

Teaching Via the Web: A Self-Evaluation Game Using Java for Learning Logical Equivalence

Alessandro Agostini
Dipartimento di Matematica
Università di Siena
Italy
agostini@unisi.it

Marco Aiello
Institute for Logic, Language and Computation
Universiteit van Amsterdam
the Netherlands
aiellom@wins.uva.nl

*Don't worry about your difficulties
in mathematics; I can ensure you
that mine are still greater.*

A. Einstein

1 A Motivating Question

We do not know if Albert Einstein's sentence could be useful to students for overcoming the wall of their ignorance in mathematics, or to face with the psychological block that prevents them from efficient learning. In this section, rather, we provide some basic motivation to use games for the learning of logic. A question arises: Why games?

A quick answer is on offer. Learning is a cognitive process, a "mental action," and it is well expressed by playing a game. Games involve competition, death and victory, deception and frustration, emotions—exactly as in real life. Games in logic, or logical games as a general case, are a natural learning environment, constructive situations, a "cognitive laboratory for studying deliberation, action, communication and information flow" [van Benthem, 1998], p.7. Besides more specific studies concerning the impact of emotions in the learning of mathematics, it is a fact that games are widely used in primary school teaching. As referred by [Hodges, 1998], Dienes and Golding relied on the funny side of games when they published the volume *Learning logic, logical games* for teaching logical notions to primary school children. For example, in one of their games a child has to stand up and try to list *things that he is not*. The other children listen to catch him out when he makes a mistake: the first child to spot a mistake takes over and has to list things that *she is not*. And so on.

We believe that there is a link between games and learning (of logic) that it is time to wake up to. At present, most of the education institutions involved in the teaching of logic encounter difficulties that sometimes grow with the students' age. At the same time, the use of games in teaching decreases with the student's age. To see an example, one can compare Dienes and Golding's book with the teaching material available today at the secondary and undergraduate levels.

As far as we know, the use of games in high school and undergraduate courses is not a very common experience. Nevertheless, there is a rich literature and some interesting tools on games as an explanation of concepts from logic, language and computation. For instance, [Doets, 1996, Hodges, 1997, Ebbinghaus and Flum, 1995] give three different ways to present Ehrenfeucht-Fraïssé games. These games are the main interest of this paper. Other games include semantic evaluation games (see for instance [Hintikka and Sandu, 1997] for a survey) and dialogue games for validity. For the semantic evaluation games, Tarski's World [<http://www.csli.stanford.edu/hp/Tarski2.html>] provides a simple game that students can use when a sentence evaluates in a way they did not expect. In dialogue games, the validity of some given formula is examined in terms of a two person, perfect information game. To the best of our knowledge, [van Benthem, 1998, Hodges, 1998] are the only lecture notes available at present that use games in logic and language as a unified perspective; [Abramsky, 1997] does the same for computation.

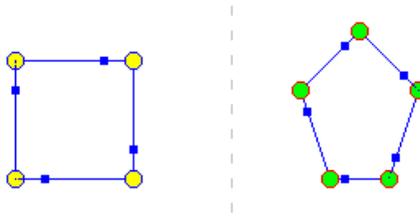


Figure 1: Example 1.

In this paper, we focus on games to be used in teaching logic. As a main result, we provide a Java application for playing logic in Ehrenfeucht-Fraïssé games as defined on graphs. Our system is available on the Web at [<http://www.wins.uva.nl/aiellom/java/ef>].

2 EF Games

Ehrenfeucht-Fraïssé (*EF*) games are two player games. To fix intuitions, let us baptize the players in the game as Professor and Student—we call this paradigm the examination paradigm. Professor and Student play a game of some length on two structures, say \mathcal{A} and \mathcal{B} . Professor wants to test the ability of Student as being examined on the question: Do you know how to compare the structures \mathcal{A} and \mathcal{B} ? A slightly different paradigm is the problem solving paradigm, where Student is being examined by Professor in solving a problem and either ‘ \mathcal{A} is similar to \mathcal{B} ’ or ‘ \mathcal{A} differs from \mathcal{B} .’ Whatever paradigm one chooses it makes no difference here. Professor and Student play as follows. Professor and Student take turns to choose elements from the structures. At each turn, Professor moves first and chooses an element from either \mathcal{A} or \mathcal{B} . Student replies by choosing an element of the other structure. Student loses if the set of atomic formulas satisfied by the elements chosen so far from one structure differs from the set of formulas satisfied by those elements chosen in the other structure (regardless of which player chose which elements). We refer to [Doets, 1996] for a general discussion on Ehrenfeucht-Fraïssé games. What is essential to define, rather, is the notion of strategy. A *strategy* for a player in a game is a set of rules which tells the player how to move, depending on what has happened so far. A strategy is *winning* if the player that uses it wins every play of the game.

