CS 405/6001: Game Theory and Algorithmic Mechanism Design **Jul–Nov 2026**

Lecture 2: Proving the Chess Theorem, and Strategic Form Games

Lecturer: Swaprava Nath *Scribe(s): Claude Scribe*

Disclaimer: *These notes aggregate content from several texts and have not been subjected to the usual scrutiny deserved by formal publications. If you find errors, please bring to the notice of the Instructor.*

2.1 Recap

Last time we stated (Theorem 1.4 of the previous lecture) that in chess, either White has a winning strategy, or Black has a winning strategy, or both players have a strategy guaranteeing at least a draw, exactly one of the three. Today we prove it, isolate exactly which properties of chess the proof needs, and then move to the *strategic form* representation of games, which will carry us through the next several weeks.

2.2 Proof I: Induction on the Game Tree

Recall that $\Gamma(x)$ denotes the subtree of the game tree rooted at game situation x, and n_x is the number of vertices in $\Gamma(x)$. If y is a child of x then $\Gamma(y)$ is a proper subtree of $\Gamma(x)$, so $n_y < n_x$; this is what lets us induct on n_x.

Theorem 2.1 *Every subgame $\Gamma(x)$ satisfies exactly one of: (i) White has a winning strategy in $\Gamma(x)$; (ii) Black has a winning strategy in $\Gamma(x)$; (iii) each player has a strategy guaranteeing at least a draw in $\Gamma(x)$.*

Proof: By strong induction on n_x.

Base case, $n_x = 1$. Then x is a terminal situation. If Black's king has already been captured, the empty strategy is trivially a winning strategy for White (there are no more moves to make); symmetrically if White's king is gone. If both kings remain, the game has ended in a draw, and the empty strategy trivially guarantees a draw for both players.

Inductive step, $n_x > 1$. Assume the theorem holds for every subgame with fewer than n_x vertices. Without loss of generality suppose White moves at x; let $C(x)$ be the set of game situations reachable from x in one move of White. Every $y \in C(x)$ has $n_y < n_x$, so the inductive hypothesis applies to $\Gamma(y)$.

Case 1. Some $y_0 \in C(x)$ has White winning in $\Gamma(y_0)$. Then White wins in $\Gamma(x)$: move to y_0, then play the winning strategy there.

Case 2. Black wins in $\Gamma(y)$ for *every* $y \in C(x)$. Then Black wins in $\Gamma(x)$: whatever White's first move, Black identifies the resulting y and plays his winning strategy in $\Gamma(y)$.

Case 3. Neither Case 1 nor Case 2 holds. Since Case 1 fails, no $y \in C(x)$ has White winning; by the inductive hypothesis, every $y \in C(x)$ has either Black winning, or both players guaranteeing a draw. Since Case 2 also fails, some $y' \in C(x)$ does *not* have Black winning, so (by the previous sentence) both players guarantee a draw at y'. White secures a draw in $\Gamma(x)$ by moving to y' and following the drawing strategy there; Black secures a draw by identifying whichever y White actually moved to, and playing his own winning-or-drawing strategy at that y (winning if he can, drawing otherwise – either way he does at least as well as a draw).

That the three cases are mutually exclusive follows from the fact that a player cannot simultaneously have and lack a winning strategy in the same subgame. ■

Setting $x = x_0$ recovers Theorem 2.1 for the whole game of chess. Note the proof is *constructive*: it exhibits, level by level from the leaves up, a strategy realizing whichever of (i)–(iii) holds.

2.3 Which Properties of Chess Did We Actually Use?

The induction never used anything about knights, bishops, or checkmate specifically. Reading the proof again, MSZ isolate exactly three structural properties [[]Ch. 1]MSZ13:

(C1) **Finiteness**: the game tree has finitely many vertices.

(C2) **No chance moves**: no dice, no card draws; the pair of strategies alone determines the entire play.

(C3) **Perfect information**: at every turn, the player to move knows every move made so far by both sides.

