Introductory note

This guide to the game **Luì** (`www.oiler.education/lui`) corresponds to Chapter 5 of the work below.

Bernardi, L. (2024). *Logic education: Playing with true and false* (Doctoral dissertation). Aix-Marseille Université & Università Roma Tre.

In addition to exploring the game rules, the guide also provides a broader theoretical framework, briefly touching on the topic of dialogical logic and offering some didactic observations.

Before going through this guide, we suggest learning to play the online games *Zermelo Game* (available for free here: `en.oiler.education/zermelo`) and *Bul Game* (available for free here: `en.oiler.education/bul`).

5. Real-World Playability and Online Software Implementation

5.1.1. The Statements

In mathematics, we aim to establish general facts, which can help us *predict* (from latin *praedicĕre*, to say beforehand), that is, to know in advance certain behaviors, effects, situations. Knowing that a single object in a set satisfies a certain property is sometimes useful in some contexts, but knowing that all objects in a set satisfy a property holds greater value, because it allows us to make predictions about objects of that kind before practical verification. Although, in the following discussion, we will limit ourselves exclusively to mathematical examples, the reader can seek examples in all branches of scientific knowledge.

Let us start by considering an extremely simple statement: "each element has a square root". In other words, using logical formalism, $\forall x\, \exists y\, (y \times y = x)$. This statement is true in some environments (like complex numbers) and false in others (like integer or real numbers).

More generally, it can be discovered that—among the objects of a set—only some particular objects satisfy a certain property. In other words, one might find that a certain property is true only for those objects in a set that have specific characteristics. For example, in the context of real numbers, it is not true that every number is smaller than its square. But if we limit ourselves to numbers greater than 1, then it becomes true. In logical formalism, even though it's false that $\forall x(x < x^2)$, it is true that $\forall x(x > 1 \to x < x^2)$. As often happens, we note that the hypothesis is a sufficient condition but not a necessary one.

Generally speaking, it's quite intuitive to state a scientific law in the following manner: $\forall x(H_1(x) \wedge H_2(x) \wedge \ldots \wedge H_n(x) \to C(x))$. Put simply, if an object x from a specific set satisfies *all* the conditions H_1, ..., H_n, then it also satisfies the conclusion C. To streamline the notation, we'll replace the conjunction symbol $\wedge$ with a comma, leading to the form $\forall x(H_1(x), \ldots, H_n(x) \to C(x))$.

5.1.2. Debating on a Statement

It's now interesting to see how a statement can undergo critical analysis. For the sake of simplicity, let us imagine a conversation between two individuals with opposing views on the validity of the statement: a dialogical analysis of a statement clearly occurs in any research context, even within a single individual's thought process. How does one argue that a particular statement of the form $\forall x(H_1(x), \ldots, H_n(x) \to C(x))$ is false?

Let us envision a brief conversation between two individuals about a very simple statement.

P: "Did you know that all polygons have at least one obtuse angle?"

O: "That's not true! Look, this triangle has all acute angles."

P: "You're right, but if the polygon has at least 5 sides, then it's true!"

In this conversation, the critical analysis of the statement led to the introduction of a new hypothesis H about the polygon to make the statement true.

It's worth noting that the two interlocutors seem to have a mutual understanding of what constitutes a polygon—for instance, excluding intertwined polygons—and of the definition of an obtuse angle. For any statement to be subjected to analysis, there must be a shared foundational knowledge between the interlocutors. This includes a common language, a set of shared true statements—denoted as $\mathbb{T}$ (which could potentially be empty)—and a set of shared false statements, henceforth referred to as $\perp\!\!\!\perp$. Although $\perp\!\!\!\perp$ can also be empty, it's safe to assume that both players agree on the fact that false is indeed false, *i.e.*, $\perp \in \perp\!\!\!\perp$. Moving forward, it's important to underline that stating $A \to \perp$ is equivalent to stating $\neg A$, meaning claiming that A is false.

Generalizing from the previous example, if one wishes to argue that a statement $\forall x(H_1(x), \ldots, H_n(x) \to C(x))$ is false, they need to present a *counterexample*. Specifically, they must identify a particular object a for which $H_1(a)$, ..., $H_n(a)$ all hold true, and yet, $C(a)$ is false. In more formal terms, one must find a witness for the following formula: $\exists x(H_1(x) \wedge \ldots \wedge H_n(x) \wedge \neg C(x))$.

Let's delve into another example, a slightly more intricate conversation, which will pave the way for generalizing our earlier analysis.

P: "Did you know that all functions defined over a bounded interval have a local maximum?"

O: "No, that's not correct! The function $f(x) = \frac{1}{x}$ doesn't have a maximum in the interval $(0,1)$."

P: "Actually, $f(x) = \frac{1}{x}$ isn't defined on $(0,1)$."

