Chapter 7

Supervision and Learning Attitudes

In philosophy, logic and psychology the word 'learning' is used to denote a variety of phenomena. In this chapter, as in the previous ones, we take the phrase 'to learn' to mean 'to acquire information in order to arrive at a certain (correct) conclusion'. In this we follow the lines of learning theory (see, e.g., Jain et al., 1999) where learning consists of a sequence of mind changes that should lead to a correct conclusion. All belief states visited on the way there, together with the correct one, are drawn from some given set of possibilities that constitutes the initial uncertainty range. To be more specific, let us try to describe in these terms the process of language learning. Agent L's (Learner's) aim is to 'arrive' at a grammar that correctly describes his native language. This scenario can be represented by a graph in which the vertices represent possible grammars and an edge from vertex $s_1$ to vertex $s_2$ labeled with $\alpha$ stands for a possibility of a mind change from grammar $s_1$ to $s_2$ that is triggered by the incoming information $\alpha$. The properties of such a graph are determined by features of the initial range of possibilities, the language being learned, and the nature of agent's learning strategy. Then, the process of learning can be represented simply as a 'journey' of $L$ through the graph until he finally reaches the correct grammar. Obviously, other inductive inference processes can also be described in this way.

The setting can be enriched by the presence of another agent, let us call her Teacher, $T$, who decides which data are presented to $L$, in which order, etc. In other words, we can introduce another player who supervises the process of learning and manipulates the data in order to influence the speed and accuracy of convergence. The analysis of the role of the teacher has been increasingly present in formal learning theory (e.g., see Angluin, 1987 for the minimally adequate teacher in learning from queries and counterexamples, and Balbach & Zeugmann, 2009 for recent developments in teachability theory).

In this chapter, we investigate the interaction between Learner and Teacher in a particular kind of supervision learning game that is played on a graph. Learner's information state changes while he moves around the graph, from one conjecture to another. Teacher, having a global perspective, knows the structure of the
7.1 Sabotage Games

As we already mentioned, our perspective on learning leads naturally to the framework of sabotage games. Sabotage games are useful for reasoning about various interactive processes involving random breakdowns or intentional obstruction in a system, from the failures of server networks to the logistics of traveling by public transport. We argue that it can also be interpreted positively, as some form of learning. But before we get to that, we first introduce the general, basic framework of sabotage games.

A sabotage game is played in a directed multi-graph, with two players, Runner and Blocker, moving in alternation, with Runner moving first. Runner moves by making a single transition from the current vertex. Blocker moves by deleting any edge from the graph. Runner wins the game if he is able to reach a designated goal vertex; otherwise Blocker wins.

To define the game formally let us first introduce the structure in which sabotage games take place, a directed multi-graph (see, e.g., Balakrishnan, 1997).

Definition 7.1.1. A directed multi-graph is a pair $G = (V, E)$ where $V$ is a set of vertices and $E : V \times V \to \mathbb{N}$ is a function indicating the number of edges between any two vertices.

The sabotage game is defined in the following way.

Definition 7.1.2 (Löding & Rohde 2003a). A sabotage game

$$\mathcal{SG} = (V, E, v_0, v_g)$$

is given by a directed multi-graph $(V, E)$ and two vertices $v_0, v_g \in V$. Vertex $v$ represents the initial position of Runner and $v_g$ represents the goal state (the aim of Runner).

Each match consists of a sequence of positions and is played as follows:

1. The initial position $(v_0, v_0)$ is given by $(E, v)$;
2. round $k + 1$ from position $(E_k, v_k)$ consists of:
   
   (a) Runner moving to some $v_{k+1}$ such that $E(v_k, v_{k+1}) > 0$, and then
   
   (b) Blocker removing an edge $(v, v')$ such that $E_k(v, v') > 0$.

The new position is $(E_{k+1}, v_{k+1})$, where $E_{k+1}(v, v') := E_k(v, v') - 1$ and, for every $(v, v') \neq (v, v'), E_{k+1}(v, v') := E_k(v, v')$.

3. The match ends if a player cannot make a move or if Learner reaches the goal state, which is the only case in which he wins.

In other words, Blocker removes an edge between two states $v, v'$ by decreasing the value of $E(v, v')$ by 1. We will see later, this description of the game based on the above definition of multi-graphs can lead to some technical problems when we want to interpret modal logic over these structures. Therefore, we will now present an alternative definition, which we later show to be equivalent with respect to the existence of a winning strategy.

Definition 7.1.3. Let $\Sigma = \{a_1, \ldots, a_n\}$ be a finite set of labels. A directed labeled multi-graph is a tuple $G^\Sigma = (V, E)$ where $V$ is a set of vertices and $\Sigma = (E_{a_1}, \ldots, E_{a_n})$ is a collection of binary relations $E_{a_i} \subseteq V \times V$ for each $a_i \in \Sigma$.

In the above definition, the labels from $\Sigma$ are used to represent multiple edges between two vertices; $E$ is simply an ordered collection of binary relations on $V$ with labels drawn from $\Sigma$. Accordingly, the modified definition of sabotage game is as follows.

---

1In what follows, we take the size of any multi-graph $G = (V, E)$ to be bounded by $|V| + \max\{E(v, w) \mid v, w \in V \setminus |V|^2\}$. 

---
Definition 7.1.4. A labeled sabotage game

\[ \text{SG}^2 = (V, E, v, v_g) \]

is given by a directed labeled multi-graph \((V, E)\) and two vertices \(v_0, v_g \in V\). Vertex \(v\) represents the initial position of Runner and \(v_g\) represents the goal state.2 Each move is played as follows:

1. The initial position \((E^0, v_0)\) is given by \((E, v)\);

2. Round \(k + 1\) from position \((E^k, v_k)\) with \(E^k = (E_{a_1}^k, \ldots, E_{a_n}^k)\), consists of:
   
   (a) Runner moving to some \(v_{k+1}\) such that \((v_k, v_{k+1}) \in E_{a_i}^k\) for some \(a_i \in \Sigma\) and then
   
   (b) Blocker removing an edge \((v, v')\) with label \(a_j \in \Sigma\) for some \(i \neq j\);

3. The match ends if a player cannot make a move or if Runner reaches the goal state, which is the only case in which he wins.