Corollary 2.2 *Theorem 2.1 holds, with the identical proof, for* any *finite two-player zero-outcome-type game satisfying (C1)–(C3), not just chess. (The general statement and proof, for n-player games with real-valued payoffs rather than win/lose/draw outcomes, is Kuhn's theorem on games with perfect information; see §2.3.1.)*

Example 2.3 (Where the theorem breaks: Matching Pennies) *Two players simultaneously show a coin as Heads or Tails. Player I wins (gets $+1$, Player II gets -1) if the coins match; otherwise Player II wins. This game is finite (C1) and has no chance moves (C2), but violates (C3): Player II moves "at the same time" as Player I, which in extensive form means Player II's single decision node has* two *game situations in the same information set – he cannot distinguish which move Player I made. One checks directly that neither player has a strategy guaranteeing even a break-even outcome in pure strategies: whatever Player I commits to, Player II can exploit it, and vice versa. Theorem 2.1 genuinely fails here; this is precisely why von Neumann's actual 1928 contribution, the minimax theorem for zero-sum games [vN28], needs a fundamentally different idea (randomizing over actions) that we will introduce as* mixed strategies *in a few weeks.*

2.3.1 A Preview: Kuhn's Theorem

Corollary 2.2 is a special case of a theorem we will meet properly once we cover extensive-form games:

Theorem 2.4 (Kuhn, informal statement) *Every finite extensive-form game with perfect information has a Nash equilibrium, computable by backward induction.*

We flag it now only so that the chess theorem is seen for what it really is: the two-player, win/lose/draw special case of a much more general fact about sequential games.

2.4 Case Study: The Halving Game

Example 2.5 (The Halving Game) *Two players, labeled $+1$ and -1, alternate turns starting with $+1$. The state is a single nonnegative integer, `currNumber`. On a turn, the player to move chooses either* Decrement *(`currNumber` $\to$ `currNumber` -1) or* Halve *(`currNumber` $\to \lfloor$`currNumber`$/2\rfloor$). The player who is* forced to move when `currNumber` $= 0$ *loses.*

This game satisfies (C1)–(C3) – finite (currNumber strictly decreases every turn, so the game ends), no chance moves, and perfect information – so Corollary 2.2 guarantees the analogue of Theorem 2.1 holds: one of the two players has a winning strategy (there is no "draw" outcome here, so case (iii) cannot arise, and in fact exactly one of (i)/(ii) must hold).

The winning player, for a given starting N, can be found by exactly the backward-induction recursion used in Proof I: define $u(\texttt{player}, n)$ recursively, with $u(\texttt{player}, 0) = \texttt{player} \cdot \infty$ (the mover at 0 "wins" with value $+\infty$ if that mover is $+1$, else $-\infty$), and for $n > 0$,

$$u(\texttt{player}, n) = \begin{cases} \max\left(u(-\texttt{player}, n-1),\ u(-\texttt{player}, \lfloor n/2 \rfloor)\right) & \text{if } \texttt{player} = +1, \\ \min\left(u(-\texttt{player}, n-1),\ u(-\texttt{player}, \lfloor n/2 \rfloor)\right) & \text{if } \texttt{player} = -1. \end{cases}$$

This is precisely the `minimaxPolicy` recursion used in the course's accompanying game-playing code: a direct, brute-force implementation of the Case-1/Case-2/Case-3 case analysis of Proof I, specialized to a two-outcome ($\pm\infty$) game. Work out by hand who wins from $N = 6$, then check your answer by running the code.

2.5 From Trees to Tables: Strategic Form

Trees record dynamics: who moves when, and who knows what. Often we only care about *outcomes as a function of the strategies chosen*. Compressing away the dynamics gives the **strategic form**, sometimes called the **normal form**.

