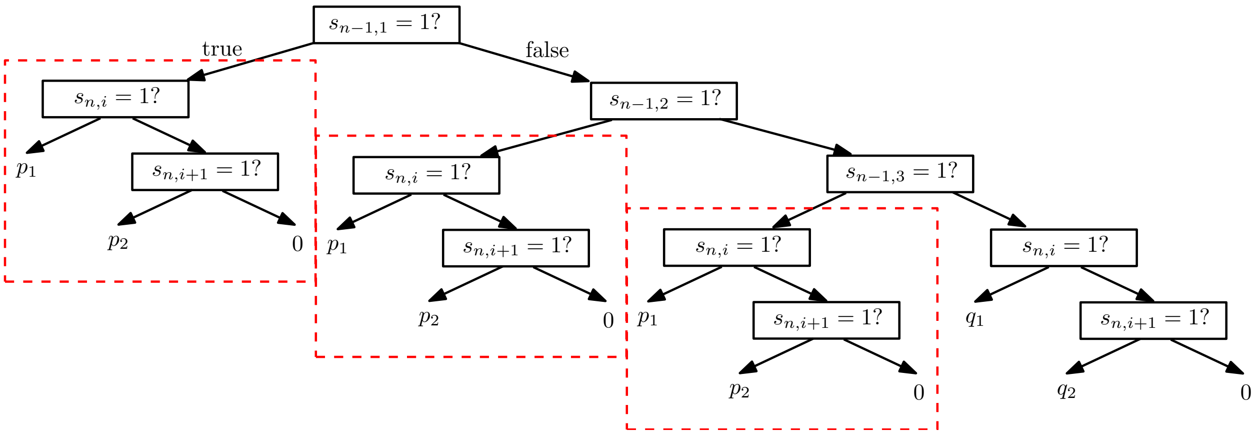
E-Companion A: Context-Specific Independence

A *decision tree* represents a function $f : \mathcal{S} \rightarrow \mathbb{R}$ as a tree that branches upon the values of some of the components of $s \in \mathcal{S}$ and that contains the function values $f(s)$ as leaf nodes. We can exploit context-specific independence in the transition probabilities $p_n(s'_n \mid s, a)$ by constructing a separate tree for each sub-state $n \in [N]$, each possible value $s'_n \in \mathcal{S}_n$ of that sub-state and each action $a \in \mathcal{A}$. Likewise, we can exploit context-specific independence in the rewards $r(s, a)$ by constructing a separate tree for each reward component r_j, $j \in [J]$, and each action $a \in \mathcal{A}$. The size of a decision tree's description scales with its number of (internal and leaf) nodes.

An *algebraic decision diagram* (ADD) $f : \mathbb{B}^I \rightarrow \mathbb{R}$ of binary vectors $x = (x_1, \ldots, x_I)$ is defined recursively as follows: An empty function $f : \varnothing \rightarrow \mathbb{R}$ satisfying $f() = c$, for any $c \in \mathbb{R}$, is an ADD. If two functions $f', f'' : \mathbb{B}^{I-i} \rightarrow \mathbb{R}$ are ADDs of binary vectors $(x_{i+1}, \ldots, x_I)$, then the function $f : \mathbb{B}^{I-i+1} \rightarrow \mathbb{R}$ satisfying $f(x_i, \ldots, x_I) = x_i \cdot f'(x_{i+1}, \ldots, x_I) + (1 - x_i) \cdot f''(x_{i+1}, \ldots, x_I)$ is an ADD of binary vectors $(x_i, \ldots, x_I)$. For each sub-state $n \in [N]$ and each possible value $s'_n \in \mathcal{S}_n$ of that sub-state, the associated one-step transition probabilities $p_n(s'_n | s, a)$ can be represented as a collection of ADDs of the state s, one for each action $a \in \mathcal{A}$. Likewise, we can exploit context-specific independence in the rewards $r(s, a)$ by constructing a separate ADD for each reward component r_j, $j \in [J]$, and each action $a \in \mathcal{A}$. To facilitate an efficient representation, the order in which the components of s are inspected can vary across different transition probabilities and reward components. The size of an ADD's description scales with the number of functions employed in its definition.

Finally, a *rule system* represents a function $f : \mathcal{S} \rightarrow \mathbb{R}$ via a number of mutually exclusive and jointly exhaustive conjunctions that involve (negations of) some of the components of the state s, together with the associated function values $f(s)$ if the respective conjunctions are satisfied. We can exploit context-specific independence in the transition probabilities $p_n(s'_n \mid s, a)$ by constructing a separate rule system for each sub-state $n \in [N]$, each possible value $s'_n \in \mathcal{S}_n$ of that sub-state, and each action $a \in \mathcal{A}$. Likewise, we can exploit context-specific independence in the rewards $r(s, a)$ by constructing a separate rule system for each reward component r_j, $j \in [J]$, and each action $a \in \mathcal{A}$. The size of a rule system's description scales with the number of conjunctions employed in its definition.

Figure 8 provides representations of the transition probabilities $p_n(s'_{n,i} = 1 \mid s, a)$ from Example 5 as trees, ADDs and rule systems.



(a) Tree representation

$$
\begin{aligned}
f^1(s_{n-1,1}, \ldots, s_{n,i+1}) &= s_{n-1,1} \cdot g^1(s_{n,i}, s_{n,i+1}) + (1 - s_{n-1,1}) \cdot f^2(s_{n-1,2}, \ldots, s_{n,i+1}) \\
f^2(s_{n-1,2}, \ldots, s_{n,i+1}) &= s_{n-1,2} \cdot g^1(s_{n,i}, s_{n,i+1}) + (1 - s_{n-1,2}) \cdot f^3(s_{n-1,3}, \ldots, s_{n,i+1}) \\
f^3(s_{n-1,3}, \ldots, s_{n,i+1}) &= s_{n-1,3} \cdot g^1(s_{n,i}, s_{n,i+1}) + (1 - s_{n-1,3}) \cdot h^1(s_{n,i}, s_{n,i+1}) \\
g^1(s_{n,i}, s_{n,i+1}) &= s_{n,i} \cdot p_1 + (1 - s_{n,i}) \cdot g^2(s_{n,i+1}) \\
g^2(s_{n,i+1}) &= s_{n,i+1} \cdot p_2 + (1 - s_{n,i+1}) \cdot 0 \\
h^1(s_{n,i}, s_{n,i+1}) &= s_{n,i} \cdot q_1 + (1 - s_{n,i}) \cdot h^2(s_{n,i+1}) \\
h^2(s_{n,i+1}) &= s_{n,i+1} \cdot q_2 + (1 - s_{n,i+1}) \cdot 0
\end{aligned}
$$

(b) ADD representation

$$
\begin{array}{l}
s_{n-1,1} = 1 \ \wedge \ s_{n,i} = 1 \ \Rightarrow \ p_1; \quad s_{n-1,1} = 1 \ \wedge \ s_{n,i+1} = 1 \ \Rightarrow \ p_2 \\
s_{n-1,2} = 1 \ \wedge \ s_{n,i} = 1 \ \Rightarrow \ p_1; \quad s_{n-1,2} = 1 \ \wedge \ s_{n,i+1} = 1 \ \Rightarrow \ p_2 \\
s_{n-1,3} = 1 \ \wedge \ s_{n,i} = 1 \ \Rightarrow \ p_1; \quad s_{n-1,3} = 1 \ \wedge \ s_{n,i+1} = 1 \ \Rightarrow \ p_2 \\
s_{n-1,1} = 0 \ \wedge \ s_{n-1,2} = 0 \ \wedge \ s_{n-1,3} = 0 \ \wedge \ s_{n,i} = 1 \ \Rightarrow \ q_1 \\
s_{n-1,1} = 0 \ \wedge \ s_{n-1,2} = 0 \ \wedge \ s_{n-1,3} = 0 \ \wedge \ s_{n,i+1} = 1 \ \Rightarrow \ q_2 \\
s_{n,i} = 0 \ \wedge \ s_{n,i+1} = 0 \ \Rightarrow \ 0
\end{array}
$$

(c) Rule-based representation

Figure 8. Different representations of the transition probabilities $p_n(s_{n,i} = 1 \,|\, s, a)$, $n > 1$, $i \in [9]$ and $a_n = 0$, for the FMDP from Example 5. In part (b), we assume that $p_n(s_{n,i} = 1 \,|\, s, a) = f^1(s)$.