To generalize our discussion, in what follows we shall refer to Professor and Student to their well known names. Thus, Professor is for *Spoiler* and Student is for *Duplicator*.

2.1 Some Examples

Is it time to play? By now we know the rules and we know that we can gain from playing, so it appears to be the time for some real games.

Example 1. Suppose that the set $A = \{a, b, c, d\}$ equipped with a relation $R_A = \{\langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle, \langle d, a \rangle\}$ and $B = \{a, b, c, d, e\}$ with $R_B = \{\langle a, b \rangle, \langle b, c \rangle, \langle c, d \rangle, \langle d, e \rangle, \langle e, a \rangle\}$ are given (as shown in [Fig. 1]).

The resulting structures are different, but the question is: how much different? In other words, how many turns will *Spoiler* need to win?

A possible play would develop as follows:

1. *Spoiler* chooses a in A , *Duplicator* answers b in B . Nothing very interesting happened here, both players just chose an element. Intuitively, both players thought: “exists x ” in terms of the language, which holds of both sets A and B .
2. *Spoiler* chooses c in A , *Duplicator* answers d in B . Here *Spoiler* knows that $\forall x \exists y \langle x, y \rangle \notin R$ is true in both structures. So, he can’t win yet, but he is preparing his victory.

- Spoiler chooses d in A , Duplicator has no good moves that leave a partial isomorphism. Spoiler's final win is justified by his thinking: "Well, the structure with domain A can be described by $\forall x \exists z (\exists y (\langle x, y \rangle \in R \wedge \langle y, z \rangle \in R) \wedge (\exists w (\langle z, w \rangle \in R \wedge \langle w, x \rangle \in R)))$, while the structure with domain B can not."

In the next section, we justify formally why Spoiler's way of thinking brought him to win.

Example 2. Suppose $A = \{1, 2, 3\}$, $B = \{8, 9, 10, 11\}$ and $<$ be the usual linear order. A possible play is the following:

- Spoiler chooses 2 in A , Duplicator answers 9 in B ;
- Spoiler chooses 11 in B , Duplicator answers 3 in A ;
- Spoiler chooses 10 in B , Duplicator has no moves.

Looking at this play and at the thoughts that drive Spoiler and Duplicator, it should give us hints of the underlying strategy they are using. The thoughts of the two players in this example are analogous to those highlighted in Example 1. One thing must be noticed though: the formula $\exists x \exists y \exists z \exists w : x < y < z < w$ does not describe the structure $\langle A, < \rangle$ but the structure $\langle B, < \rangle$. This formula is quite easy to write by looking at the picture of the two structures [Fig.2]; but one can do better and find more elegant formulas. Finding these formulas is a pedagogically good exercise. In this example, one is $\exists y \exists z : (y < z \wedge (\exists x : x < y \wedge \exists w : z < w))$. The elegance resides in the quantifier rank of the formula, notion that we define formally in the next section.

Example 2 is an instance of a more general game that can be played: the game on two structures having domains of different cardinality and equipped with linear ordering.

Example 3. A pedagogically interesting example is that of playing on the integers \mathbb{Z} and the rationals \mathbb{Q} equipped with the usual $<$ ordering relation.

As in the above example Spoiler has a winning strategy for all games of 3 or more moves. This is due to density, a property that \mathbb{Q} has, but \mathbb{Z} doesn't, and that can be expressed by the first order formula: $\forall x \forall y : (x < y \Rightarrow \exists z : x < z < y)$. Again, notice that the number of nested quantifier is 3, like the number of turns that Spoiler needs to win.