O: "Yes, it is. At which point do you think it's undefined?"

P: "At 0!"

O: "Look, 0 isn't part of the interval $(0,1)$."

P: "Right..."

The same conversation, expressed in logical formalism, goes as follows:

P: "$\forall f \forall I(H_1(f,I), H_2(I) \to C(f,I))$"

Where H_1 stands for the condition that f is defined over I, H_2 indicates that the interval is bounded, and C means that f has a local maximum within the interval I.

O: "$(\forall f \forall I(H_1(f,I), H_2(I) \to C(f,I))) \to \bot$. In fact, $H_1(\frac{1}{x},(0,1)) \wedge H_2((0,1)) \wedge \neg C(\frac{1}{x},(0,1))$"

P: "$H_1(\frac{1}{x},(0,1)) \to \bot$. Indeed $\frac{1}{0}$ is not defined."

O: "$0 \notin (0,1)$"

P reflects and returns to **O** with a statement refined with a new hypothesis.

P: "Did you know that all functions defined over a closed and bounded interval have a local maximum?"

O: "That's still not right! Consider the parabola $f(x) = 1 - x^2$ and, at its vertex, redefine the function to be 0."

P: "The parabola you're describing isn't defined over a closed interval!"

O: "Yes, it is. Just define it over the interval $[-1,1]$."

P: "You're right..."

The revised statement introduced by **P** is $\forall f \forall I(H_1(f,I), H_2(I), H_3(I) \to C(f,I))$, where H_3 represents the property of the interval being closed.

P reflects further and returns to **O** with an even more refined statement.

P: "Did you know that all continuous functions defined over a closed and bounded interval have a local maximum?"

The statement is $\forall f \forall I(H_1(f,I), H_2(I), H_3(I), H_4(f,I) \to C(f,I))$ where H_4 express the propriety of a function being continuous on a domain.

We now observe that, in the preceding example, the hypotheses H may have a layered nature. Generally speaking, the hypotheses H can mirror the form of a statement seen earlier: $H = \forall y(G_1(y) \wedge \ldots \wedge G_m(y) \to G_0(y))$. By adopting this approach, we can expand upon the dialogue rules previously discussed: once **O** asserts that, with a certain witness t, $H_1(t) \wedge \ldots \wedge H_n(t)$ holds true, **P** can counter by claiming that a specific $H_i(t) = \forall y(G_1(y) \wedge \ldots \wedge G_m(y) \to G_0(y)$ is actually false. That is to say, for a particular u, while $G_1(u) \ldots G_n(u)$ all hold true, $G_0(u)$ turns out to be false, thereby continuing the discussion.

In our presented case, the conclusion C also has a layered nature. However, as presented in Chapter 1, one can always rewrite the formula to ensure that C is of a simpler nature, in a sense directly verifiable, shifting all the complexity to the hypotheses.

This kind of dynamic discussion seems to aptly simulate the process of scientific discovery. One might initially conjecture $\forall x C(x)$. However, upon observation and contemplation, it's revealed that not all x satisfy the property C, but only those for which the hypotheses H_1, ... H_n hold, leading to the formulation of a theorem. It is interesting to note that the word "theorem" originates from the Greek *theorema*, meaning 'contemplation', which in turn derives from *theoréo*, 'I see', 'I observe'.

In concluding this section, we emphasize that the scientific process and progress have three foundational characteristics: firstly, creating new hypotheses to be tested out; secondly, testing a hypothesis by challenging oneself, others, or reality through experiments; and lastly, developing theories to categorize and interpret the knowledge. Not only does the game $\mathbf{T}\mathscr{UVA}$ provide an environment to test one's hypotheses, but the set **T** captures the idea of adding new facts to shared knowledge, as already discussed at the beginning of chapter 2.

5.2. Online Implementation: the λuì Software

In this section, we present an educational transposition of the game $\mathbf{T}\mathscr{UVA}$, exploring its practical usability in a learning context. As we will see at the end of the chapter, the software could also be valuable as a proof-search program.

In presenting the software, we consider mathematical environments that on one hand are relevant in educational practice, and on the other hold distinct logical value, even from a historical perspective. Besides "pure" contexts of propositional logic and first-order logic, the online implementation allows playing also in the environments of natural numbers, mathematical analysis, and Euclidean geometry. Specifically, PA (Peano Arithmetic) is of fundamental importance in logic for various reasons including the incompleteness theorems, and for obvious reasons in education. In this regard, we note that PA, with the successor function, reflects the intuitive idea of a numerical system that is built up gradually by adding ever larger numbers. Moreover, it is notably the first area where students encounter nested quantifiers and serves as an environment for studying constructivism. Euclidean geometry (where it is still part of the curriculum!) serves as a fundamental setting for learning the logical structure of statements, with particular emphasis on implication. Historically, it has epitomized logical rigor within mathematics, to such an extent that, in 1821, A. L. Cauchy began his *Cours d'Analyse de l'École Royale Polytechnique* by stating his intention to endow the methods of analysis with "all the rigor that is demanded in geometry". Furthermore, in any logic course that discuss axioms, Euclid's axioms are invariably explored.