Definition 2.6 (MSZ, Def. 4.2) *A game in **strategic form** is a triple $G = (N, (S_i)_{i\in N}, (u_i)_{i\in N})$, where $N = \{1, \ldots, n\}$ is a finite set of players, S_i is player i's strategy set, and $u_i : S \to \mathbb{R}$ (with $S = S_1 \times \cdots \times S_n$) gives player i's payoff for each strategy vector $s = (s_1, \ldots, s_n) \in S$.*

For two players, G is just a payoff matrix: rows are Player 1's strategies, columns Player 2's, and each cell holds a pair of payoffs. Table 1.1 from the previous lecture (the AI Lab Dilemma) is already an example.

2.5.1 Notation for "Everyone Else"

We fix notation used for the rest of the course, following MSZ [[]Ch. 4, §4.4]MSZ13 (Osborne and Rubinstein's textbook [OR94] uses the same convention with different symbols, typically a_{-i}).

Definition 2.7 *For $s = (s_1, \ldots, s_n) \in S$ and player i, write $s_{-i} := (s_1, \ldots, s_{i-1}, s_{i+1}, \ldots, s_n)$, the strategies of all players other than i, and $S_{-i} := \prod_{j\neq i} S_j$ for the set of all such vectors. We then write a full strategy vector as $s = (s_i, s_{-i})$ and player i's payoff as $u_i(s_i, s_{-i})$.*

Every solution concept for the remainder of the course, domination, best response, Nash equilibrium, is a comparison of $u_i(s_i, s_{-i})$ across choices of s_i, holding s_{-i} fixed. The notation isolates exactly the one thing player i actually controls.

2.6 Classic Games in Strategic Form

2.6.1 The Prisoner's Dilemma

Relabel the AI Lab Dilemma's actions Test Thoroughly → Cooperate, Ship Fast → Defect:

		Player II: Cooperate	Player II: Defect
Player I	Cooperate	$4, 4$	$0, 5$
	Defect	$5, 0$	$1, 1$

This is the **Prisoner's Dilemma**, the most-cited game in the field. It originates with an experiment run by Flood and Dresher at RAND in 1950; the "prisoners" framing and name are due to Albert Tucker, who used it to illustrate the result to a psychology audience [[]Ch. 4, Remarks]MSZ13. Both players jointly prefer $(4, 4)$ to $(1, 1)$, yet, as we will prove formally next week using dominance, each player's individually best action is to defect regardless of what the other does. The tension between individually and collectively rational play, first crisply exposed by this example, recurs throughout the rest of the course, most notably when we study externalities and the inefficiency of equilibria (the "price of anarchy").

2.6.2 Rock-Paper-Scissors

		Player II: Rock	Player II: Paper	Player II: Scissors
Player I	Rock	$0, 0$	$-1, 1$	$1, -1$
	Paper	$1, -1$	$0, 0$	$-1, 1$
	Scissors	$-1, 1$	$1, -1$	$0, 0$

A **zero-sum** game: $u_1(s) + u_2(s) = 0$ for every s. Zero-sum games are historically where game theory began (chess and Rock-Paper-Scissors both belong to this class) and remain analytically special: we will see in a few weeks that for such games, the security-level (maxmin) analysis and the equilibrium (Nash) analysis agree, which is not true for general games [[]Ch. 4, §4.12]MSZ13.

2.6.3 Chess, Compressed

Let each row of a matrix be one complete strategy s_W for White (a plan of action for *every* game situation) and each column one complete strategy s_B for Black; each cell records W/B/D. The matrix is finite, since chess is (C1), but its size is far beyond anything that could be written down. Theorem 2.1, restated in this language: some row is all-W, or some column is all-B, or there is a row that is all W-or-D and a column that is all B-or-D.