Example 4. Another appealing example of games on ordered sets is that playable on the rationals \mathbb{Q} and the reals \mathbb{R} equipped with the usual linear order $<$. It is a fact of model theory that their models $\langle \mathbb{Q}, < \rangle$ and $\langle \mathbb{R}, < \rangle$ cannot be distinguished in first order logic. This means that it is not possible to write a first order formula that is true in $\langle \mathbb{R}, < \rangle$ and false in $\langle \mathbb{Q}, < \rangle$. For instance, we can express that the rationals and the reals are dense. But there is no way to express, say, that between $314/100$ and $315/100$ there is a number representing the ratio between the circumference and its radius.

One can familiarize oneself quite easily with the notion of first order expressibility by playing an *EF* game on $\langle \mathbb{Q}, < \rangle$ and $\langle \mathbb{R}, < \rangle$ as follows: Spoiler tries his best in as many rounds as he wants; but if Duplicator knows how to play, Spoiler has no chance of winning. Thus, Duplicator has a winning strategy for every length of the game, which means that every first order formula which is true on the rationals is also true on the reals.

In terms of learning and getting feelings of model-theoretic concepts, it is useful to play a game on structures where the domain is a set of numbers. Moreover, making the moves for both Spoiler and Duplicator is a unique experience for the student that should be integrated with formal explanations.

2.2 The Adequacy Theorem

We have been educated to compare *sets*. Elementary school has first provided us for sets of sweets, big and small. A common question then was: "What is bigger?" More complex issues appear later. For instance, we have been trained by comparing mathematical objects as algebraic expressions and polynomials, and we meet equality. Moreover, Euclidean geometry told us more on comparison by providing new insight into equality of figures on a plane, or solids in the 3-dimension space. But we have not been well trained in comparing *structures*. For doing this, we introduce one well known and important concept in logic: *elementary equivalence*.

Elementary equivalence is a basic notion in model theory. We refer the reader to [Doets, 1996] for general terminology. Informally, elementary equivalence is the property that two structures have when they model the same

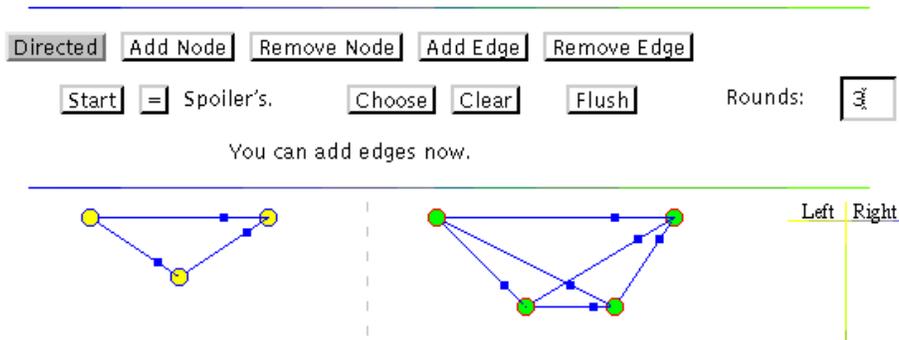


Figure 2: drawing the graphs.

set of sentences of a given language. Formally, we consider a language L and two structures that interpret L , that we call L -structures. We also stipulate the proviso that the vocabulary of L is fixed, finite, and contains only relation symbols and constants (no function symbols).

Definition 1 (Elementary equivalence) Let \mathcal{A} and \mathcal{B} be L -structures. We say that \mathcal{A} and \mathcal{B} are elementarily equivalent—in symbols: $\mathcal{A} \equiv \mathcal{B}$ —if and only if \mathcal{A} and \mathcal{B} satisfy the same sentences of L .

Definition 1 has a simple extension that accounts explicitly for the quantifier rank of the sentences of L . Recall that the *quantifier rank* $d(\varphi)$ of a formula φ is the number of nested quantifiers occurring in it (see for instance [Doets, 1996]).

Definition 2 (n -Equivalence) Let \mathcal{A} and \mathcal{B} be L -structures and n be a natural number. We say that \mathcal{A} and \mathcal{B} are n -elementarily equivalent—in symbols: $\mathcal{A} \equiv_n \mathcal{B}$ —if and only if \mathcal{A} and \mathcal{B} satisfy the same sentences of L of quantifier rank $\leq n$.