In contrast to trees, ADDs and rule systems, MILP-based feature representations can exploit parsimony in both the state and the action space.

Example 7 (Predictive Maintenance, Cont'd)**.** *Consider a variant of our maintenance problem that comprises N machines in total, out of which m should be repaired in each period. In this case,* *the action space satisfies* $|\mathcal{A}| = \binom{N}{m}$*. Since the representations of trees, ADDs and rule systems grow linearly in the number of actions, they grow exponentially in a natural problem description: for* $N = 100$ *machines and* $m = 20$ *repairs per period, for example, we have approximately* $5.4 \cdot 10^{20}$ *actions. In contrast, for each transition probability* $p_n(s'_{n,i} = 1 \,|\, s, a)$ *the MILP-based feature representation of Example 5 can readily accommodate this setting through the inclusion of a single additional feature* $\phi_4(s, a) = \mathbf{1}[a_n = 1]$ *for each machine* $n \in [N]$*.*

Recall that our MILP-based features can be naturally represented in size $F_c \cdot (F + F_l + F_b)$, the number of feature constraints times the number of features and auxiliary (continuous and binary) variables. We next show these MILP-based feature representations are at least as efficient and sometimes significantly more efficient than trees, ADDs and rule systems in characterizing context-specific independence. To see this, consider a variant of our maintenance problem where the transition probabilities of each machine $n > 1$ depend on whether or not at least 50% of its predecessor machines are in states 8, 9 or 10. In this case, any representation of the transition probabilities via trees or rule systems would necessarily grow exponentially in the problem description since every subset of at least half of each machine's predecessors needs to be covered either by a path from the root node to a leaf (in the case of trees) or a probability rule (in the case of rule systems). In contrast, ADDs as well as MILP-based feature representations can handle this variant efficiently, the latter through the inclusion of $n - 1$ features $\phi_n(s, a) = \mathbf{1}[\sum_{i \in [n]} (s_{i,8} + s_{i,9} + s_{i,10}) \geqslant n/2]$, $n \in [N - 1]$. More formally, we can establish that *any* of the three representations from the literature is dominated by our MILP-based feature representation in the following sense.

Theorem 5 (Complexity of Representing Context-Specific Independence)**.**

(i) *For any description of a function* $f : \mathbb{B}^I \to \mathbb{R}$ *as a tree, ADD or rule system, there is an equivalent description as an MILP-based feature representation whose size is polynomially bounded in the original description.*

(ii) *There are descriptions of functions* $f : \mathbb{B}^I \to \mathbb{R}$ *as MILP-based feature representations for which any equivalent description as a tree, ADD or rule system requires a size that is exponential in the original description.*

E-Companion B: Proofs

Proof of Theorem 1. In view of the first assertion, fix any feature map $\phi : \mathcal{S} \times \mathcal{A} \to \mathbb{B}^F$ and define $\Phi_l = \{(s, a, \varphi_l) \in \mathcal{S} \times \mathcal{A} \times \mathbb{B} \, : \, \phi_l(s, a) = \varphi_l\}$ as the graph of the l-th component of ϕ. We then note that for all $(s, a) \in \mathcal{S} \times \mathcal{A}$ and $l \in [F]$, we have that

$$\phi_l(s,a) = \varphi_l \quad \Longleftrightarrow \quad (s, a, \varphi_l) \in \Phi_l \quad \Longleftrightarrow \quad (s, a, \varphi_l) \neq (s', a', \varphi_l') \;\; \forall (s', a', \varphi_l') \in [\mathcal{S} \times \mathcal{A} \times \mathbb{B}] \backslash \Phi_l.$$

The latter condition holds precisely when

$$\begin{aligned}
& \left[s \neq s' \;\vee\; a \neq a' \;\vee\; \varphi_l \neq \varphi_l' \right] && \forall (s', a', \varphi_l') \in [\mathcal{S} \times \mathcal{A} \times \mathbb{B}] \backslash \Phi_l \\
\Longleftrightarrow \quad & \left[\bigvee_{n \in [N]} \bigvee_{i \in [S_n]} s_{ni} \neq s_{ni}' \;\vee\; \bigvee_{m \in [A]} a_m \neq a_m' \;\vee\; \varphi_l \neq \varphi_l' \right] && \forall (s', a', \varphi_l') \in [\mathcal{S} \times \mathcal{A} \times \mathbb{B}] \backslash \Phi_l \\
\Longleftrightarrow \quad & \left[\sum_{n \in [N]} \sum_{i \in [S_n]} \frac{s_{ni}' - s_{ni}}{2 s_{ni}' - 1} \geqslant 1 \;\vee\; \sum_{m \in [A]} \frac{a_m' - a_m}{2 a_m' - 1} \geqslant 1 \;\vee\; \frac{\varphi_l' - \varphi_l}{2 \varphi_l' - 1} \geqslant 1 \right] && \forall (s', a', \varphi_l') \in [\mathcal{S} \times \mathcal{A} \times \mathbb{B}] \backslash \Phi_l \\
\Longleftrightarrow \quad & \sum_{n \in [N]} \sum_{i \in [S_n]} \frac{s_{ni}' - s_{ni}}{2 s_{ni}' - 1} + \sum_{m \in [A]} \frac{a_m' - a_m}{2 a_m' - 1} + \frac{\varphi_l' - \varphi_l}{2 \varphi_l' - 1} \geqslant 1 && \forall (s', a', \varphi_l') \in [\mathcal{S} \times \mathcal{A} \times \mathbb{B}] \backslash \Phi_l.
\end{aligned}$$

In summary, for each $l \in [F]$ the constraint $\phi_l(s, a) = \varphi_l$ can be represented by a system of at most $2 \cdot |\mathcal{S} \times \mathcal{A}|$ linear constraints, together with the bounds $0 \leqslant \varphi_l \leqslant 1$, and with all coefficients in $\{-1, 0, 1\}$. The encoding size of this representation is therefore polynomial in F, $|\mathcal{S} \times \mathcal{A}|$, and $\sum_{n \in [N]} S_n + A$, which proves the first assertion.

As for the second assertion, we reduce from 3SAT, which is known to be NP-hard (M. R. Garey and D. S. Johnson, 1979). Given a 3SAT instance encoded in d bits, we construct an FMDP with $|\mathcal{A}| = 1$ and $\mathcal{S} = \{0, 1\}^d$, so that each state encodes a candidate instance and $\log |\mathcal{S}| = d$. The remaining components of the FMDP (q, p, r, γ) are not relevant for our construction and are thus omitted. We choose the feature ϕ to evaluate the satisfiability of the 3SAT instance encoded by each state, that is, $\phi(s, a)$ evaluates to 1 if and only if the instance encoded by s is satisfiable.