In other words, this chapter aims to serve as a bridge between the first four chapters, which are exclusively logical in their nature, and the last four, which are predominantly related to Mathematics Education.

Let us now delve into the online implementation of the $\mathbf{T}\mathscr{UVA}$ game, called λ**ui**. The game was developed with the essential contributions of Mattia Sanchioni (for managing the code logic and generally the backend) and Luca di Pietro Martinelli (for managing the website where the code is run and generally the frontend, handling UI and UX).

The game features two players challenging each other regarding the truth of F, a

Figure 5.1.: Homepage of λui.

formula written in Krivine's normal form. The *Proponent*, abbreviated as **P**, believes F to be true, while the Opponent, abbreviated as **O**, believes it to be false. The game follows dialogue rules that formalize the insights given in the previous section.

The software is split into a backend, accessible at `www.galua.cc`, which manages the game's logic and data storage, and a frontend available at `www.oiler.education/lui`, where the game is intended to be played, which exclusively handles the visualization of the game's dynamics[1]. The name "Luì" is inspired by the French pronunciation of Jean-Louis Krivine, the designer of $\mathscr{U}\mathscr{V}\mathscr{A}$ game (Krivine and Legrandgérard 2007).

At `www.oiler.education/lui`, users can find and freely access the software. Here, they have the choice of playing in either propositional logic or first-order logic (Figure 5.1). Within the domain of first-order logic, there are four available theories to select from: **pure logic** (where no specificy theory **T** is involved), **Giuseppe Peano** (PA arithmetic), **Auoquamel** (analysis), and **Alfred Tarski** (Euclidian geometry). As we will later discuss, adjustments have been made on these theories to ensure their playability.

Once users have picked their desired theory, they proceed to select a formula of that theory and start the game by clicking on 'START'. As can be seen from Figure 5.1, a 'User vs PC' mode is also planned for the future. For now, the game is played between two real players (*Proponent* and *Opponent*) who play on the same device.

During the game, the three sets $\mathscr{U}$, $\mathscr{V}$, and $\mathscr{A}$ are referred to as $\mathbf{O}_\top$, $\mathbf{P}_\top$, and $\perp\!\!\!\perp$ respectively, for easier comprehension on their status. Indeed, $\mathbf{O}_\top$ represents what is true for the Opponent, $\mathbf{P}_\top$ what is true for the Proponent, while $\perp\!\!\!\perp$ represents what is false for both players. When a theory is present, it is denoted by $\top\!\!\!\top$, underlining its symmetric and opposite relationship with $\perp\!\!\!\perp$.

In fact, there is a strong duality between the sets $\top\!\!\!\top$ and $\perp\!\!\!\perp$: the former contains

[1]The frontend and backend communicate via REST API: the frontend initiates a call to an endpoint on `www.galua.cc`, passing all required parameters, and the server responds with the requested data.

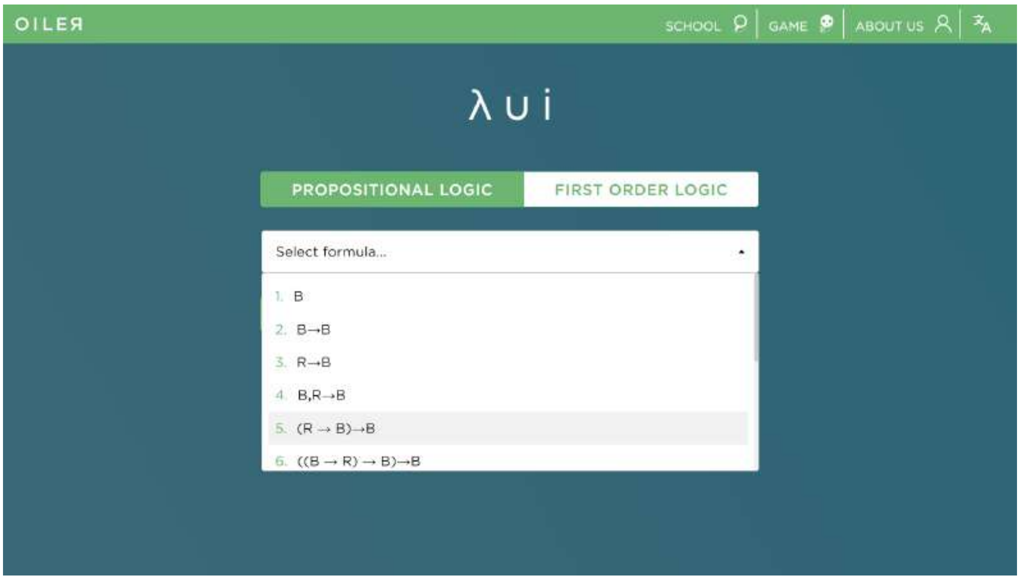

Figure 5.2.: In Propositional Logic, users have a choice among 13 distinct formulas.