We are now ready to state the result by Ehrenfeucht (1961) which allows us to explain elementary equivalence in terms of games. We denote by $EF_n(\mathcal{A}, \mathcal{B})$ the Ehrenfeucht-Fraïssé game of length n (with n a natural number) on the structures \mathcal{A} and \mathcal{B} .

Theorem 1 (Adequacy Theorem) For L -structures \mathcal{A}, \mathcal{B} and $n \geq 0$ the following are equivalent.

- (a) Spoiler has a winning strategy in $EF_n(\mathcal{A}, \mathcal{B})$.
- (b) $\exists \varphi : d(\varphi) \leq n$ and $\mathcal{A} \models \varphi$ and not $\mathcal{B} \models \varphi$, i.e., \mathcal{A} and \mathcal{B} are not n -elementarily equivalent.

We have so far stressed the importance of EF games in the context of learning model theoretic notions. By the Adequacy Theorem we get what we hinted in the examples above, namely, the formal correlation between winning strategies and syntactic properties, such as the quantifier rank of a formula.

3 ...Via the Web

We now describe how we have brought these ideas into a Java applet—we called our application $ELgA$. We chose Java for its well known features of portability, integration in HTML documents and interface primitives. All these features allow $ELgA$ to be of potential interest for a wide audience as the WWW presently has. $ELgA$ provides a self-evaluating environment for students in the context of graphs, where both directed and undirected graphs can be played on. (Recall that a *graph* is a structure $\langle G, E \rangle$. The elements in G are called *nodes*; E is a binary relation on G whose elements are called *edges*. A graph is said *directed* if E is symmetric, and *undirected* if E is asymmetric.) Reflexive relations are also allowed: edges that start and end in the same node can be constructed.

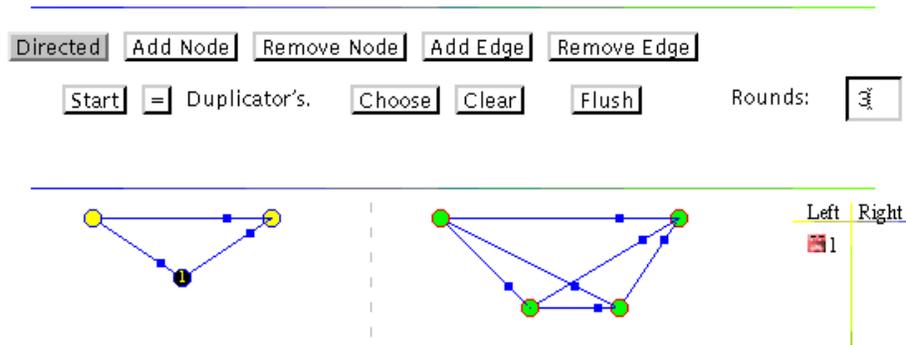


Figure 3: beginning to play.

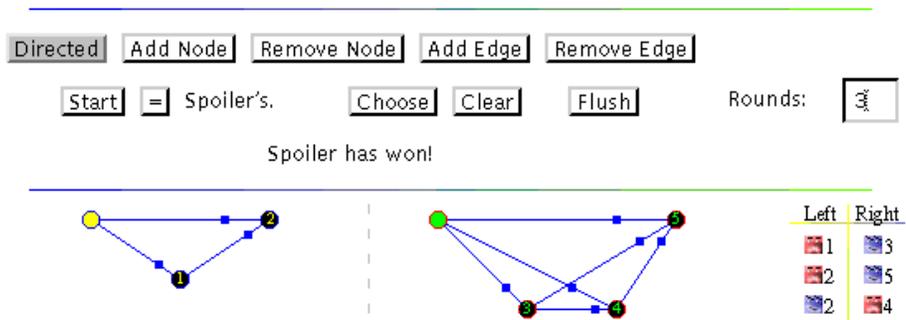


Figure 4: Spoiler has won.

3.1 Sessions

A session starts by choosing whether to have directed or undirected graphs. Then, thanks to a graphical editor it is possible to draw the two graphs.

The first move is granted to Spoiler. The node that he chooses is highlighted in black and a log of moves on the left side is updated. During a game it is possible to change the number of rounds, to decide whether playing with equality or not. A status bar always indicates who's winning, if a move is legal or not, and it helps students in their self-evaluation learning process.