Suppose that ϕ admits a linear representation without auxiliary variables and with total encoding size L. Then ϕ can be evaluated by solving a linear program with input size L, which takes time poly(L). If L were polynomial in $d = \log |\mathcal{S}|$, this would yield a polynomial-time algorithm for 3SAT, contradicting P $\neq$ NP. Thus, the encoding size L must be super-polynomial in $\log |\mathcal{S}|$. $\square$

Proof of Proposition 1. In view of the first assertion, note that

$$\phi(s,a) \;=\; \mathbf{1}[s_n \in \mathcal{S}'_n] \;=\; \sum_{s'_n \in \mathcal{S}'_n} \mathbf{1}[s_n = s'_n],$$

and that each indicator function $\mathbf{1}[s_n = s'_n]$ can be expressed by a variable $\zeta_{s'_n} \in \mathbb{R}_+$ satisfying

$$1 + \sum_{j \in [S_n]} \frac{s_{n,j} - s'_{n,j}}{2s'_{n,j} - 1} \;\leqslant\; \zeta_{s'_n} \;\leqslant\; \min_{j \in [S_n]} \left\{ \frac{s'_{n,j} + s_{n,j} - 1}{2s'_{n,j} - 1} \right\} \qquad \forall s'_n \in \mathcal{S}'_n.$$

This representation has $F_l = |\mathcal{S}'_n|$ auxiliary variables and $F_c = 1 + |\mathcal{S}'_n| + S_n \cdot |\mathcal{S}'_n|$ constraints.

As for the second assertion, note that

$$\phi(s,a) \;=\; \mathbf{1}[s_n \in \mathcal{S}'_n \;\; \forall n \in \mathcal{N}] \quad \Longleftrightarrow \quad \left(\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] \right) - (|\mathcal{N}| - 1) \;\leqslant\; \phi(s,a) \;\leqslant\; \min_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n],$$

and that each indicator function $\mathbf{1}[s_n \in \mathcal{S}'_n]$ can be represented using $|\mathcal{S}'_n|$ auxiliary variables and $(1 + S_n) \cdot |\mathcal{S}'_n|$ constraints (*cf.* Assertion 1). The overall representation thus has $F_l = \sum_{n \in \mathcal{N}} |\mathcal{S}'_n|$ auxiliary variables and $F_c = \sum_{n \in \mathcal{N}} (1 + S_n) \cdot |\mathcal{S}'_n| + 1 + |\mathcal{N}|$ constraints as claimed.

In view of the third assertion, note that

$$\phi(s,a) \;=\; \mathbf{1}[\exists n \in \mathcal{N} \,:\, s_n \in \mathcal{S}'_n] \;=\; 1 - \mathbf{1}[s_n \in \mathcal{S}_n \backslash \mathcal{S}'_n \;\; \forall n \in \mathcal{N}],$$

and we can thus reuse the representation from Assertion 2.

As for the fourth assertion, note that

$$\phi(s,a) \;=\; \mathbf{1}[s_n \in \mathcal{S}'_n \text{ for at least } \nu \text{ different } n \in \mathcal{N}] \quad \Longleftrightarrow \quad \phi(s,a) \;=\; \mathbf{1}\left[\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] \geqslant \nu \right],$$

and that the latter equation holds if and only if $\phi(s,a) = \eta$ for the auxiliary variable $\eta \in \mathbb{B}$ satisfying

$$\frac{\left(\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] \right) - (\nu - 1)}{|\mathcal{N}| + 1} \;\leqslant\; \eta \;\leqslant\; 1 + \frac{\left(\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] \right) - \nu}{|\mathcal{N}|}.$$

Since each indicator function $\mathbf{1}[s_n \in \mathcal{S}'_n]$ can be represented using $|\mathcal{S}'_n|$ auxiliary variables and

$(1 + S_n) \cdot |\mathcal{S}'_n|$ constraints (*cf.* Assertion 1), the overall representation has $F_l = \sum_{n \in \mathcal{N}} |\mathcal{S}'_n|$ and $F_b = 1$ auxiliary variables as well as $F_c = \sum_{n \in \mathcal{N}} (1 + S_n) \cdot |\mathcal{S}'_n| + 2$ constraints.

In view of the last assertion, finally, note that

$$\phi(s,a) \;=\; \mathbf{1}[s_n \in \mathcal{S}'_n \text{ for at most } \nu \text{ different } n \in \mathcal{N}] \quad \Longleftrightarrow \quad \phi(s,a) \;=\; \mathbf{1}\left[\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] \leqslant \nu\right],$$

and that the latter equation holds if and only if $\phi(s,a) = \eta$ for the auxiliary variable $\eta \in \mathbb{B}$ satisfying

$$\frac{\nu - \left(\sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n] - 1\right)}{|\mathcal{N}| + 1} \;\leqslant\; \eta \;\leqslant\; 1 + \frac{\nu - \sum_{n \in \mathcal{N}} \mathbf{1}[s_n \in \mathcal{S}'_n]}{|\mathcal{N}|}.$$

Since each indicator function $\mathbf{1}[s_n \in \mathcal{S}'_n]$ can be represented using $|\mathcal{S}'_n|$ auxiliary variables and $(1 + S_n) \cdot |\mathcal{S}'_n|$ constraints (*cf.* Assertion 1), the overall representation has $F_l = \sum_{n \in \mathcal{N}} |\mathcal{S}'_n|$ and $F_b = 1$ auxiliary variables as well as $F_c = \sum_{n \in \mathcal{N}} (1 + S_n) \cdot |\mathcal{S}'_n| + 2$ constraints. ☐

Similar to Theorem 1 of M. L. Littman, 1997, our proof of Theorem 2 relies on a reduction to a combinatorial game proposed by L. J. Stockmeyer and A. K. Chandra, 1979. In contrast to our setting, however, M. L. Littman, 1997 studies propositional planning problems that are structurally different from the FMDPs considered in this paper, and as a result our reduction, while relying on the same combinatorial game, requires a different proof strategy.

Proof of Theorem 2. Since the feature map ϕ is assumed to be the identity, we will use Definition 1 (relating to FMDPs) rather than Definition 2 (relating to feature-based FMDPs) throughout this proof. To see that computing the optimal expected total discounted reward of an FMDP is EXPTIME-hard, we consider the following game proposed by L. J. Stockmeyer and A. K. Chandra, 1979:

Two-Person Combinatorial Game G_4.

Instance. Given a 13-DNF formula[2]$F : \mathbb{B}^k \times \mathbb{B}^k \to \mathbb{B}$ and a starting position $(x^0, y^0) \in \mathbb{B}^k \times \mathbb{B}^k$. Player 1 takes the first turn.

Game. The players take turns in switching at most one of their variables $\{x_i\}_{i=1}^k$ (player 1) and $\{y_i\}_{i=1}^k$ (player 2). The player whose switch causes $F(x, y)$ to evaluate to 1 wins.

Question. Does the starting position admit a winning strategy for player 1, that is, a strategy under which player 1 always wins, no matter what player 2 does?

Game G_4 is known to be EXPTIME-complete (L. J. Stockmeyer and A. K. Chandra, 1979, Theorem 3.1).

Without loss of generality, we can assume that each player switches exactly one variable in each turn. Indeed, the possibility to not switch any variable in a particular move can be catered for by adding an auxiliary variable that does not impact the value of F. Moreover, we will use the fact that if there is a winning strategy for player 1, then there is a winning strategy under which the player always wins in at most $2^{\mathcal{O}(k)}$ steps (L. J. Stockmeyer and A. K. Chandra, 1979, p. 160).

For a given instance of G_4, we first construct a simplified FMDP with scope $\mathcal{O}(k)$. We then outline how the FMDP can be modified so that is has scope $\mathcal{O}(\log k)$.

State Space. The state space is $\mathcal{S} = \mathbb{B}^2 \times \mathbb{B}^k \times \mathbb{B}^k \times \mathbb{B}^{\lceil \log_2 T \rceil}$. A state $(w, x, y, z) \in \mathcal{S}$ records the status w of the game ($(0,0)$ = open, $(1,0)$ = player 1 won, $(0,1)$ = player 2 won), the current position (x, y) as well as the number z of completed turns (in binary representation).