formulas that are true for both players, and the latter contains formulas that are false for both players. The union of these two sets constitutes what can be referred to as *common knowledge*. However, as we will see in more detail when presenting the rules of the game, this shared knowledge has a peculiarity, and the dualism between ⫪ and ⫫ becomes even more evident: the formulas in the set ⫪ can only be invoked by **P** (we might say removed from ⫪, barring contractions), while **O** is the only one who can add formulas to ⫫. The ⫪ remains constant during the game, whereas the ⫫ is not. Indeed, this asymmetry makes sense: the Opponent, in their attempt to deny the formula F, has no interest in making concessions of truths. Symmetrically, the Proponent has no interest in conceding false formulas.

To facilitate reading, if during the game the user hover the cursor over a formula without clicking it, the hypotheses of a formula are marked in orange, while the conclusion is in dark blue. The top-level quantifiers ∀ and the main implication → are in black. As an example, a formula is written as $\forall x(F_1(x),\ldots,F_n(x) \to A(x))$.

For every game modality (*i.e.*, every theory), we will provide the specific rules of that modality, evidently adapted from the general rules outlined in Chapter 1.

5.2.1. Propositional Logic

The game is played between **P** and **O** on a propositional formula F; in Propositional Logic, users have a choice among 13 distinct formulas (Figure 5.2). The formulas aim to provide a progressive and meaningful approach to propositional logic.

The game initializes with $\mathbf{O}_\top = \{F \to \bot\}$ (*i.e.*, **O** believes F to be false), $\mathbf{P}_\top = \{F\}$ (*i.e.*, **P** believes F to be true), and ⫫ $= \{\bot\}$ (*i.e.*, both players concede that ⊥ is false). The game starts with **O** playing first, and the turns alternate thereafter.

- **O** plays by choosing a formula $F \in \mathbf{P}_\top$. They add all the premises $F_1,\ldots,F_n$ of F

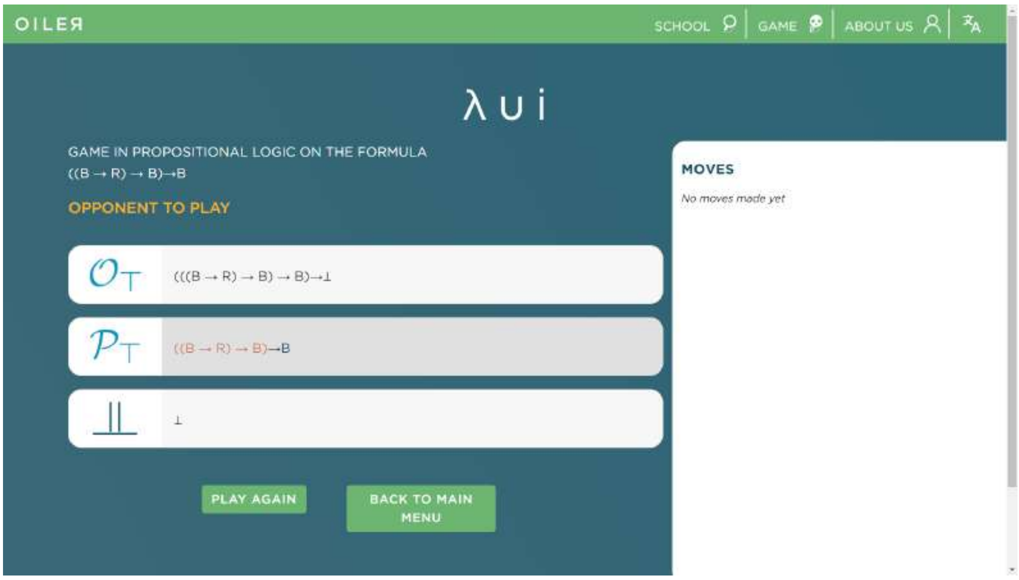

Figure 5.3.: The Opponent's opening move, where they pick $((B \to R) \to B) \to B$.

to $\mathbf{O}_\top$ and the conclusion F_0 to $\perp\!\!\!\perp$. In particular, if $\mathbf{P}_\top$ is empty then $\mathbf{O}$ cannot move.

- $\mathbf{P}$ plays by choosing a formula $F \in \mathbf{O}_\top$ such that the conclusion F_0 is in $\perp\!\!\!\perp$. They replace the set $\mathbf{P}_\top$ with $\{F_1, \ldots, F_n\}$.