It is easy to see that both versions of sabotage games have the history-free determinacy property: if one of the players has a winning strategy then (s)he has a winning strategy that depends only on the current position. Then, each round can be viewed as a transition from a sabotage game \(\text{SG}^0 = (V, E, v_0, v_g)\) to another sabotage game \(\text{SG}^1 = (V, E^{k+1}, v_{k+1}, v_g)\), since all previous moves become irrelevant. We will use this fact throughout the whole paper.

It is easy to see that in labeled sabotage games, the label of the edge removed by Blocker is irrelevant with respect to the existence of a winning strategy. What matters is the number of edges that is left.

Observation 7.1.5. Let \(\text{SG}^0 = (V, E, v_0, v_g)\) and \(\text{SG}^2 = (V, E, v_0, v_g)\) be two labeled sabotage games that differ only in the labels of their edges, that is,

\[ \text{for all } (v, v') \in V \times V, |\{E_{a_i}^k | (v, v') \in E_{a_i}^k\}| = |\{E_{a_i}^k | (v, v') \in E_{a_i}^k\}|, \]

where \( | \cdot | \) stands for cardinality. Then Runner has a winning strategy in \(\text{SG}^2\) iff he has a winning strategy in \(\text{SG}^2\).

The existing results on sabotage have been given for the non-labeled version of the game. In what follows we show that the problems of deciding whether Runner has a winning strategy in sabotage games \(\text{SG}^2\) and \(\text{SG}^3\) are polynomially equivalent.

7.1. Sabotage Games

By doing this we establish that our modification of the definition makes only a slight difference and that the previous contribution is valid for our notion. We start by formalizing the two problems.

Definition 7.1.6 (Sabotage Decision Problem).

Instance Sabotage game \(\text{SG} = (V, E, v_0, v_g)\).

Question Does Runner have a winning strategy in \(\text{SG}^2\)?

The \(\Sigma\)-Sabotage Decision Problem is very similar. The only difference is that it is concerned with slightly modified structures — labeled sabotage games.

Definition 7.1.7 (\(\Sigma\)-Sabotage Decision Problem).

Instance Labeled sabotage game \(\text{SG}^2 = (V, E, v_0, v_g)\).

Question Does Runner have a winning strategy in \(\text{SG}^2\)?

Theorem 7.1.8. Sabotage and \(\Sigma\)-Sabotage are polynomially equivalent.

Proof. The two problems can be polynomially reduced to each other.

(\(\rightarrow\) ) Sabotage can be reduced to \(\Sigma\)-Sabotage. Given a sabotage game \(\text{SG} = (V, E, v_0, v_g)\), let \(m\) be the maximal number of edges between any two vertices in the graph, i.e.,

\[ m := \max \{ E(u, u') \mid (u, u') \in (V \times V) \}. \]

Then, we define the labeled sabotage game \(f(\text{SG}) := (V, E, v_0, v_g)\), where \(E := (E_1, \ldots, E_m)\) and each \(E_i\) is given by \(E_i := \{(u, u') \in V \times V \mid |E(u, u')| \geq i\}\).

We have to show that Runner has a winning strategy in \(f(\text{SG})\) iff he has one in \(f(\text{SG})\). The proof is by induction on \(m\) — the number of edges in \(\text{SG}\), i.e.,

\[ n := \sum_{i=0}^{\infty} \sum_{(u, u') \in V \times V} |E(u, u')|. \]

Note by definition of \(f(\text{SG})\) has the same number of edges, i.e.,

\[ n = \sum_{i=0}^{\infty} |E_i|. \]

The base case

Straightforward. In both games Runner has a winning strategy iff \(v_0 = v_g\).

The inductive case

(\(\Rightarrow\) ) Suppose that Runner has a winning strategy in the game \(\text{SG} = (V, E, v_0, v_g)\) with \(n + 1\) edges. Then, there is some \(v_i \in V\) such that \(E(v_i, v_i) > 0\) and Runner has a winning strategy for all games \(\text{SG}' = (V, E, v_0, v_g)\) that result from Blocker removing any edge \((u, u')\) with \(E(u, u') > 0\). Note that all such games \(\text{SG}'\) have just \(n\) edges, so by the induction hypothesis Runner has a winning strategy in \(f(\text{SG})\). But then, by Observation 7.1.5, Runner also has a winning strategy in all games \(f(\text{SG})\) that result from removing an arbitrary edge from \(f(\text{SG})\). This is so because for any removed edge \((u, u')\), the only possible difference between \(f(\text{SG})\) and \(f(\text{SG}')\) is in the labels of the edges between \(u\) and \(u'\) (in \(f(\text{SG})\) the removed label was the largest, in \(f(\text{SG}')\) the removed label is arbitrary). Now, by definition
of \( f \), choosing \( v_t \) is also a legal move for Runner in \( f(\Sigma^2) \) and, since he can win every \( f(\Sigma^2) \), he has a winning strategy in \( f(\Sigma^2) \).

(⇒) Runner having a winning strategy in \( f(\Sigma^2) \) means that he can choose some \( v_t \) with \( (v_0, v_t) \in E \), for some \( i \leq m \) such that he has a winning strategy in all games \( f(\Sigma^2) \) resulting from Blocker’s move. Choosing \( v_t \) is also a legal move of Runner in \( f(\Sigma^2) \). Suppose that Blocker replies by choosing \( (v', v') \). Let us call the resulting game \( \Sigma^2 \). By assumption and Observation 7.1.5, Runner also has a winning strategy in the game \( f(\Sigma^2) \) which is the result from Blocker choosing \((v_0, v') \) and \( E(v, v') \). Since \( f(\Sigma^2) = f(\Sigma^2) \), we can apply the inductive hypothesis.