Action Space. We set $\mathcal{A} = \{\mathrm{e}_i \, : \, i \in [k]\} \subseteq \mathbb{B}^k$, where $a \in \mathcal{A}$ records for each variable $i \in [k]$ whether x_i is being flipped (if $a_i = 1$) or not (if $a_i = 0$) by player 1.

Initial Distribution. The initial state is (w^0, x^0, y^0, z^0) with $w^0 = (0,0)$ and $z^0 = (0, \ldots, 0)$.

Transition Probabilities. If the game status is $(1,0)$ or $(0,1)$, then the FMDP remains at the current state with probability 1, independent of the selected action. If the game status is $(0,0)$, on the other hand, then the position x of player 1 transitions deterministically to $x'_i = x_i + (1 - 2x_i) \cdot a_i$, $i \in [k]$, where $a \in \mathcal{A}$ is the current action. The position y of player 2 transitions to each of the states $y' = y + (1 - 2y_i) \cdot \mathrm{e}_i$, $i \in [k]$, with probability $1/k$ each. The turn counter z is increased by one (in binary representation). The game status w, finally, is set to $w' = (1,0)$ if $F(x', y) = 1$, it is set to $w' = (0,1)$ if $F(x, y) = 1$ or $z = (1, \ldots, 1)$, and it remains $w' = w$ otherwise. Note that the

---

[2] 13-DNF: $C_1 \vee \ldots C_m$, $m \in \mathbb{N}$, where each C_i is a conjunction of at most 13 variables (or their negations).

game status is backward looking with respect to the move of player 2, that is, a winning move of player 2 in the z-th turn is only recognized by the new game status w' at the end of turn $z+1$.

Reward Function. We set $r((w,x,y,z),a) = -1$ if $w = (0,1)$ and $r((w,x,y,z),a) = 0$ otherwise.

Discount Factor. We choose any discount factor $\gamma \in (0,1)$.

We claim that the optimal expected total discounted reward of the above FMDP vanishes if and only if player 1 has a winning strategy in game G_4. Indeed, one readily verifies that the FMDP tracks the game's evolution in a canonical way. Moreover, if player 1 has a winning strategy, then player 2 cannot win the game, which implies that no non-zero rewards are earned under the winning strategy, thus resulting in zero expected total discounted reward. If there is a policy earning zero expected total discounted reward, finally, then it corresponds to a strategy under which player 1 wins in no more than T turns, irrespective of the moves of player 2 – this is, by definition, a winning strategy.

The FMDP described above can be implemented with a scope of $\mathcal{O}(k)$, that is, a scope that does not depend on the duration T of the game. Indeed, through a judicious choice of features, the transition probabilities for each position x'_i of player 1, $i \in [k]$, can be implemented with a scope of 4 as they depend on the game status w, the previous position x_i as well as the action a_i selected for position x_i. The transition probabilities for each position y'_i of player 2, $i \in [k]$, on the other hand, require a scope of $k+2$ as they depend on the game status w as well as all previous positions $\{y_j\}_{j=1}^k$; this is needed to ensure that exactly one position is switched. The transition probabilities for each bit z'_i of the turn counter, $i \in [\lceil \log_2 T \rceil]$, as well as the status w' of the game, finally, can be implemented with a scope of 2 since they evolve deterministically.

The above FMDP can be modified so that it has a scope of $\mathcal{O}(\log k)$. To this end, we augment the state space to $\mathcal{S} = \mathbb{B}^2 \times \mathbb{B}^k \times \mathbb{B}^k \times \mathbb{B}^{\lceil \log_2 k \rceil} \times \mathbb{B}^{\lceil \log_2 T \rceil}$. A state (w,x,y,u,z) now additionally records in the game status w whose turn it is ($(0,0)$ = player 1's turn, $(0,1)$ = player 2's turn, $(1,0)$ = player 1 has won, $(1,1)$ = player 2 has won) as well as, if player 2 moves next, which of her variables y_j, $j = \sum_{l=1}^{\lceil \log_2 k \rceil} 2^{l-1} \cdot u_l$, is switched. In that case, the scope of the transition probabilities for each position y'_j of player 2, $j \in [k]$, can be reduced to $\log(k)$ since they become deterministic functions of the components of u, while the transition probabilities of u have scope 0 since they do not depend on any sub-state or action. For the sake of brevity, we omit the details of this tedious but otherwise straightforward extension. $\square$

Proof of Proposition 2. Replacing the value function v in (2) with our basis function approximation results in the following LP:

$$
\begin{array}{ll}
\underset{w}{\text{minimize}} & \displaystyle\sum_{s\in\mathcal{S}} \left[\prod_{n\in[N]} q_n(s_n)\right] \left[\sum_{k\in[K]} w_k \cdot \nu_k(s)\right] \\
\text{subject to} & \displaystyle\sum_{k\in[K]} w_k \cdot \nu_k(s) \geqslant \sum_{j\in[J]} r_j(\phi(s,a)) + \gamma \sum_{s'\in\mathcal{S}} \left[\prod_{n\in[N]} p_n(s'_n \mid \phi(s,a))\right] \left[\sum_{k\in[K]} w_k \cdot \nu_k(s')\right] \\
& \hfill \forall (s,a) \in \mathcal{S}\times\mathcal{A} \\
& w \in \mathbb{R}^K.
\end{array}
$$

In view of the objective function of this problem, we note that

$$
\begin{aligned}
&\sum_{s\in\mathcal{S}} \left[\prod_{n\in[N]} q_n(s_n)\right] \left[\sum_{k\in[K]} w_k \cdot \nu_k(s)\right] \;=\; \sum_{k\in[K]} w_k \sum_{s\in\mathcal{S}} \nu_k(s) \cdot \left[\prod_{n\in[N]} q_n(s_n)\right] \\
&= \sum_{k\in[K]} w_k \sum_{\substack{s_n\in\mathcal{S}_n:\\ n\in\bar{\mathfrak{s}}[\nu_k]}} \; \sum_{\substack{s_{n'}\in\mathcal{S}_{n'}:\\ n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]}} \nu_k(s) \cdot \left[\prod_{n\in\bar{\mathfrak{s}}[\nu_k]} q_n(s_n)\right] \cdot \left[\prod_{n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]} q_{n'}(s_{n'})\right] \\
&= \sum_{k\in[K]} w_k \sum_{s\in\overline{\mathfrak{S}}[\nu_k]} \; \sum_{\substack{s'_{n'}\in\mathcal{S}_{n'}:\\ n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]}} \nu_k(s) \cdot \left[\prod_{n\in\bar{\mathfrak{s}}[\nu_k]} q_n(s_n)\right] \cdot \left[\prod_{n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]} q_{n'}(s'_{n'})\right], \qquad (8)
\end{aligned}
$$

where the first identity applies standard algebraic manipulations, the second identity decomposes each state $s \in \mathcal{S}$ into its components s_n, $n \in \bar{\mathfrak{s}}[\nu_k]$, and $s_{n'}$, $n' \in [N]\setminus\bar{\mathfrak{s}}[\nu_k]$, and the third identity utilizes our definition of $\overline{\mathfrak{S}}[\nu_k]$ and the fact that ν_k does not depend on the components $s_{n'}$, $n' \in [N]\setminus\bar{\mathfrak{s}}[\nu_k]$. We next observe that (8) can be re-expressed as