If the play is finite, which is equivalent to say that **O** can no longer make a move, then **P** is the winner; otherwise, **O** wins. Clearly, since **O** wins if and only if the play is infinite, the message "Opponent wins" does not exist.

As an example, let us analyze a match on Pierce's formula $((B \to R) \to B) \to B$.

The game begins with the Opponent's move (see 5.3), where they select the only formula they can from the set $\mathbf{P}_\top$, namely $((B \to R) \to B) \to B$. They assert that B is false, placing it into $\perp\!\!\!\perp$, and that $((B \to R) \to B)$ is true, moving it to $\mathbf{O}_\top$.

The Proponent now has two options (see 5.4): either restart the game by choosing $(((B \to R) \to B) \to B) \to \perp$ (which is the negation of Pierce's formula), or selecting $((B \to R) \to B)$. The move is valid because $B \in \perp\!\!\!\perp$. Clearly, continuously restarting the game in the long run isn't a favorable strategy, so **P** opts for $((B \to R) \to B)$.

The Opponent, still following a predetermined path, proceeds by adding $R \in \perp\!\!\!\perp$ and $B \in \mathbf{O}_{\top\!\top}$ (see 5.5).

The Proponent wins since B belongs to both $\mathbf{O}_\top$ and $\perp\!\!\!\perp$ (see 5.6 and 5.7).

5.2.2. First Order Logic: Pure Logic

The game is played between **P** and **O** on a first-order formula F. In Pure Logic users have a choice among 13 distinct formulas, as shown in Figure 5.8. Similarly to Propositional Logic environment, the formulas strive to follow a progressive development of skills: starting from very simple formulas, moving through the Drinker Paradox

Figure 5.4.: Proponent's first move.

(formula number 7), to binary predicates[2]. The game initializes with $\mathbf{O}_\top = \{F \to \bot\}$, $\mathbf{P}_\top = \{F\}$, and $\perp\!\!\!\perp = \{\bot\}$. The game begins with **O** playing first, and the turns alternate.

- **O** plays by choosing a formula $F \in \mathbf{P}_\top$ and closed terms $\vec{b}$ to be substituted to variables $\vec{x}$ of the top-level quantifiers of F. They add $F(\vec{b})_1, \dots, F(\vec{b})_n$ to $\mathbf{O}_\top$ and $F(\vec{b})_0$ to $\perp\!\!\!\perp$. In particular, if $\mathbf{P}_\top$ is empty then **O** cannot move.
- **P** plays by choosing a formula $F \in \mathbf{O}_\top$ and closed terms $\vec{b}$ such that $F(\vec{b})_0 \in \perp\!\!\!\perp$. They replace the set $\mathbf{P}_\top$ with $\{F(\vec{b})_1, \dots, F(\vec{b})_n\}$.

We note that a closed term—in the Pure Logic mode—is simply a letter (*e.g.*, a, b, ...); at each turn, the player can choose whether to play a letter that has been played previously in the game or a new letter, referred to as a *fresh constant*[3]. If the play is finite, which is to say that **O** can no longer make a move, then **P** is the winner; otherwise, **O** wins. Clearly, in this instance as well, since **O** wins if and only if the play is infinite, the message "Opponent wins" does not exists.

Let us see a play on the formula number 5, which is $\forall x((\forall y(G(y) \to \bot) \to \bot) \to G(x))$. On the right, we can see the set of moves: each move is presented as $(F, \vec{b})$ where F is the picked formula and $\vec{b}$ are the picked closed terms.

The first to move is **O**. They select the only formula in $\mathbf{P}_\top$ and choose the closed terms to replace x. Since no closed term has been played yet, the only move they can make is to play a fresh constant, namely the first letter a.

[2]For completeness, we inform the reader that the formula $\forall x(\forall y(A(x,y) \to \bot), (\forall z(\forall w A(w,z) \to \bot) \to \bot) \to \bot)$ corresponds to $\exists y \forall x A(x,y) \to \forall x \exists y A(x,y)$ while the formula $\forall x(\forall y A(y,x) \to \bot), \forall z(\forall w(A(z,w)\bot) \to \bot) \to \bot)$ correspond to $\forall x \exists y A(x,y) \to \exists y \forall x A(x,y)$. Clearly, the first one is false and the second is true.

[3]In other words, during their move, the two players can choose the witness from among those played up to that point, or a new one.

Figure 5.5.: Opponent's move.

The conclusion $G(a)$ is subsequently added to $\perp\!\!\!\perp$, and the only premise is added to $\mathbf{O}_\top$. It's now to **P** to play. They have the choice of either playing $F \to \perp$ (restarting the game) or playing the formula, just added by $\mathbf{O}_\top$, $\forall y(G(y) \to \perp) \to \perp$: they're allowed to make this move since $\perp \in \perp\!\!\!\perp$.