(⇐) \( \Sigma \)-SABOTAGE can be reduced to SABOTAGE. Given a labeled sabotage game \( \Sigma^2 = (\Sigma, E, v, v_0) \), define the sabotage game \( f'(\Sigma^2) = (\Sigma, E', v, v_0) \), where \( E'(v, v') = \{ (v_0, v') \in E_0 \} \).

Showing that Runner has a winning strategy in \( \Sigma^2 \) iff he has one in \( f(\Sigma^2) \) is straightforward, and can be done by induction on the number of edges in \( \Sigma^2 \), i.e., on \( n := \sum_{e \in E} |E_e| \).

Finally, let us observe that both \( f \) and \( f' \) that encode the procedures of transforming one type of graph to another, are polynomial, so the proof is complete.

7.2 Sabotage Modal Logic

Sabotage modal logic (SML) has been introduced by Van Benthem (2005) to investigate the complexity of reachability-type problems in dynamic structures, such as the graph of our sabotage games. Besides the standard modalities, it also contains ‘transition-deleting’ modalities for reasoning about model change when a transition (an edge) is removed. To be more precise, we have formulæ of the form \( \varphi \rightarrow \psi \), expressing that it is possible to delete a pair from the accessibility relation such that \( \varphi \) holds in the resulting model at the current state.

Definition 7.2.1 (SML; Language; Syntax). Let \( \mathsf{Prop} \) be a countable set of propositional letters and let \( \Sigma \) be a finite set of labels. Formulæ of the language of sabotage modal logic are given by:

\[
\varphi ::= p \mid \lnot \varphi \mid \varphi \lor \psi \mid \Box_a \varphi \mid \Diamond_a \varphi
\]

with \( p \in \mathsf{Prop} \) and \( a \in \Sigma \). The formula \( \Box_a \psi \) is defined as \( \lnot \Diamond_a \lnot \psi \), and we will write \( \Box \varphi \) for \( \Box_{a \in \Sigma} \Box_a \varphi \) and \( \Diamond \varphi \) for \( \forall_{a \in \Sigma} \Diamond_a \varphi \).

The sabotage modal language is interpreted over Kripke models, that are here called sabotage models.

Definition 7.2.2 (Löding & Rohde 2003b). Given a countable set of propositional letters \( \mathsf{Prop} \) and a finite set \( \Sigma = \{ a_1, \ldots, a_n \} \), a sabotage model is a tuple \( M = (W, (R_a)_{a \in \Sigma}, \mathsf{Val}) \) where \( W \) is a non-empty set of worlds, each \( R_a \subseteq W \times W \)

7.3 Sabotage Learning Games

is an accessibility relation and \( \mathsf{Val} : \mathsf{Prop} \rightarrow \mathcal{P}(W) \) is a propositional valuation function. We will call a pair \((M, w)\) with \( w \in W \) a pointed sabotage model.

To get to the semantics of sabotage modal language, we first have to define the model that results from removing an edge.

Definition 7.2.3. Let \( M = (W, R_1, \ldots, R_n, \mathsf{Val}) \) be a sabotage model. The model \( M_{(v, v')}^\mathsf{Val} \) that results from removing the edge \((v, v') \in R_a \) is defined as \( M_{(v, v')}^\mathsf{Val} = (W, R_1, \ldots, R_n, \mathsf{Val} \setminus \{ (v, v') \}, R_a, \ldots, R_n, \mathsf{Val}) \).

Definition 7.2.4 (SML; Semantics). Given a sabotage model \( M = (W, R_a)_{a \in \Sigma}, \mathsf{Val}) \) and a world \( w \in W \), atomic propositions, negations, disjunctions and standard modal formulæ are interpreted as usual. For the case of ‘transition-deleting’ formulæ, we have:

\[
(M, w) = (\varphi\leftarrow) \text{ iff } \exists w' \in W ((v, v') \in R_a \land (M_{(v, v')}^\mathsf{Val}, w) \models \varphi).
\]

One SML result that is of great importance to us is the SML model checking complexity (combined complexity model checking, see Vardi, 1982). We will use it to reason about the difficulty of our learning scenarios.

Theorem 7.2.5 (Löding & Rohde 2003b). The computational complexity of model checking for SML is PSPACE-complete.

7.3 Sabotage Learning Games

In this section, we reinterpret the sabotage game in the broader perspective of learning. We introduce variants of the winning condition of the game. For each variant, we will provide a sabotage modal logic formula characterizing the existence of a winning strategy. We also prove complexity results for model checking in each case. We will work with previously introduced labeled sabotage games, using the labeling of the edges to represent different kinds of information changes that take Learner from one state into another.

7.3.1 Three Variations

A sabotage learning game is defined as follows.

Definition 7.3.1. A sabotage learning game (SLG) is a labeled sabotage game between Learner (\( L \), taking the role of Runner) and Teacher (\( T \), taking the role of Blocker). We distinguish three different versions, SLGtrue, SLGfalse and SLGtrue. Moves allowed for both players are those of the sabotage game. There is also no difference in the arena in which the game is played. However, the winning conditions vary from version to version (Table 7.1).
7.3. Sabotage Learning Games

Unhelpful Teacher and Eager Learner (SL2ue) Let us first consider SL2ue, the original sabotage game (Van Bentham, 2005). For any \( n \in \mathbb{N} \), we define the formula \( \gamma_n \) inductively as follows:

\[
\gamma_0 := \text{goal}, \quad \gamma_{n+1} := \text{goal} \lor \bigcirc \gamma_n.
\]

The following result is Theorem 7 of Löding & Rohde (2005b) rephrased for labeled sabotage games. We provide a detailed proof to show how our labeled definition avoids a technical issue present in the original proof.