$$
\begin{aligned}
&\sum_{k\in[K]} w_k \left(\sum_{s\in\overline{\mathfrak{S}}[\nu_k]} \nu_k(s) \cdot \left[\prod_{n\in\bar{\mathfrak{s}}[\nu_k]} q_n(s_n)\right]\right) \cdot \left(\sum_{\substack{s'_{n'}\in\mathcal{S}_{n'}:\\ n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]}} \; \prod_{n'\in[N]\setminus\bar{\mathfrak{s}}[\nu_k]} q_{n'}(s'_{n'})\right) \\
&= \sum_{k\in[K]} w_k \left(\sum_{s\in\overline{\mathfrak{S}}[\nu_k]} \nu_k(s) \cdot \left[\prod_{n\in\bar{\mathfrak{s}}[\nu_k]} q_n(s_n)\right]\right),
\end{aligned}
$$

where the first row uses the distributive property of multiplication over addition, and the identity holds since the second multiplier in the first row evaluates to one. Similar arguments show that

$$\sum_{s' \in \mathcal{S}} \left[\prod_{n \in [N]} p_n(s'_n \,|\, \phi(s,a)) \right] \left[\sum_{k \in [K]} w_k \cdot \nu_k(s') \right] \;=\; \sum_{k \in [K]} w_k \left(\sum_{s' \in \overline{\mathfrak{S}}[\nu_k]} \nu_k(s') \left[\prod_{n \in \overline{\mathfrak{s}}[\nu_k]} p_n(s'_n \,|\, \phi(s,a)) \right] \right),$$

and we thus conclude that under our value function approximation, (2) indeed simplifies to (3).

To prove the second part, we show that Problem (3) is feasible and not unbounded. The statement then follows from the fact that every feasible finite-dimensional LP attains its optimal value if the latter is finite. To see that (3) is feasible, fix $C = (1-\gamma)^{-1} \cdot \max\{r(\phi(s,a)) \,:\, (s,a) \in \mathcal{S} \times \mathcal{A}\}$ and let $w^{\mathbf{1}} \in \mathbb{R}^K$ be such that $\sum_{k \in [K]} w^{\mathbf{1}}_k \cdot \nu_k(s) = 1$ for all $s \in \mathcal{S}$. We then observe that

$$\begin{aligned}
\sum_{k \in [K]} (C \cdot w^{\mathbf{1}}_k) \cdot \nu_k(s) \;&=\; C \;=\; \frac{1}{1-\gamma} \cdot \max\left\{ r(\phi(s',a)) \,:\, (s',a) \in \mathcal{S} \times \mathcal{A} \right\} \\
&=\; \max\left\{ r(\phi(s',a)) \,:\, (s',a) \in \mathcal{S} \times \mathcal{A} \right\} + \gamma \cdot C \\
&\geqslant\; \max_{a \in \mathcal{A}} \left[r(\phi(s,a)) + \gamma \sum_{k \in [K]} (C \cdot w^{\mathbf{1}}_k) \sum_{s' \in \overline{\mathfrak{S}}[\nu_k]} \nu_k(s') \prod_{n \in \overline{\mathfrak{s}}[\nu_k]} p_n(s'_n \,|\, \phi(s,a)) \right]
\end{aligned}$$

for all $s \in \mathcal{S}$, where the first two identities are due to the definitions of $w^{\mathbf{1}}$ and C, respectively, the third identity applies basic algebraic manipulations, and the inequality follows from the definition of $w^{\mathbf{1}}$, the fact that $\max\{r(\phi(s,a)) \,:\, a \in \mathcal{A}\} \leqslant \max\{r(\phi(s',a)) \,:\, (s',a) \in \mathcal{S} \times \mathcal{A}\}$, as well as the inequality $\sum_{s' \in \overline{\mathfrak{S}}[\nu_k]} \prod_{n \in \overline{\mathfrak{s}}[\nu_k]} p_n(s'_n \,|\, \phi(s,a)) \leqslant 1$, which holds for all $(s,a) \in \mathcal{S} \times \mathcal{A}$.

To see that Problem (3) is not unbounded, finally, note that Problem (2) bounds (3) from below. As we have discussed in Section 4, however, the optimal value of (2) coincides with the expected total discounted reward of an optimal policy to the FMDP, which is bounded by construction. □

Proof of Theorem 3. To see that Algorithm 1 terminates in finite time, we note that by construction of the algorithm, every constraint of Problem (3) is added at most once. The claim now follows from the fact that Problem (3) comprises finitely many constraints.

To see that $\hat{w}$ is feasible in Problem (3), fix any $(s, a) \in \mathcal{S} \times \mathcal{A}$ and observe that

$$\begin{aligned} \sum_{k \in [K]} \hat{w}_k \cdot \nu_k(s) \quad &= \quad \left[\sum_{k \in [K]} w_k^\epsilon \cdot \nu_k(s) \right] + \frac{\epsilon}{1-\gamma} \\ &\geqslant \; r(\phi(s,a)) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid \phi(s,a)) \left[\sum_{k \in [K]} w_k^\epsilon \cdot \nu_k(s') \right] - \epsilon + \frac{\epsilon}{1-\gamma} \\ &= \; r(\phi(s,a)) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid \phi(s,a)) \left[\left(\sum_{k \in [K]} w_k^\epsilon \cdot \nu_k(s') \right) + \frac{\epsilon}{1-\gamma} \right], \end{aligned}$$

where the first identity employs the definition of $\hat{w}$, the first inequality exploits the ϵ-feasibility of w^ϵ, and the second identity reorders terms and uses that $\sum_{s' \in \mathcal{S}} p(s' \mid \phi(s,a)) = 1$. Applying the identity from the first row again to the term inside the square bracket in the last row, we obtain

$$\sum_{k \in [K]} \hat{w}_k \cdot \nu_k(s) \quad \geqslant \quad r(\phi(s,a)) + \gamma \sum_{s' \in \mathcal{S}} p(s' \mid \phi(s,a)) \left[\sum_{k \in [K]} \hat{w}_k \cdot \nu_k(s) \right] \qquad \forall (s,a) \in \mathcal{S} \times \mathcal{A},$$

that is, $\hat{w}$ is indeed feasible in Problem (3). The feasibility of $\hat{w}$ in (3) and the fact that w^ϵ optimally solves a relaxation of (3) then implies that $F(w^\epsilon) \leqslant F^\star \leqslant F(\hat{w})$.

The upper bound for $F(\hat{w})$ can be proved as follows:

$$F(\hat{w}) - F^\star \quad \leqslant \quad F(\hat{w}) - F(w^\epsilon) \quad = \quad \sum_{s \in \mathcal{S}} q(s) \cdot \left[\sum_{k \in [K]} (\hat{w}_k - w_k^\epsilon) \cdot \nu_k(s) \right] \quad = \quad \frac{\epsilon}{1-\gamma},$$

where the first inequality holds since w^ϵ optimally solves the master problem of the last iteration of Algorithm 1, which itself is a relaxation of Problem (3), the first identity employs formulation (2) and Proposition 2 to evaluate $F(\hat{w})$ and $F(w^\epsilon)$, and the last identity follows from the definitions of $\hat{w}$ and $w^{\mathbf{1}}$ as well as the fact that $\sum_{s \in \mathcal{S}} q(s) = 1$.