O is now obligated to select the only formula present in $\mathbf{P}_\top$. However, they do have the choice to replace y with either a previously played closed term (namely a) or introduce a new one. This decision is crucial.

If **O** makes the unfortunate choice of playing a, then **P** will win in the subsequent turn. On the other hand, by choosing a new letter (and doing so every time the opportunity arises), the Opponent manages to perpetually continue the game, thus winning. It's worth noting here—as extensively discussed earlier—that the formula is false because a winning strategy exists for the Opponent. However, if the Opponent makes erroneous choices, they can still lose.

5.2.3. Giuseppe Peano

5.2.3.1. PA Theory

Peano's theory is an axiomatic system that aims to describe the set of natural numbers with elementary operations. It was introduced by the Italian mathematician Giuseppe Peano in 1889. The updated first-order theory is today referred to as Peano Arithmetic, or more simply PA. The language used in PA includes symbols for functions $0, s, +$, and $\times$ where s denotes the function that assigns the successor to every natural number. The only relation symbol is $=$. The axioms, in addition to the three standard ones for equality[4], are as follows:

[4]Hereafter, we will refer to these axioms as EQ1 (reflexivity), EQ2 (symmetry), and EQ3 (transitivity).

Figure 5.6.: Proponent wins by picking B from $\mathbf{O}_\top$.

- (PA1) $\forall x \neg(s(x) = 0)$
- (PA2) $\forall x \forall y (s(x) = s(y) \rightarrow x = y)$
- (PA3) $\forall x (x + 0 = x)$
- (PA4) $\forall x \forall y (x + s(y) = s(x + y))$
- (PA5) $\forall x (x \times 0 = 0)$
- (PA6) $\forall x \forall y (x \times s(y) = (x \times y) + x)$
- (PA7) the axiom schema for the Induction Principle

5.2.3.2. On the theory used in the game

First of all, let us notice that the theory we just introduced has closed terms: consequently, it would be possible to effectively play it in a $\mathbf{T}\mathscr{UVA}$ game. However, doing so would be exceedingly cumbersome and tedious, even for a logic enthusiast. If we hope to achieve a truly playable game, one requirement we cannot avoid is that the two players should be able to directly input, via keyboard, natural numbers as closed terms. This requirement makes the function s superfluous, as it can be emulated by the unary function $x + 1$[5]. The need to enter natural numbers brings with it the introduction of another axiom schema: for each n, we have $\overline{n} + 1 = \overline{n+1}$, where n is a constant symbol, and $\overline{n}$ is the interpretation of the constant in the model. However, if a player were to justify the status of the number n based on these axioms every time n it's played, the game would still be overwhelmingly tedious.

[5]Consequently, the axioms PA1 and PA2 are rewritten using the function +.

Figure 5.7.: Victory screen for **P**.

In the same fashion, justifying every single operation based on the axioms of addition and multiplication would be exceedingly long. For this reason, we decided to let the program handle every operation and the status of the natural numbers by itself. Specifically, every time a closed term appears in the game, it is replaced with the corresponding natural number. In other words, by doing so, the program autonomously manages the axioms PA3, PA4, PA5, PA6, which can thus be dropped from $\mathbb{T}$.

We have also added, as symbols for predicates, the usual predicates in elementary arithmetic practice: $<$, $>$, `EVEN`, `ODD`, `DIVIDE`, `PRIME`. Every predicate P thus added, which we will call a *derived* predicate, is inserted into the theory $\mathbb{T}$ with the axiom

$$P(x) \iff F_P(x)$$

where $F_P(x)$ is a formula written in the language without P. We note that clearly `PRIME` depends on `DIVIDE`: before being able to define `PRIME` as a derived predicate, it is advisable to enrich the language with the symbol for the `DIVIDE` predicate. Since the connective $\iff$ is not part of the language, for each predicate P, formulas P_1 and P_2 have been added to the axioms $\mathbb{T}$, one for each side of the implication.

The symbol for the `DIVIDE` predicate is expressed as $\lhd$ even though standard mathematical practice uses the symbol $|$. This was done because the symbol $\lhd$ captures much more effectively than $|$ the fact that the binary predicate `DIVIDE` induces an order relation over $\mathbb{N}$. We believe that the order relation thus induced is an order relation that is worth delving into at an educational level, for two distinct reasons: firstly, unlike the classic $<$, it is a non total order relation and, additionally, it admits both a minimum, which is 1, and a maximum, which is 0. This is of interest in the definition of the least common multiple: in elementary definitions, "least" refers to the $<$ order; it is therefore necessary to exclude 0 in the definition, limiting oneself to

Figure 5.8.: In Pure Logic (FOL), users have a choice among 13 distinct formulas.

positive numbers.