**Theorem 7.3.3.** Learner has a winning strategy in the SL2ue game \( \mathcal{S}_2 \) if and only if \( M(\mathcal{S}_2), v_0 \models \gamma_n \) when \( n \) is the number of edges of \( \mathcal{S}_2 \).

**Proof.** The proof is by induction on \( n \).

**The base case**

\( \Rightarrow \) If \( L \) has a winning strategy in a graph \( \mathcal{S}_2 \) with no edges, then \( v_0 = v_g \). Thus, \( M(\mathcal{S}_2), v_0 \models \text{goal} \) and hence, \( M(\mathcal{S}_2), v_0 \models \gamma_0 \).

\( \Leftarrow \) If \( M(\mathcal{S}_2), v_0 \models \gamma_0 \) then \( M(\mathcal{S}_2), v_0 \models \text{goal} \). Since \( v_0 \) is the only state where \( \text{goal} \) holds, then we have \( v_0 = v_g \) and therefore \( \mathcal{S}_2 \) wins immediately.

**The inductive case**

\( \Rightarrow \) Suppose that \( \mathcal{S}_2 \) has \( n+1 \) edges, and assume \( L \) has a winning strategy. There are two possibilities: \( L \)'s current state is the goal state (that is, \( v_0 = v_g \), or it is not.

In the first case, we get \( M(\mathcal{S}_2), v_0 \models \text{goal} \) and hence \( M(\mathcal{S}_2), v_0 \models \gamma_{n+1} \).

In the second case, since \( L \) has a winning strategy in \( \mathcal{S}_2 \), there is some state \( v_0 \in V \) reachable from \( v_0 \), i.e., for some \( v_0 \in \Sigma \) that is, \( (v_0, v_1) \in E_0 \) such that in all edges \( E_0, (v_0, v_1) \) we get \( M(\mathcal{S}_2), (v_0, v_1) \models \text{goal} \).

All such games have \( n \) edges, so by inductive hypothesis we have

\[
M(\mathcal{S}_2, (v_0, v_1)) \models \gamma_n,
\]

for every edge \((v_0, v_1)\) and label \( a_i \). Now, the key observation is that each \( M \)-image of the graph that results from \( L \) moving to \( v_1 \) and \( T \) removing edge \((v_0, v_1)\) with label \( a_j \) is exactly the model that results from removing edge \((v_0, v_1)\) with \( a_j \) from the model \( M(\mathcal{S}_2) \).

Then, for all such \((v_0, v_1)\) and \( a_j \), we have

\[
M(\mathcal{S}_2, (v_0, v_1)) \models \gamma_n.
\]

\( \exists \text{The collection } E_0, (v_0, v_1) \text{ is given by } (a_1, a_2, \ldots, a_{n+1} - (v_0), (v_1), a_{n+2}, \ldots, a_{\text{end}}) \).

\( \text{In the original definition of a sabotage game this is not the case. In the game, the edges are implicitly ordered by numbers (the existence of an edge labeled with } k \text{ implies the existence of edges labeled with } 1, \ldots, k-1); \text{ in the model, this is not the case. When we remove an edge from a game we always remove the one with the highest label, but when we remove an edge from a model we remove an arbitrary one; the operations of removing an edge and turning a game into a model do not commute.} \)
It follows that \((M(SG^2), v_0) = \exists v \neq 0\) and therefore \((M(SG^2), v_0) = \emptyset \exists v \neq 0\), that is, \((M(SG^2), v_0) = \gamma_{m+1}'\).

(\rightarrow) Suppose that \((M(SG^2), v_0) = \text{goal} \cup \exists v \neq 0\). Then, \(v_0\) is the goal state or else there is a state \(v_1\) accessible from \(v_0\) such that \((M(SG^2), v_1) = \exists v \neq 0\), that is, \((M(SG^2)_u(v_0, v_1)) = v_0\), for all edges \((u, u')\) and labels \(a_i\). By inductive hypothesis, \(L\) has a winning strategy in each game that correspond to each pointed model \((M(SG^2)_u(v_0, v_1)) = v_0\). But these games are exactly those that result from removing any edge from the game \((V, E^0, v_0, v_1)\) after \(L\) moves from \(v_0\) to \(v_1\). Hence, \(L\) has a winning strategy in \((V, E^0, v_0, v_1)\), the game that corresponds to the pointed model \((M(SG^2), v_0)\), as required. □

Helpful Teacher and unwillig Learner (SLGu) Now consider SLGu, the game with helpful Teacher and unwillig Learner. We define \(\gamma^m_0\) inductively, as
\[\gamma^m_0 := \text{goal}, \quad \gamma^m_{m+1} := \text{goal} \cup (\emptyset \exists \gamma^m_0)\]
In this case, Teacher has to be sure that Learner does not get stuck before he has reached the goal state — this is why the conjunct \(\emptyset \exists \gamma^m\) is needed in the definition of \(\gamma^m_{m+1}\). We show that this formula corresponds to the existence of a winning strategy for Teacher.

**Theorem 7.3.4.** Teacher has a winning strategy in the SLGu
\[SG^2 = (V, E^0, v_0, v_1)\]
if and only if \((M(SG^2), v_0) = \gamma^m_0\), where \(m\) is the number of edges of \(SG^2\).

Proof. Similar to the proof of Theorem 7.3.3. □

Helpful Teacher and eager Learner (SLGhe) Finally, for SLGhe, the corresponding formula is defined as
\[\gamma^m_0 := \text{goal}, \quad \gamma^m_{m+1} := \text{goal} \cup (\emptyset \gamma^m_0)\]

**Theorem 7.3.5.** Teacher and Learner have a joint winning strategy in SLGhe
\[SG^2 = (V, E^0, v_0, v_1)\]
if and only if \((M(SG^2), v_0) = \gamma^m_0\), where \(n\) is the number of edges of \(SG^2\).