To see that $F(w^\epsilon) \geqslant F^\star - \epsilon(1-\gamma)^{-1}$, we recall that $\hat{w} = w^\epsilon + \frac{\epsilon}{1-\gamma} \cdot w^{\mathbf{1}}$. Using this and the linearity of the objective function F, we get

$$F(\hat{w}) = F(w^\epsilon) + \frac{\epsilon}{1-\gamma}.$$

The feasibility of $\hat{w}$ then yields the desired lower bound:

$$F(w^{\epsilon}) = F(\hat{w}) - \frac{\epsilon}{1-\gamma} \geqslant F^{\star} - \frac{\epsilon}{1-\gamma}.$$

This completes the proof. ☐

Proof of Theorem 4. As discussed in Algorithm 1, a maximally violated constraint can be extracted from an optimal solution to the optimization problem

$$\begin{array}{ll} \underset{s,\,a,\,\varphi,\,\zeta}{\text{maximize}} & \displaystyle\sum_{j \in [J]} r_j(\varphi) + \gamma \sum_{k \in [K]} w_k^{\star} \cdot \overline{\nu}_k(\varphi) - \sum_{k \in [K]} w_k^{\star} \cdot \nu_k(s) \\ \text{subject to} & (s,a) \in \mathcal{S} \times \mathcal{A}, \;\; (\varphi, \zeta; s, a) \in \mathcal{F}, \;\; \zeta \in \mathbb{R}^{F_l} \times \mathbb{B}^{F_b}. \end{array} \tag{9}$$

The reward components $r_j(\varphi)$, $j \in [J]$, in this problem can be reformulated as

$$r_j(\varphi) \;\; = \sum_{f \in \mathfrak{S}[r_j]} r_j(f) \cdot \mathbf{1}\left[\varphi_i = f_i \;\; \forall i \in \mathfrak{s}[r_j]\right] \;\; = \sum_{f \in \mathfrak{S}[r_j]} r_j(f) \cdot \eta_{jf}$$

for $\eta_{jf} = \mathbf{1}[\varphi_i = f_i \;\; \forall i \in \mathfrak{s}[r_j]]$, $j \in [J]$ and $f \in \mathfrak{S}[r_j]$. One readily verifies that $\eta_{jf} = \mathbf{1}[\varphi_i = f_i \;\; \forall i \in \mathfrak{s}[r_j]]$ if and only if $\eta_{jf} \in [0,1]$ and

$$\eta_{jf} \leqslant (2f_i - 1)\varphi_i + 1 - f_i \;\; \forall i \in \mathfrak{s}[r_j] \quad \text{as well as} \quad \eta_{jf} \geqslant 1 + \sum_{i \in \mathfrak{s}[r_j]} \frac{\varphi_i - f_i}{2f_i - 1}.$$

Similar arguments show that

$$\overline{\nu}_k(\varphi) \;\; = \sum_{f \in \mathfrak{S}[\overline{\nu}_k]} \overline{\nu}_k(f) \cdot \mathbf{1}\left[\varphi_i = f_i \;\; \forall i \in \mathfrak{s}[\overline{\nu}_k]\right] \;\; = \sum_{f \in \mathfrak{S}[\overline{\nu}_k]} \overline{\nu}_k(f) \cdot \xi_{kf}$$

for $\xi_{kf} = \mathbf{1}[\varphi_i = f_i \;\; \forall i \in \mathfrak{s}[\overline{\nu}_k]]$, $k \in [K]$ and $f \in \mathfrak{S}[\overline{\nu}_k]$, which holds if and only if $\xi_{kf} \in [0,1]$ and

$$\xi_{kf} \leqslant (2f_i - 1)\varphi_i + 1 - f_i \;\; \forall i \in \mathfrak{s}[\overline{\nu}_k] \quad \text{as well as} \quad \xi_{kf} \geqslant 1 + \sum_{i \in \mathfrak{s}[\overline{\nu}_k]} \frac{\varphi_i - f_i}{2f_i - 1},$$

as well as

$$\nu_k(s) \quad = \sum_{s' \in \mathfrak{S}[\nu_k]} \nu_k(s') \cdot \mathbf{1}\left[[s]_i = [s']_i \;\; \forall i \in \mathfrak{s}[\nu_k]\right] \quad = \sum_{s' \in \mathfrak{S}[\nu_k]} \nu_k(s') \cdot \beta_{ks'}$$

for $\beta_{ks'} = \mathbf{1}[[s]_i = [s']_i \;\; \forall i \in \mathfrak{s}[\nu_k]]$, $k \in [K]$ and $s' \in \mathfrak{S}[\nu_k]$, which holds if and only if $\beta_{ks'} \in [0,1]$ and

$$\beta_{ks'} \leqslant (2[s']_i - 1)[s]_i + 1 - [s']_i \;\; \forall i \in \mathfrak{s}[\nu_k] \quad \text{as well as} \quad \beta_{ks'} \geqslant 1 + \sum_{i \in \mathfrak{s}[\nu_k]} \frac{[s]_i - [s']_i}{2[s']_i - 1}.$$

The statement now follows if we replace $r_j(\varphi)$, $\overline{\nu}_k(\varphi)$ and $\nu_k(s)$ in (9) with their reformulations. □

Proof of Proposition 3. Theorem 4 determines a maximally violated constraint through the solution of an MILP whose size is polynomial in the description of the feature-based FMDP. We thus conclude that determining a maximally violated constraint is in NP. To see that the problem is also strongly NP-hard, consider the following integer feasibility problem:

0/1 INTEGER PROGRAMMING.

Instance. Given are $G \in \mathbb{Z}^{L \times N}$ and $h \in \mathbb{Z}^L$.

Question. Is there a vector $x \in \{0,1\}^N$ such that $Gx \leqslant h$?

For a given instance of the integer feasibility problem, we construct a feature-based FMDP with the state space $\mathcal{S} = \times_{n \in [N]} \mathcal{S}_n$ with $\mathcal{S}_n = \mathbb{B}$, $n \in [N]$, the action space $\mathcal{A} = \mathbb{B}$, the feature map $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{B}$ satisfying $\phi(s,a) = \mathbf{1}[Gs \leqslant h]$, the reward function satisfying $r(s,a) = \phi(s,a)$, any scope-1 initial distribution q, any scope-1 transition kernel p as well as any discount factor; we not specify those components further since they will not impact our argument. We set the basis function weights w to zero so that the choice of basis functions for the value function is irrelevant as well. In this case, any maximally violated constraint $(s^\star, a^\star)$ simply maximizes $\phi(s,a)$ over $\mathcal{S} \times \mathcal{A}$, and we conclude that the maximum constraint violation is 1 precisely when the answer to the integer feasibility problem is affirmative. Since the integer feasibility problem is strongly NP-complete (M. R. Garey and D. S. Johnson, 1979), the statement of the proposition then follows.

It remains to be shown that the feature map $\phi : \mathcal{S} \times \mathcal{A} \rightarrow \mathbb{B}$ admits a representation that is polynomial in the length of the integer feasibility instance. To this end, we claim that

$$\phi(s,a)=\varphi \quad\Longleftrightarrow\quad \left[\begin{array}{ll} {g_l}^\top s \geqslant h_l + \dfrac{1}{2} - \mathrm{M}\cdot\zeta_l & \forall l\in[L] \\ {g_l}^\top s \leqslant h_l + \mathrm{M}\cdot(1-\zeta_l) & \forall l\in[L] \\ \varphi\leqslant\zeta_l & \forall l\in[L] \\ \varphi\geqslant\left(\displaystyle\sum_{l\in[L]}\zeta_l\right)-(L-1) & \end{array}\right] \text{ for some } \zeta\in\mathbb{B}^L, \tag{10}$$

where ${g_l}^\top \in \mathbb{Z}^N$ is the l-th row of matrix G and $\mathrm{M} = \max_{l\in[L]}\left[1+|h_l|+\sum_{n\in[N]}|g_{ln}|\right]$. To see this, note that thanks to the integrality of G and h, the equation system (10) ensures that $\zeta_l = \mathbf{1}[{g_l}^\top s \leqslant h_l]$ for all $l\in[L]$ as well as $\varphi=\min\{\zeta_l : l\in[L]\}$. Thus, whenever $\phi(s,a)=1$, that is, whenever $Gs\leqslant h$, we have $\zeta=\mathbf{1}$ and thus $\varphi=1$. In contrast, whenever $\phi(s,a)=0$, that is, whenever $Gs\nleqslant h$, we have $\zeta_l=0$ for at least one $l\in[L]$, and thus $\varphi=0$ as desired. $\square$