5.2.3.3. The Game

In the GIUSEPPE PEANO game mode, the user chooses which formula to play from 26 options and whether to play in *SHORTCUT* or *STANDARD MODE* (see Figure 5.13), a distinction that will be elaborated upon later.

The 26 formulas aim to capture some important aspects of number theory. The first three formulas, one true and two false, are used exclusively to familiarize players with the dynamics of the game. Subsequently, simple yet fundamental situations involving even and odd numbers—often overlooked in traditional teaching—are addressed. Following this, Fermat's Last Theorem is proposed, limited to the cases $n = 2$ (*i.e.*, the search for Pythagorean triples) and $n = 3$, where the search for triples can be intriguing and stimulating, although fruitless as demonstrated by L. Euler.

Formula 12[6] is connected to the search for fractions that approximate the square root of 2. Indeed, the term $\frac{1}{x^2}$ becomes increasingly irrelevant as x and y grow. The possible approximation has been known since ancient times (Maracchia 2005), and the numbers xs that satisfy the relation are called *lateral numbers*, while the ys are *diagonal numbers*. It should be noted that to find all lateral and diagonal numbers, one should also study the formula $\forall x, y(2 \times x^2 = y^2 + 1 \rightarrow \bot)$.

Formulas 13, 14, and 15 allow for an in-depth exploration of divisibility and, more generally, proofs in PA. For example, with formula 13, although **P** can always win easily, it's not immediately obvious why this is the case. From 16 to 20, the focus is exclusively on the definition of the PRIME predicate, a foundational concept in number theory. From 21 to 26, typical number theory formulas involving primality are proposed. It

[6] $\forall x, y(2 \times x^2 + 1 = y^2 \rightarrow \bot)$

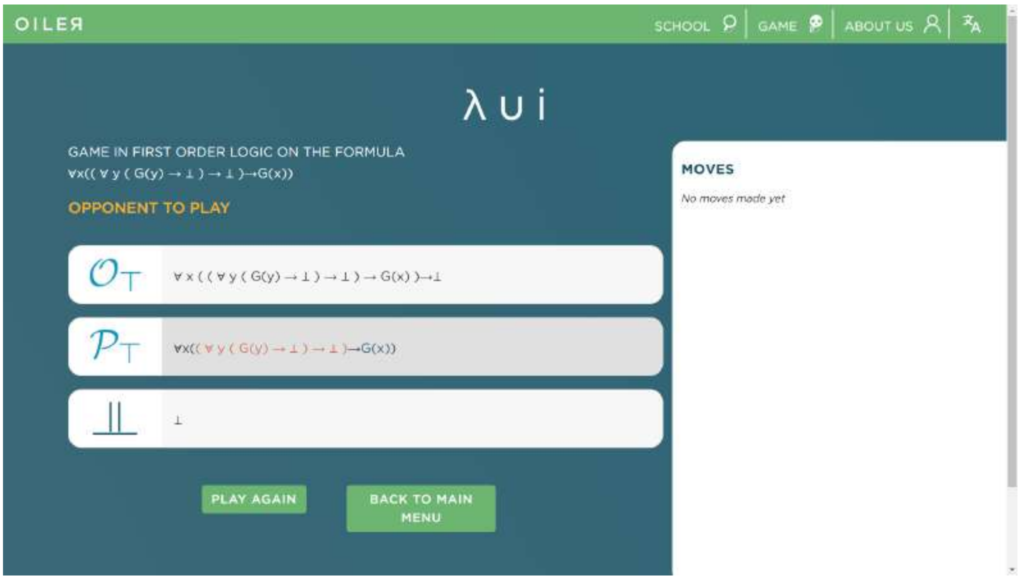

Figure 5.9.: Initial position: **O** plays the only available formula in $\mathbf{P}_\top$.

was decided to conclude with the Goldbach Conjecture[7], as easy to understand as it is difficult to prove. In fact, the conjecture as stated is *false*: we did not specify the condition that the even number must be greater than 2. It will be interesting to see if the Opponent can win by exploiting this gap.

As already mentioned, when players choose to play in GIUSEPPE PEANO mode, they can decide whether to play in *STANDARD MODE* or *SHORTCUT MODE*.

STANDARD MODE

The game is played between **P** and **O** using a first-order formula F written in PA. The game is initialized as usual.

- **O** plays by choosing a formula $F \in \mathbf{P}_\top$ and natural numbers $\vec{n}$ to be substituted to variables $\vec{x}$ of the top-level quantifiers of F. They add $F(\vec{n})_1, \ldots, F(\vec{n})_n$ to $\mathbf{O}_\top$ and $F(\vec{n})_0$ to $\perp\!\!\!\perp$. In particular, if $\mathbf{P}_\top$ is empty then **O** cannot move.
- **P** plays by choosing a formula $F \in \mathbf{O}_\top$ and natural numbers $\vec{n}$ such that $F(\vec{n})_0 \in \perp\!\!\!\perp$. They replace the set $\mathbf{P}\top$ with $\{F(\vec{b})_1,\ldots,F(\vec{b})_n\}$.