Proof. L and T have a joint winning strategy if and only if there is a path from \(v_0\) to \(v_1\). From left to right this is obvious. From right to left, if there is such a path, then there is also one without cycles, and a joint winning strategy is the one that follows the path and at each step removes the edge that has just been used (it is essential that \(L\) moves first). The theorem follows by observing that \(\gamma^m_0\) expresses the existence of such path. □

The above results for the three scenarios are summarized in Table 7.2.

7.3. Sabotage Learning Games

<table>
<thead>
<tr>
<th>Game</th>
<th>Winning Condition in SML</th>
<th>Winner</th>
</tr>
</thead>
<tbody>
<tr>
<td>SLGu</td>
<td>(\gamma^m_0 := \text{goal}), (\gamma^m_{m+1} := \text{goal} \cup (\emptyset \exists \gamma^m_0))</td>
<td>Learner</td>
</tr>
<tr>
<td>SLGhe</td>
<td>(\gamma^m_0 := \text{goal}), (\gamma^m_{m+1} := \text{goal} \cup (\emptyset \gamma^m_0))</td>
<td>Teacher</td>
</tr>
<tr>
<td>SLGhe</td>
<td>(\gamma^m_0 := \text{goal}), (\gamma^m_{m+1} := \text{goal} \cup (\emptyset \exists \gamma^m_0))</td>
<td>Both</td>
</tr>
</tbody>
</table>

Table 7.2: Winning Conditions for SLG in SML

7.3.3 Complexity of Sabotage Learning Games

We have characterized the existence of a winning strategy in our three versions of SLG by means of sabotage modal logic formulae. In this section, we investigate the complexity of deciding whether such formulae are true in a given pointed model, i.e., the complexity of checking whether there is a winning strategy in the corresponding game.

By Theorem 7.2.5, the model checking problem of sabotage modal logic is PSPACE-complete. This gives us PSPACE upper bounds for the complexity of deciding whether a player can win a given game. In the three cases, we can also give tight lower bounds.

Unhelpful Teacher and eager Learner (SLGhe) For SLGhe, which can be identified with the standard sabotage game, PSPACE-hardness is shown by reduction from Quantified Boolean Formula (QBF) (Löding & Rohde 2003b).

**Theorem 7.3.6.** (Löding & Rohde 2003b). SLGhe is PSPACE-complete.

Helpful Teacher and unwillig Learner (SLGu) Whereas at first sight, SLGu and SLGhe might seem to be duals of each other, the relationship between them is more complex due to the different nature of the players' moves: Learner moves locally by choosing an state accessible from the current one, while Teacher moves globally by removing an arbitrary edge. Nevertheless, we can show PSPACE-hardness for SLGu. In the proof we will use the Quantified Boolean Formula (QBF) problem, known to be PSPACE-complete.

**Definition 7.3.7 (Quantified Boolean Formula Problem).**

**Instance** Let \(\varphi\) be an instance of QBF, i.e., a formula:
\[\varphi := \exists x_1 \exists x_2 \exists x_3 \ldots Q x_n \psi\]
where \(Q = \exists\) for \(n\) odd, and \(\forall\) for \(n\) even, and \(\psi\) is a quantifier-free formula in conjunctive normal form.

**Question** Is \(\varphi\) satisfiable?

\[\]
Theorem 7.3.8. SLGn is PSPACE-complete.

Proof. From Theorem 7.2.5 and Theorem 7.3.4 it follows that SLGn is in PSPACE. PSPACE-hardness of SLGn is proved by showing that the Quantified Boolean Formula (QBF) problem, can be polynomially reduced to SLGn.

We will construct a directed game arena for SLGn, such that Learner has a winning strategy in the game iff the formula \( \varphi \) is satisfiable.

The 3-gadget. Figure 7.1 represents the situation in which Learner chooses the assignment for \( x_i \) when \( i \) is odd. This part corresponds to assigning the value to an existentially quantified variable. Learner starts in \( A \) and moves either left or right, if he wants to make \( x_i \) true, then he moves to \( X_i \), otherwise to \( X_i \). Let us assume that he moves right, towards \( X_i \). Then Teacher has exactly four moves to remove all the edges leading to the dead-end #. At this point, Teacher cannot remove any edge in some other place in the graph without losing. So, Learner reaches \( X_i \), and Teacher is forced to remove the edge that leads from \( B \) to \( X_i \), because otherwise Teacher would allow Learner to reach the dead-end #. At this point Learner moves towards \( B \), and in the next step exits the gadget. Moving back towards \( X_i \) would cause him to lose, because then Teacher could remove the edge between \( B \) and \( X_i \), and Learner would be forced to enter the goal.

The V-gadget. Figure 7.2 represents the situation in which Teacher chooses the assignment for \( x_i \) when \( i \) is even. This part corresponds to assigning the value to a universally quantified variable.

Let us assume that Teacher wants to make \( x_i \) false. Then she leads Learner towards \( X_i \) by successively removing the edges between \( C \) and \( X_i \). When Learner already is on the path to \( X_i \), Teacher starts removing an edge going from \( X_i \) to

\[ \text{ Figure 7.2: The V-gadget. } \]

The verification gadget. Figure 7.3 represents the situation in which Teacher chooses one clause from \( \psi \). If Teacher chooses \( c \), then Learner can choose one literal \( x_i \) from \( c \). There are edges from \( x_i \) to an S-gadget if \( i \) is odd, and to a V-gadget otherwise, leading directly to \( X_i \), if \( x_i \) is positive in \( c \) and to \( X_i \) otherwise. So, if the chosen assignment satisfies \( \psi \), then for all clauses there is at least one literal which is true, and leads to the opposite truth value in the corresponding gadget, from which in turn Learner can get to the dead-end # (there are four edges left) and win the game.