Proof of Corollary 1. The statement follows immediately from the optimization problem in Theorem 4 if we fix the state s, disregard the last expression from the objective function and remove the auxiliary variables $\{\beta_{ks'}\}$. $\square$

Proof of Proposition 4. Let $F^\star$ denote the minimum cardinality of any feature set that exactly recovers p, and define

$$\delta \;=\; \left[\begin{array}{ll} \underset{\Gamma,\psi}{\text{minimize}} & \displaystyle\sum_{n\in[N]}\sum_{s\in\mathcal{S}}\sum_{a\in\mathcal{A}}\left\|p_n(\cdot\mid s,a)-\psi_n(\phi_\Gamma(s,a))\right\|_2^2 \\ \text{subject to} & \Gamma\subseteq\Phi,\;\; |\Gamma|<F^\star \;\text{ and }\; \psi_n:\mathbb{B}^{|\Gamma|}\to\Delta(\mathcal{S}_n),\; n\in[N] \end{array}\right]$$

as the minimum unregularized objective value of any feature selection Γ with less that $F^\star$ features in Problem (5'). We claim that the statement of the proposition holds for $\lambda_0=\delta/F^\star$. We prove our claim by showing that *(i)* any feature selection with less than $F^\star$ features attains an objective value exceeding $\lambda\cdot F^\star$ in Problem (5'); *(ii)* any feature selection with $F^\star$ features that exactly recovers p achieves an objective value of exactly $\lambda\cdot F^\star$ in (5'); *(iii)* any feature selection with $F^\star$ features that does not exactly recover p, as well as any feature set with more than $F^\star$ features, attains an objective value of exceeding $\lambda\cdot F^\star$ in (5'); and *(iv)* any feature selection Γ attaining an objective value of

$\lambda \cdot F^\star$ in (5') must have $F^\star$ features and recover p exactly. The proof then concludes since $\delta > 0$, for otherwise there would be a feature set with cardinality less than $F^\star$ that exactly recovers p.

In view of claim *(i)*, note that any feature set $\Gamma \subseteq \Phi$ with $|\Gamma| < F^\star$ must have an objective value of at least $\delta = \lambda_0 \cdot F^\star > \lambda \cdot F^\star$ by definition of δ. Claims *(ii)–(iv)* follow immediately from the construction of the objective function in Problem (5'). As for the last claim, note first that the claims *(i)* and *(iii)* imply that $|\Gamma| = F^\star$, for otherwise the objective value would have to exceed $\lambda \cdot F^\star$. In that case, however, the objective value of Γ satisfies

$$\sum_{n\in[N]} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \left\| p_n(\cdot \mid s,a) - \psi_n(\phi_\Gamma(s,a)) \right\|_2^2 + \lambda \cdot F^\star \quad = \quad \lambda \cdot F^\star,$$

which immediately implies that ϕ_Γ recovers p exactly. □

Proof of Observation 1. Note that the objective function in Problem (6) satisfies

$$\sum_{n\in[N]} \sum_{s\in\mathcal{S}} \sum_{a\in\mathcal{A}} \left\| p_n(\cdot \mid s,a) - \psi_n(\phi_{\Gamma\cup\{\varphi\}}(s,a)) \right\|_2^2 \quad = \quad \sum_{n\in[N]} \sum_{\phi\in\mathbb{B}^{|\Gamma|+1}} \sum_{(s,a)\in\Omega(\phi)} \left\| p_n(\cdot \mid s,a) - \psi_n(\phi) \right\|_2^2,$$

and that the right-hand side of this equation can be expressed as

$$\sum_{n\in[N]} \sum_{\phi\in\mathbb{B}^{|\Gamma|+1}} \left(|\Omega(\phi)| \cdot \left\| \frac{1}{\Omega(\phi)} \cdot \sum_{(s,a)\in\Omega(\phi)} p_n(\cdot \mid s,a) - \psi_n(\phi) \right\|_2^2 + c_n(\phi) \right), \tag{11}$$

where the term

$$c_n(\phi) = \sum_{(s,a)\in\Omega(\phi)} \left\| p_n(\cdot \mid s,a) \right\|_2^2 - \frac{1}{|\Omega(\phi)|} \cdot \left\| \sum_{(s,a)\in\Omega(\phi)} p_n(\cdot \mid s,a) \right\|_2^2$$

does not depend on ψ_n. The expression (11) has $\psi_n^\star$ from the statement of the observation as an unconstrained minimizer. Since $\psi_n^\star(\phi) \in \Delta(\mathcal{S}_n)$ for all $n \in [N]$ and $\phi \in \mathbb{B}^{|\Gamma|+1}$, it must also be the constrained minimizer. □

Proof of Theorem 5. In view of statement *(i)*, consider first a representation of the function

$f : \mathbb{B}^I \to \mathbb{R}$ as a rule system. Concretely, fix a representation of the form

$$\bigwedge_{(i,v) \in \mathfrak{A}_r} (x_i = v) \quad \Rightarrow \quad f(x) = c_r \qquad \forall r \in \mathfrak{R},$$

where $\mathfrak{R}$ is the set of rules to determine f, $\mathfrak{A}_r$ is the set of antecedents (or premises) for rule r and c_r is the consequent (or conclusion) of r. This rule-based representation is equivalent to the MILP-based feature representation under which $f(x) = c_r$ if and only if $\phi_r(x) = 1$, $r \in \mathfrak{R}$, where the MILP-based features $\phi_r(x) = \mathbf{1}[\bigwedge_{(i,v) \in \mathfrak{A}_r} (x_i = v)]$ have the polynomial-size representation

$$\left.\begin{array}{l} \phi_r(x) \leqslant 1 - x_i \ \ \forall (i,0) \in \mathfrak{A}_r; \quad \phi_r(x) \leqslant x_i \ \ \forall (i,1) \in \mathfrak{A}_r \\ \phi_r(x) \geqslant \displaystyle\sum_{(i,0) \in \mathfrak{A}_r} (1 - x_i) + \sum_{(i,1) \in \mathfrak{A}_r} x_i - [|\mathfrak{A}_r| - 1] \end{array}\right\} \ \forall r \in \mathfrak{R}.$$

This concludes statement *(i)* for rule systems. Since every decision tree can be represented as a rule system with one rule for each path from the root node to one of the leaf nodes, our proof for rule systems immediately applies to decision tree representations as well.