Remark 2.8 (The map to strategic form loses information) *Two of chess's astronomically many rows can be behaviorally identical, differing only in a branch of the tree that particular strategy never actually* *reaches. MSZ call the version with such duplicate rows/columns merged the* ***reduced strategic form*** *[[]Ch. 4, §4.1]MSZ13. More strikingly, genuinely different extensive-form games – different trees, different information structure – can reduce to the identical strategic-form table; Thompson characterized exactly when two extensive-form games share a strategic form, via three elementary tree transformations [Tho52]. The practical moral: strategic form is the right tool for the questions we ask over the next several weeks (dominance, Nash equilibrium), but it deliberately discards information about* when *moves and information arrive, which is exactly what we will need back once we study subgame perfection and credible threats later in the course.*

2.6.4 Coordination Games and Focal Points

		Group B	
		PyTorch	TensorFlow
Group A	PyTorch	$3, 3$	$0, 0$
	TensorFlow	$0, 0$	$1, 1$

Two collaborating research groups independently choose a deep-learning framework. Both (PyTorch, PyTorch) and (TensorFlow, TensorFlow) are stable, in the sense that neither group benefits from unilaterally switching once matched, but they are not equally good for the groups. How do real players, without communicating, end up coordinating on one equilibrium rather than another, or rather than miscoordinating entirely? Schelling's classic answer is the notion of a **focal point**: a strategy vector made salient by convention, precedent, or a name, rather than by payoffs alone [Sch60]. (Why did the deep learning community actually converge on one dominant framework circa 2019–2020? A mix of precedent, tooling network effects, and community momentum, exactly the kind of focal-point reasoning Schelling describes, not anything visible in the payoff matrix itself.)

2.6.5 The AI Safety Race

		Lab B	
		Invest	Skip
Lab A	Invest	$4, 4$	$1, 3$
	Skip	$3, 1$	$2, 2$

Two AI labs each decide whether to invest in costly safety research or skip it to ship faster. This is structurally the **Security Dilemma** from international relations, where two states each choose whether to arm, fearing that restraint while the other arms leaves them exposed [Jer78]; MSZ present the same structure with the USSR and USA choosing whether to build nuclear weapons [[]Ch. 4, Example 4.22]MSZ13. The AI-racing version above is not merely a stylized analogy: formal game-theoretic models of exactly this race dynamic, in which competing developers trade off safety investment against speed of deployment, appear in the literature on AI governance, e.g. Armstrong, Bostrom and Shulman's model of racing to the precipice [ABS16]. We will be able to say something sharper about which outcome is more "dangerous but tempting" once we cover Nash equilibrium next week.

2.7 Summary and Preview

We proved the chess theorem two ways (tree induction; quantifier duality), isolated the three properties (C1–C3) that made the proof work and saw where it breaks (Matching Pennies), analyzed the Halving Game by the same recursion, and introduced the strategic-form representation together with the notation (s_{-i}) we will use every week from here on. Next week: domination, iterated elimination of dominated strategies, and Nash equilibrium, the tools to actually *solve* the games we have only been able to describe so far.

References

[ABS16] S. Armstrong, N. Bostrom and C. Shulman, "Racing to the precipice: a model of artificial intelligence development," *AI & Society*, 31(2), 2016, pp. 201–206.

[Jer78] R. Jervis, "Cooperation under the security dilemma," *World Politics*, 30(2), 1978, pp. 167–214.

[MSZ13] M. Maschler, E. Solan and S. Zamir, *Game Theory*, Chapters 1 and 4, Cambridge University Press, 2013.

[NRTV07] N. Nisan, T. Roughgarden, E. Tardos and V. V. Vazirani, eds., *Algorithmic Game Theory*, Cambridge University Press, 2007.

[OR94] M. J. Osborne and A. Rubinstein, *A Course in Game Theory*, MIT Press, 1994.

[Sch60] T. C. Schelling, *The Strategy of Conflict*, Harvard University Press, 1960.

[Tho52] F. B. Thompson, "Equivalence of games in extensive form," RAND Corporation Research Memorandum RM-759, 1952.

[vN28] J. von Neumann, "Zur Theorie der Gesellschaftsspiele," *Mathematische Annalen*, 100(1), 1928, pp. 295–320.