For the converse, if the chosen assignment does not satisfy \( \psi \), then Learner gets to a corresponding point in a proper gadget, Teacher removes the edge from this point to \( B \), and the only option left for Learner is to enter the goal, which means that he loses the game.
7.4 Relaxing Strict Alternation

As mentioned above, in sabotage (learning) games the players' moves are asymmetric: Learner moves locally (moving to a vertex accessible from the current one) while Teacher moves globally (removing any edge from the graph, and thereby manipulating the space in which Learner is moving). Intuitively, both for a helpful and an unhelpful Teacher, it is not always necessary to react to a move of Learner by giving immediate feedback (here, by removing an edge). This leads us to a variation of SLG in which Learner's move need not in principle be followed by Teacher's move, i.e., Teacher has the possibility of skipping a move.

Definition 7.4.1. A sabotage learning game without strict alternation (for $T$) is a tuple $\text{SLG}^*$ = $(V, E, v_0, v_T)$. Moves of Learner are as in SLG and, once he has chosen a vertex $v_l$, Teacher can choose between removing an edge, in which case the next game is given as in SLG, and doing nothing, in which case the next game is $(V, E, v_l, v_T)$. Again, there are three versions with different winning conditions, now called SLG*ue, SLG*hu and SLG*he.

After defining the class of games $\text{SLG}^*$, the natural question arises of how the winning abilities of the players change from SLG to $\text{SLG}^*$, since in the latter

Table 7.3. Complexity Results for Sabotage Learning Games

<table>
<thead>
<tr>
<th>Game</th>
<th>Winning Condition</th>
<th>Complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{SLG}^*\text{ue}$</td>
<td>Learner wins iff he reaches the goal state, Teacher wins otherwise</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>$\text{SLG}^*\text{hu}$</td>
<td>Teacher wins iff Learner reaches the goal state, Learner wins otherwise</td>
<td>PSPACE-complete</td>
</tr>
<tr>
<td>$\text{SLG}^*\text{he}$</td>
<td>Both players win iff Learner reaches the goal state. Both lose otherwise</td>
<td>NL-complete</td>
</tr>
</tbody>
</table>

Helpful Teacher and eager Learner ($\text{SLG}^*\text{he}$) Finally, let us have a look at $\text{SLG}^*\text{he}$. This game is different from the two previous ones: $L$ and $T$ win or lose together. Then, a winning strategy for each of them needs to take into account all possible moves of the other. This suggests that this version should be less complex than $\text{SLG}^*\text{ue}$ and $\text{SLG}^*\text{hu}$.

As mentioned in the proof of Theorem 7.3.5, $L$ and $T$ have a joint winning strategy if and only if the goal vertex is reachable from $L$'s position. Thus, determining whether they can win $\text{SLG}^*\text{he}$ is equivalent to solving the REACHABILITY ($\text{ST CONNECTIVITY}$) problem, which is known to be non-deterministic logarithmic space complete (NL-complete) (Papadimitriou, 1994).

Theorem 7.3.9. $\text{SLG}^*\text{he}$ is NL-complete.

Proof. Polynomial equivalence of $\text{SLG}^*\text{he}$ and REACHABILITY follows from the argument given in the proof of Theorem 7.3.5.

Table 7.3 summarizes the complexity results for the different versions of SLG. The complexity of these problems can be interpreted as the difficulty of deciding whether certain aims of Teacher can be fulfilled. We can attribute the question of the existence of the winning strategy to Teacher, as she already has the global perspective on the situation anyway. Our results say how difficult it is for her to decide whether there is a chance of success. The results agree with our intuition, as coming up with a strategy to teach is easier if Learner and Teacher cooperate.

Following our interpretation it is the easiest for Teacher to check whether the teaching will work out if Teacher assumes eagerness of Learner and she herself does her best to ensure that he succeeds. Moreover, the remaining two cases turn out to be equally difficult — it is as difficult to decide whether Teacher can force an unwilling Learner to learn as it is to decide whether an eager Learner can learn in the presence of an unhelpful Teacher.

From the perspective of the standard sabotage games, our complexity result for $\text{SLG}^*\text{he}$ means that with an additional safety\footnote{Safety concerns those properties that we want to hold throughout the process, in this case it is $T$ reaching the goal. Safety is usually contrasted with reachability, which requires the system to get into a certain configuration at some moment, here $L$ reaching the goal (see, e.g., Rachmeler & Thomas, 2008).} winning condition, sabotage games are PSPACE-complete.
Teacher can choose between removing an edge or doing nothing. In the rest of this section, we show that for all three winning conditions (SLG^u, SLG^h, S^L^G^o), the winning abilities of the players remain the same as in the case in which players move in strict alternation. This is surprising in the SLG^o case. It is by no means obvious that if Teacher has a winning strategy in the game without strict alternation then she also has one in the regular version of the game, because we might expect that removing an edge instead of skipping the move could result in blocking the way to the goal.

We start with the case of an unhelpful Teacher and an eager Learner, SLG^u. Note that although in this new setting matching can be infinite, e.g., in the game with unwilling L, if T skips her moves indefinitely, L will just keep moving, and hence the game will not be finished in finite time. However, in fact if Learner can win the game, he can do so in a finite number of rounds. We start with a lemma stating that if Learner can win some SLG^u in some number of rounds, then he can do so also if the underlying multi-graph has additional edges.

**Definition 7.4.2.** Let \( \Sigma = \{a_1, \ldots, a_k\} \) be a finite set of labels. For directed labeled multi-graphs, \( G^2 = (V, E) \) and \( G'^2 = (V', E') \), we say that \( G'^2 \) is a subgraph of \( G^2 \) if \( V \subseteq V' \) and \( E_{a_i} \subseteq E'_{a_i} \) for all labels \( a_i \in \Sigma \).