Once a game is over, it is possible to play again on the same structures, eventually modifying some parameters like the number of rounds and the use of equality, or to change the game by drawing new graphs.

3.2 Example

Let us consider Example 2 above. To be able to play that game we have to input the structures. The relation $<$ is asymmetric, thus we play on directed graphs. We add 3 nodes to the left side and 4 to the right. We connect the nodes as shown in the [Fig. 2]. We are now ready to play. We set the number of rounds to 3. Spoiler can pick the central node on the left side (see [Fig. 3]). Then, Duplicator picks one of the two central nodes on the right side, and so on as in Example 2. The final situation is depicted in [Fig. 4].

4 Future Developments

The self-evaluating paradigm is not the only one from which a student can benefit. There are at least two further approaches that we plan to explore in the future.

- The possibility of playing against a *perfect opponent* should be investigated. The student that hasn't understood the game yet, should be able to play it against a perfect player. By losing the first games, a perfect

player will make the student eager to develop a new strategy and to play again, thus improving his/her understanding of the notion in the game. A client-server architecture would be the best solution, keeping the Java applet as the interface with the user on the client and implementing the perfect opponent as a running program on the server. Whenever the user makes a move, a request is sent to the central server (via a usual CGI architecture), that computes the best possible move.

Computing such move amounts to checking for partial isomorphisms between graphs. A heuristic should be devised if the player to automate is *Spoiler*. In fact, if the structures in the game are not isomorphic, *Spoiler* should pick his elements in such a way that he wins in the least possible amount of moves. On the other hand, it is easy for *Duplicator* to win in the case there is an isomorphism. He has to find the isomorphism and always play isomorphic elements.

The problem of computing moves is exponential in the dimension of the graphs on game. However, given the low number of nodes in each graph, the system would still give an output in a very short time. The server computing the moves for the automatic player could be easily implemented in Prolog or, to increase performance, in C++. The student would have the possibility to play against a perfect player and to switch sides while playing, or to look at the same game as being played between perfect players.

- Another paradigm to explore is that of having the student play against a *human expert*, say, a teacher. The teacher does not necessarily have to play the best move, and he/she can decide what to do in order to let the student learn. A space for written comments should then be available. The teacher could comment each move and explain what is going on to the student.

The implementation of these features could be obtained by the Java capabilities for establishing TCP/IP connections. The connection could be either one-one (teacher-student) or one-many (teacher-class).

Acknowledgments

The idea of implementing Ehrenfeucht-Fraïssé games came up while the second author was attending Johan van Benthem's course *Logic and Games*. We are grateful for his suggestions and the time spent to discuss the topic. We thank Johan van Benthem and Fausto Giunchiglia for useful feedback on previous versions of this article. This article was written while the first author was visiting the ILLC, Amsterdam. He thanks ILLC for his fruitful visit. The research of the second author is partially funded by CNR, borsa-203.07.27, 9/4/97.

References

- [Abramsky, 1997] Abramsky, S. (1997). Semantics of Interaction. In *Lectures notes of Joint Newton Institute CLiCS II Summer School on Semantics and Logics of Computation*. Cambridge University Press. <http://www.dcs.ed.ac.uk/home/samson/coursenotes.ps.gz>.
- [Doets, 1996] Doets, K. (1996). *Basic Model Theory*. Studies in Logic, Language and Information. CSLI Publications & FoLLI, Stanford, CA.
- [Ebbinghaus and Flum, 1995] Ebbinghaus, H.-D. and Flum, J. (1995). *Finite Model Theory*. Springer.
- [Hintikka and Sandu, 1997] Hintikka, J. and Sandu, G. (1997). Game-theoretical semantics. In van Benthem, J. and ter Meulen, A., editors, *Handbook of Logic and Language*, pages 361–410. Elsevier Science B.V., Amsterdam.
- [Hodges, 1997] Hodges, W. (1997). *A Shorter Model Theory*. Cambridge University Press, Cambridge.
- [Hodges, 1998] Hodges, W. (1998). Invitation to logical games. Lecture notes. Mathematics Department, Queen Mary's College, London.
- [van Benthem, 1998] van Benthem, J. (1998). Logic and games. Lecture notes for a graduate course. ILLC, UvA.