(say y_1) and, with this, Bob can color y_2 with a different color of y_1 in the next move, winning the game, as there are $k-1$ colors left to color C. If Alice starts at another vertex of G', Bob can ensure that Alice will color s due to the parity of $V' - (V \cup \{s\})$. After this, he can use his winning strategy in G (being the first to play there) and ensure that at least one vertex in V cannot be colored, winning the game. ■

Unlike the proofs of PSPACE-hardness of GEOGRAPHY and Convex Set Forming games, in which the winning strategies were almost directly related by the obtained reduction, in Theorem 5.6 the winning strategy of Bob in G' must take into account the possibility of Alice starting the coloring in G. By doing this, Alice prevents Bob from using the winning strategy he had for the "Bob first" Coloring Game in G. In this case then, Bob uses a strategy in G' completely unrelated to the original and simply waits until y_1 or y_2 is colored.

This should be a constant concern in complexity proofs in games. In order to obtain a reduction f from a game J_1 to a game J_2 ($J_1 \preceq_P J_2$), a player X with winning strategy in an instance i of J_1 must have a winning strategy in the instance $f(i)$ of J_2 regardless of how his opponent behaves, which may mean having to adopt a new and completely different strategy from the original strategy for J_1 in i.

5.4 EXPTIME-completeness and Universal Games

Stockmeyer and Chandra (1979) proved some of the first EXPTIME-complete games in the paper "*Provenly hard combinatorial games*", such as the ASAT game (Alternating SAT)[10]. In this game, given a logical formula Φ in CNF (Conjunctive Normal Form) over a set of variables $X_A \cup X_B$, with initial values given, Alice and Bob alternately change the value of at most one of the variables. In each turn, Alice can change the value of a variable from X_A, while Bob can change the value of a variable in X_B. Alice wins if she can make the formula true at some point; otherwise, Bob wins. To ensure the game is finite, we can assume that Bob wins if any assignment of values repeats during the game.

[10] Game G_6 of (Stockmeyer and Chandra 1979), also called ABF (Alternating Boolean Formula game) by Kinnersley (2015).

Theorem 5.7 (Stockmeyer and Chandra 1979). *The Alternating* S*AT game (AS*AT*) is* E*XPTIME-complete.*

Stockmeyer and Chandra (1979) classified EXPTIME-complete games as Universal Games in the sense that "*if Y denotes one of these games and X denotes any member of a large class of combinatorial games (including Chess, Go, and many other games of popular or mathematical interest), then the problem of determining the outcome of X is reducible in polynomial time to the problem of determining the outcome of Y*".

The Activation game is EXPTIME-complete

We start with the following partizan game in graphs: the Activation game. Let G be a graph in which every vertex is *activated* or *deactivated.* The activation of a deactivated vertex v is a process in two steps: in the 1st step, v is activated but every neighbor of v is deactivated and, in the 2nd step, every deactivated vertex with no activated neighbor is activated. It is synchronous: all vertices update their status at the same time. Note that this affects only the vertices at distance 1 and 2 from v.

The Activation game is a partizan game in which the instance is a graph G and an integer k. Some vertices of G are labeled A or B and the others are unlabeled. The vertices of G are in an initial configuration: every vertex is activated or deactivated. Alice and Bob alternately activate a deactivated vertex in such a way that Alice (resp. Bob) can only activate vertices labeled A (resp. B). If the number of activated vertices is at most k at the beginning of some turn, Alice wins. If some configuration repeats, Bob wins. This game is easy in complete graphs K_n and complete bipartite graphs $K_{m,n}$, but in general it is very hard.

Theorem 5.8. *The Activation game is* E*XPTIME-complete.*

Proof. It is easy to see that it is in EXPTIME (Ex. 5.4). We obtain a reduction from ASAT with formula Φ over a set $X_A \cup X_B$ of n variables. Let $k = n$. Also let G the following graph. For every variable x_i, add to G two vertices x_i and $\overline{x_i}$. Label x_i and $\overline{x_i}$ with A if $x_i \in X_A$; otherwise, label them with B. Also add the edge $x_i\overline{x_i}$. For every clause c_j, create the unlabeled vertex c_j. If the clause c_j contains x_i, add the edge x_ic_j. If the clause c_j contains $\overline{x_i}$, add the edge $\overline{x_i}c_j$.

If x_i is true in the initial configuration of Φ, make the vertex x_i activated and the vertex $\overline{x_i}$ deactivated at the beginning. Otherwise, make the vertex

x_i deactivated and the vertex $\overline{x_i}$ activated. If the clause c_j is satisfied by the initial configuration of Φ, make the vertex c_j deactivated; otherwise, make it activated.

Notice that the activation of the vertex x_i deactivates $\overline{x_i}$ and every vertex c_j neighbor of x_i, and activates every vertex c_ℓ neighbor of $\overline{x_i}$ without activated neighbors. Also, no deactivated vertex x_ℓ (resp. $\overline{x_\ell}$) at distance two from x_i is activated, since $\overline{x_\ell}$ (resp. x_ℓ) is activated.

Therefore, the state activate or deactivate of a variable x_i is related to its value true or false. On the other hand, a clause vertex c_j is activated (resp. deactivated) if the clause c_j is unsatisfied (resp. satisfied). Furthermore, since either x_i or $\overline{x_i}$ is activated at each turn for any variable x_i, then the number of activated vertices is at least $k = n$. Then, if all clauses are satisfied, the number of activated vertices is exactly $k = n$.

If Alice wins ASAT, she can satisfy every clause of Φ and, by following the corresponding strategy in the Activation game, she can deactivate every clause vertex, winning the game. If Bob wins ASAT, he guarantees that there is at least one unsatisfied clause of Φ at each turn and, by following the corresponding strategy in the Activation game, he guarantees that there is at least one activated clause vertex at each turn, winning the game. ■

COPS AND ROBBER is EXPTIME-complete

In Chapter 9, we will study in detail the Cops and Robber game ($\mathcal{C}\&\mathcal{R}$). In this game, a graph G and an integer k are given, and there are two players: $\mathcal{C}$ (Cops) and $\mathcal{R}$ (Robber). Initially $\mathcal{C}$ places k cops on the vertices of G and then $\mathcal{R}$ places a robber on a vertex of G. The players then alternate moving their pieces between adjacent vertices of G. A player is allowed to decide not to move some of their pieces (or even all of them) on their turn. The cops win if a cop occupies the same vertex of the robber at some turn (in this case we say that the robber was captured). The robber wins if he can avoid capture indefinitely. To ensure the game is finite, we can assume that the robber wins if any configuration of the cops and the robber repeats during the game.

Deciding whether player $\mathcal{C}$ has a winning strategy with k cops on a graph $G = (V, E)$ can certainly be done by an exponential time algorithm on the game tree (Berarducci and Intrigila 1993; Hahn and MacGillivray 2006). As seen in Chapter 1, the algorithm has polynomial time if k is constant. Therefore, $\mathcal{C}\&\mathcal{R} \in$ EXPTIME.