**Lemma 7.4.3.** If Learner has a strategy for winning the SLG^u \( (V, E, v_0, v_0) \) in at most \( n \) rounds, then he can also win any SLG^u \( (V', E', v_0, v_0) \) in at most \( n \) rounds, where \( (V, E) \) is a subgraph of \( (V', E') \).

**Proof.** The proof is by induction on \( n \). In the inductive step, for the case that \( T \) removes an edge which was not in the original multi-graph, note that the resulting graph is a supergraph of the original one. Then we can use the inductive hypothesis.

**Theorem 7.4.4.** Let us consider the SLG \( (V, E, v_0, v_0) \) with \( (V, E) \) being a directed labeled multi-graph and \( v_0, v_0 \in V \). Learner has a winning strategy in the corresponding SLG^u if he has a winning strategy in the corresponding SLG^u.

**Lemma 7.4.5.** Consider the SLG^h \( (V, E, v_0, v_0) \). If there is a path from \( v_0 \) to \( v_0 \) and there is no path from \( v_0 \) to a state from where \( v_y \) is not reachable, then Teacher has a winning strategy.

**Proof.** Let us assume that all states reachable from \( v_0 \) are on paths to \( v_0 \). Then, even if \( T \) removes from removing any edge, \( L \) will be on a path to the goal. Now, since the path to the goal does not include a loop or it does if it does not then \( T \) can simply wait until \( L \) arrives at the goal. If it does, \( T \) can remove the edges that lead to the loops in such a way that \( v_y \) is still reachable from any vertex.

**Lemma 7.4.6.** For all SLG^h \( (V, E, v_0, v_0) \), if Teacher has a winning strategy and there is an edge \( (v, v') \in E \) for some \( a \in \Sigma \) such that no path from \( v_0 \) to \( v_0 \) were \( (v, v') \), then Learner also has a winning strategy in \( (V', E', v_0, v_0) \), where \( E' \) results from removing \( (v, v') \) from \( E \).

**Proof.** If \( v \) is not reachable from \( v_0 \), it is easy to see that the claim holds. Assume that \( v \) is reachable from \( v_0 \). T's winning strategy should prevent \( L \) from moving from \( v \) to \( v' \) (otherwise \( L \) wins). Hence, \( T \) can also win if \( (v, v') \) is not there.

**Theorem 7.4.7.** If Teacher has a winning strategy in the SLG^h \( (V, E, v_0, v_0) \), then she also has a winning strategy in which she removes an edge in each round.

**Proof.** The proof is by induction on the number of edges \( n = \sum_{a \in \Sigma} |E_a| \).

**The base case**

**The inductive case**

**Assume that**

**T has a winning strategy in SLG^h \( (V, E, v_0, v_0) \) with \( \sum a \in \Sigma |E_a| = n = 1 \).

If \( v_0 = v_y \), it is obvious. Otherwise, since \( T \) can win, there is some \( v_1 \in V \) such that \( (v_0, v_1) \in E_a \) for some \( a \in \Sigma \) and for all such \( v_1 \) we have:

1. There is a path from \( v_1 \) to \( v_y \), and
2. (a) \( T \) can win \( (V, E, v_1, v_y) \), or (b) there is a \( (v, v') \in (V \times V) \times \Sigma \) such that \( (v, v') \in E_a \) and \( T \) can win \( (V, E', v_0, v_0) \) where \( E' \) is the result from removing \( (v, v') \) from \( E_a \).

If \( 2b \) holds, since \( \sum a \in \Sigma |E_a| = n \), we are done — we use the inductive hypothesis to conclude that \( T \) has a winning strategy in which she removes an edge in each round (in particular, her first choice is \( (v, v') \)). Let us show that \( 2b \) holds.

If there is some \( (v, v') \in V \times V \) such that \( (v, v') \in E_a \) for some \( a \in \Sigma \) and this edge is not part of any path from \( v_0 \) to \( v_y \) then by Lemma 7.4.5, \( T \) can remove this edge and \( 2b \) holds, so we are done.

If all edges in \( (V, E) \) belong to a path from \( v_1 \) to \( v_y \), from 1, there are two cases: either there is only one, or there is more than one path from \( v_1 \) to \( v_y \).
In the first case (only one path) \((u_0, u_1)\) can be chosen since it cannot be part of the unique path from \(v_1\) to \(v_2\). Assume now that there is more than one path from \(v_1\) to \(v_2\). Let \(p = (v_1, \ldots, v_k)\) be the shortest path from \(v_1\) to \(v_2\). This path cannot contain any loops. Then, from this path take \(v_i\) such that \(i\) is the smallest index for which it holds that from \(v_1\) there is a path \((v_1, v_{i+1}, \ldots, v_k)\) to \(v_i\) that is at least as long as the path following \(p\) from \(v_i\) (i.e., \((v_i, v_{i+1}, \ldots, v_k)\)).

Intuitively, when following path \(p\) from \(v_1\) to \(v_2\), \(v_1\) is the first point at which one can deviate from \(p\) in order to take another path to \(v_2\) (recall that we consider the case where every vertex in the graph is part of a path from \(v_1\) to \(v_2\)). Now it is possible for \(T\) to choose \((v_1, v_2, \ldots, v_{k-1})\) in \(E_k\). Let \(E'\) be the resulting set of edges after removing \((v_1, v_{i+1})\) from \(E_k\). Then we are in the game \((V, E', v_1, v_2)\). Note that due to the way we chose the edge to be removed, in the new graph it still holds that from \(v_0\) there is no path to a vertex from which \(v_0\) is not reachable (this holds because from \(v_1\) the goal \(v_2\) is still reachable) and by Lemma 7.4.5, \(T\) can win \((V, E', v_1, v_2)\), which then implies 2b.

Hence, we conclude that 2b is the case and thus using the inductive hypothesis, \(T\) can win \((V, E, v_0, v_2)\) also by removing an edge in every round. \(\square\)

**Corollary 7.4.8.** Teacher has a SLG\(^{\text{hu}}\) winning strategy in \((V, E, v_0, v_2)\) if she has a SLG\(^{\text{hu}}\) winning strategy.