To see that statement *(i)* also holds for ADDs, we first observe that the empty function $f : \varnothing \to \mathbb{R}$ satisfying $f() = c$ has a trivial MILP-based feature representation. Assume now that the two functions $f', f'' : \mathbb{B}^{I-i}$ are ADDs of binary vectors $(x_{i+1}, \ldots, x_I)$ with polynomial-sized MILP-based feature representations $f'(x_{i+1}, \ldots, x_I) = c'_r$ if $\phi'_r(x_{i+1}, \ldots, x_I) = 1$, $r \in \mathfrak{R}'$, and $f''(x_{i+1}, \ldots, x_I) = c''_r$ if $\phi''_r(x_{i+1}, \ldots, x_I) = 1$, $r \in \mathfrak{R}''$. This implies that there are polyhedra $\mathcal{F}', \mathcal{F}''$ as well as auxiliary vectors $\zeta' \in \mathbb{R}^{F'_l} \times \mathbb{B}^{F'_b}$ and $\zeta'' \in \mathbb{R}^{F''_l} \times \mathbb{B}^{F''_b}$, all of polynomial size, such that $\phi'(x_{i+1}, \ldots, x_I) = \varphi \in \mathbb{B}^{|\mathfrak{R}'|}$ precisely when $(\varphi, \zeta; x_{i+1}, \ldots, x_I) \in \mathcal{F}'$ and $\phi''(x_{i+1}, \ldots, x_I) = \varphi \in \mathbb{B}^{|\mathfrak{R}''|}$ precisely when $(\varphi, \zeta; x_{i+1}, \ldots, x_I) \in \mathcal{F}''$, respectively. We show that in this case, the function $f : \mathbb{B}^{I-i+1}$ satisfying $f(x_i, \ldots, x_I) = x_i \cdot f'(x_{i+1}, \ldots, x_I) + (1 - x_i) \cdot f''(x_{i+1}, \ldots, x_I)$ has an MILP-based feature representation $f(x_i, \ldots, x_I) = c_r$ if $\phi_r(x_i, \ldots, x_I) = 1$, $r \in \mathfrak{R}$, of polynomial size. Indeed, fix $\mathfrak{R} = \{(1, r') : r' \in \mathfrak{R}'\} \cup \{(0, r'') : r'' \in \mathfrak{R}''\}$, $c_r = c'_{r'}$ for $r = (1, r') \in \mathfrak{R}$, $r' \in \mathfrak{R}'$, and $c_r = c''_{r''}$ for $r = (0, r'') \in \mathfrak{R}$, $r'' \in \mathfrak{R}''$. The claim follows if we can show that the feature map ϕ satisfying

$$\begin{aligned} & \phi_r(x_i, \ldots, x_I) = \mathbf{1}[x_i = 1 \text{ and } \phi'_{r'}(x_{i+1}, \ldots, x_I) = 1] \quad \text{for } r = (1, r') \in \mathfrak{R}, r' \in \mathfrak{R}' \\ \text{and} \quad & \phi_r(x_i, \ldots, x_I) = \mathbf{1}[x_i = 0 \text{ and } \phi''_{r''}(x_{i+1}, \ldots, x_I) = 1] \quad \text{for } r = (0, r'') \in \mathfrak{R}, r'' \in \mathfrak{R}'' \end{aligned}$$

has an MILP representation of polynomial size. This is the case since we have

$$\phi(x_i, \ldots, x_I) = \varphi \in \mathbb{B}^{|\mathfrak{R}|} \quad \Longleftrightarrow \quad \left[\begin{array}{ll} \varphi \geqslant \varphi' - (1 - x_i) \cdot \mathrm{e}, & \varphi \leqslant \varphi' + (1 - x_i) \cdot \mathrm{e} \\ \varphi \geqslant \varphi'' - x_i \cdot \mathrm{e}, & \varphi \leqslant \varphi'' + x_i \cdot \mathrm{e} \\ (\varphi', \zeta'; x_{i+1}, \ldots, x_I) \in \mathcal{F}_{r'} & \\ (\varphi'', \zeta''; x_{i+1}, \ldots, x_I) \in \mathcal{F}_{r''} & \end{array} \right].$$

As for statement *(ii)*, consider first the parity function $f(x_1, \ldots, x_I) = 1$ if $\sum_{i=1}^{I} x_i$ is even; $= 0$ otherwise. This function cannot be computed without considering the value x_i of each of its inputs $i \in [I]$, and thus any representation of this function as a tree or rule system necessarily scales exponentially in I. In contrast, we have $f(x_1, \ldots, x_I) = 1$ if $\phi(x_1, \ldots, x_I) = 1$; $= 0$ otherwise for the feature map $\phi(x_1, \ldots, x_I) = \mathbf{1}[\sum_{i=1}^{I} x_i \text{ is even}]$, which has the MILP representation

$$\phi(x_1, \ldots, x_I) = \varphi \quad \Longleftrightarrow \quad \left[\begin{array}{l} \varphi = 1 - z_I, \quad z_1 = x_1 \\ \left. \begin{array}{l} z_{i+1} = z_i + x_{i+1} - 2y_{i+1} \\ y_{i+1} \leqslant z_i, \quad y_{i+1} \leqslant z_{i+1} \\ y_{i+1} \geqslant z_i + z_{i+1} - 1 \end{array} \right\} \ \forall i \in [I-1] \end{array} \right],$$

$\varphi \in \mathbb{B}$, that is of polynomial size. Note that in this formulation, we have $y_{i+1} = \mathbf{1}[z_i = 1 \wedge z_{i+1} = 1]$ for all $i \in [I-1]$ and $z_i = \mathbf{1}[\sum_{i'=1}^{i} x_{i'} \text{ is odd}]$ for all $i \in [I]$.

The argument from the previous paragraph does not extend to ADDs; in fact, ADDs can also represent the parity function efficiently. To see that statement *(ii)* nevertheless extends to ADDs, consider the function $f(a_1, \ldots, a_I; b_1, \ldots, b_I)$ that computes one of the $2I$ bits of the result of multiplying the binary numbers $(a_1, \ldots, a_I)$ and $(b_1, \ldots, b_I)$. It follows from R. E. Bryant, 1991 that ADDs cannot represent this function in polynomial size. In contrast, we show next that this function can be represented efficiently via MILP-based features. To this end, consider the following feature map, which computes the entire result of the multiplication:

$$\phi(a_1, \ldots, a_I; b_1, \ldots, b_I) = \varphi \quad \Longleftrightarrow \quad \left[\begin{array}{l} \displaystyle\sum_{i \in [2I]} 2^{i-1} \cdot \varphi_i \ = \sum_{i,j \in [I]} 2^{i+j-2} \cdot \zeta_{ij} \\ \zeta_{ij} \leqslant a_i, \ \ \zeta_{ij} \leqslant b_j, \ \ \zeta_{ij} \geqslant a_i + b_j - 1 \quad \forall i, j \in [I] \end{array} \right]$$

To see that the above representation is correct, observe first that for all $i, j \in [I]$ we have $\zeta_{ij} = a_i \cdot b_j$.

The right-hand side of the first constraint therefore evaluates to

$$\sum_{i,j\in[I]} 2^{i+j-2} \cdot a_i \cdot b_j \quad = \quad \left(\sum_{i\in[I]} 2^{i-1} a_i \right) \left(\sum_{j\in[I]} 2^{j-1} b_j \right),$$

where $\sum_{i\in[I]} 2^{i-1} \cdot a_i$ and $\sum_{j\in[I]} 2^{j-1} \cdot b_j$ are the numbers represented by the binary vectors $(a_1, \ldots, a_I)$ and $(b_1, \ldots, b_I)$, respectively. The left-hand side of the first constraint thus ensures that the number represented by the binary vector $(\varphi_1, \ldots, \varphi_{2I})$ equals the product of $(a_1, \ldots, a_I)$ and $(b_1, \ldots, b_I)$, as desired. Our constraints defining the feature map ϕ involve coefficients that are exponential in the length $2I$ of its input; this can be avoided by emulating a binary multiplier through MILP constraints. We omit the details of this tedious but otherwise straightforward representation. $\square$

References

L. J. Stockmeyer and A. K. Chandra (1979). “Provably difficult combinatorial games”. In: *SIAM Journal on Computing* 8.2, pp. 151–174.

M. L. Littman (1997). “Probabilistic Propositional Planning: Representations and Complexity”. In: *Proceedings of the 14th AAAI Conference*, pp. 748–754.

M. R. Garey and D. S. Johnson (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman & Co.

R. E. Bryant (1991). “On the Complexity of VLSI Implementations and Graph Representations of Boolean Functions with Application to Integer Multiplication”. In: *IEEE Transactions on Computers* 40.2, pp. 205–213.