As the reader might have noticed, the result that if Teacher can win a SLG\(^{\text{hu}}\), then she can also win the corresponding SLG\(^{\text{hu}}\), relies on the fact that Learner is the first to move. For instance, in a graph with two vertices and one edge — leading from the initial vertex to the goal vertex — if Teacher was to move first, she can win the SLG\(^{\text{hu}}\) only by skipping the move.

Finally, let us consider the case of helpful Teacher and eager Learner.

**Theorem 7.4.9.** If Learner and Teacher have a joint SLG\(^{\text{hu}}\) winning strategy in \((V, E, v_0, v_2)\), then they have a joint SLG\(^{\text{hu}}\) winning strategy.

**Proof.** If the players have a joint SLG\(^{\text{hu}}\) winning strategy, then there is an acyclic path from \(v_0\) to \(v_2\), which \(L\) can follow. At each round, \(T\) has to remove the edge that has just been used by \(L\). \(\square\)

Let us briefly conclude this section. We have shown that allowing Teacher to skip moves does not change the winning abilities of the players. Using these results, both the complexity and definability results from the previous section also apply to their versions without strict alternation, in which Teacher can skip a move.

### 7.5 Conclusions and Perspectives

We have provided a game theoretical approach to learning that takes into account different levels of cooperativeness between Learner and Teacher in a game of perfect information based on sabotage games. Because of our new interpretation we were able to define sabotage learning games with three different winning conditions. Then, following the strategy of Löding & Rohde (2003b), we have shown how sabotage modal logic can be used to reason about these games and, in particular, we have identified formulas of the language that characterize the existence of winning strategies in each of the two remaining cases. We also provided complexity results for the model-checking problem of these formulas. Our complexity results support the intuitive claim that cooperation of agents facilitates learning. Moreover, in our framework it turns out to be as difficult to decide whether a Teacher can force an unwilling Learner to learn as it is to decide whether an eager Learner can learn in the presence of an unhelpful Teacher.

Viewed from the perspective of the standard sabotage games, our complexity result for the game between a helpful Teacher and an unwilling Learner means that also with a safety winning condition, sabotage games are PSPACE-complete.

From the game-theoretical perspective, sabotage learning games can be extended to more general scenarios by relaxing the strict alternation. The moves of the players are of a different nature. Learner's moves can be seen as internal ones, moving to a state that is reachable from the current one, while Teacher's moves can be interpreted externally, removing any edge of the underlying graph. Once this asymmetry is observed, it becomes natural to ask what happens if from time to time Learner's move is not followed by Teacher's move (e.g., Learner can perform several changes of his information state before Teacher makes a restriction). Our results of Section 7.4 show that if we allow Teacher to skip a move, the winning abilities of the players do not change with respect to the original versions of the games. In the case of helpful Teacher and unwilling Learner, the result is quite surprising since it says that if Teacher can force Learner to learn in the game with non-strict alternation, then she can also do it when she is forced to remove edges in each round. This result crucially depends on the fact that Learner is the first to move, and does not hold in case Teacher starts the game.

In this chapter, we have described the learning process purely as changes in information states, without going further into their epistemic and/or doxastic interpretation.

We understand successful learning as the ability to reach an appropriate information state, not taking into account what happens afterwards. Formal learning theory that treats the inductive inference type of learning situates our work close to the concept of finite identification (Muranichi, 1992) treated extensively in previous chapters. In particular, we are not concerned with the stability of the resulting state. Identification in the limit (Gold, 1967) extends finite identification by looking beyond reachability in order to describe 'ongoing behavior'. Fixed-point logics, such as the modal \(\mu\)-calculus (Kozen, 1983; Scott & Bakker, 1969), can provide us with tools to express this notion of learnability. Further work involves investigating how fixed points can enrich sabotage-based learning analysis.

Moreover, in natural learning scenarios, e.g., language learning, the goal of
Chapter 8

The Muddy Scientists

Imagine you are one of ten prisoners locked up for extensive use of logic. To make you even more miserable, the guard comes up with a puzzle. He gathers all ten of you and says: ‘Each of you will be assigned a random hat, either black or white. You will be lined up single file where each can see the hats in front of him but not behind. Starting with the prisoner in the back of the line and moving forward, you must each, in turn, say only one word which must be ‘black’ or ‘white’. If the word you uttered matches your hat color you are released, if not, you are killed on the spot. You have half an hour to pray for your life.’ Then he leaves. One of the prisoners says: ‘I have a plan! If you agree on it, 9 of us 10 will definitely survive, and the remaining one has a 50/50 chance of survival.’ What does he have in mind?

Most probably the strategy that he wants to implement is as follows. First, the prisoners have to agree on the following meaning of the utterance of the one who is the last in the line. If he says ‘white’, it means that he sees an even number of white hats in front of him. If he says ‘black’ it means that he sees an odd number of white hats in front of him. Hence, his utterance has nothing to do with what he thinks his own hat is. There is a 50/50 chance of the total number of black hats being odd or even, and a 50/50 chance of his hat being black or white, so his chance of survival is the same. However, after this utterance the prisoner that stands in front of him knows for sure the color of his hat—he compares the utterance of his predecessor with the number of white hats he sees in front of him. If the parity is the same, he concludes that his hat is black, otherwise it is white. He makes his guess aloud. Now the person in front of her takes into account the first announcement and the second utterance, sees the number of white hats in front of her, and now she is also certain about her hat’s color, etc.

This epistemic scenario shows the power of multi-agent information exchange. A very simple quantitative public announcement carries powerful qualitative information. Agents can easily deduce nontrivial facts from implicit and indirect information. Obviously, the information has to be relevant to make certain deductions possible. For example, in the above scenario announcing ‘At least 5