This mode, despite its theoretical interest, is still too complex for practical play, at least initially. Therefore, in addition to all the modifications already implemented to facilitate the game, an additional rule is added in SHORTCUT mode to further simplify it.

SHORTCUT MODE

In the SHORTCUT mode, axioms PA1, PA2, EQ1, EQ2, and EQ3 are removed from the theory $\top\!\!\!\top$. However, a new rule for **P** is introduced: every time a true equality appears in $\perp\!\!\!\perp$, the Proponent has the right to click on it, winning the match. Similarly,

[7]Which, in fact, is attributed to L. Euler!

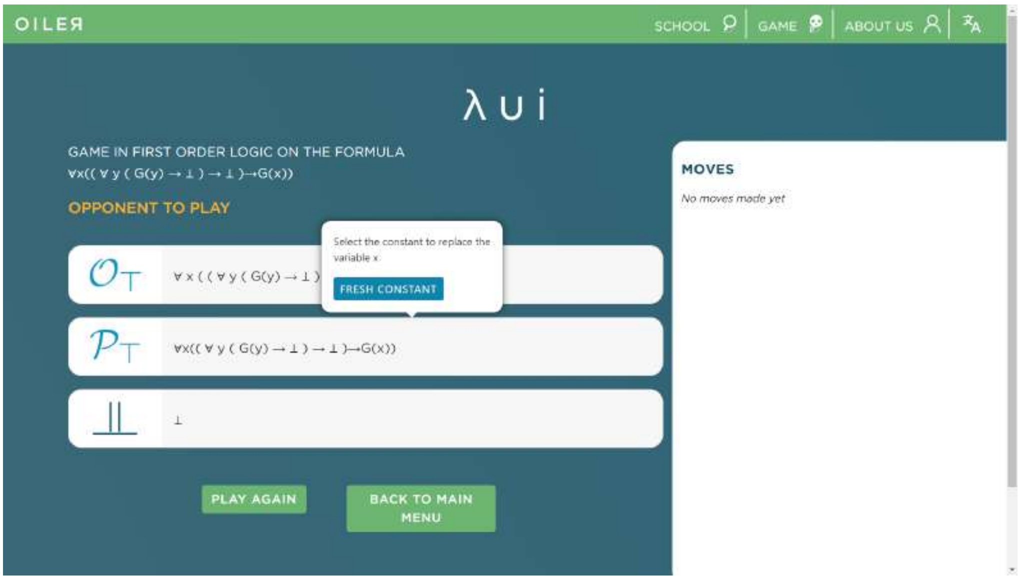

Figure 5.10.: **O** selects the only available formula and replace x with a fresh constant.

whenever a formula of the kind $\forall\vec{x}\ (F_1,\ldots,F_n \to t_1(\vec{x}) = t_2(\vec{x}))$ appears in $\mathbf{O}_\top$, the Proponent can always play it, provided that—after the substitution of variables with closed terms—$t_1 = t_2$ is false. This approach is virtually the same to including all true equalities into ⫪ and all the false ones into ⫫. This concept also echoed in Krivine's work (Krivine 2006). In other words, **P** and **O** leave the burden and honor of evaluating equalities to the computer.

More generally, considering potential future developments, the SHORTCUT rule can be implemented in reference to any predicate P, leaving it to the computer to evaluate the predicate. This is done by virtually inserting all $P(\vec{(n)})$ for which P is false into ⫫, and all $P(\vec{(n)})$ for which P is true into ⫪. In a sense, once one becomes accustomed to handling a certain predicate, a method for evaluating its truthfulness is also shared, without having to justify it each time up to basic definitions. Similarly, once the winning strategy for **P** on a formula F is found, this can be inserted into **T**, emulating the evolution of shared knowledge between **P** and **O**

Let us consider an example, in SHORTCUT mode, of a game on the formula PRIME(221). The game is initialized with **P** asserting that 221 is prime, and **O** asserting that it is not prime, as shown in Figure 5.14.

In the first move of the game, Player **O** places PRIME(221) in ⫫, see Figure 5.15.

At this point, **P** recalls the definition of a prime number, stating that if **O** claims that it is not true that 221 is prime, **O** must provide a number that divides 221 that is different from both 1 and 221 (Figure 5.16).

Now, **O** could lose if they provided a wrong witness, such as a number that does not divide 221. However, since 221 is not prime, there are correct witnesses, such as 13. At this point, Player **O** claims that 13 divides 221, but that 13 is neither equal to 1 nor to 221 (see Figure 5.17).

The game could continue with **P** recalling the definition of the `DIVIDE` predicate, to

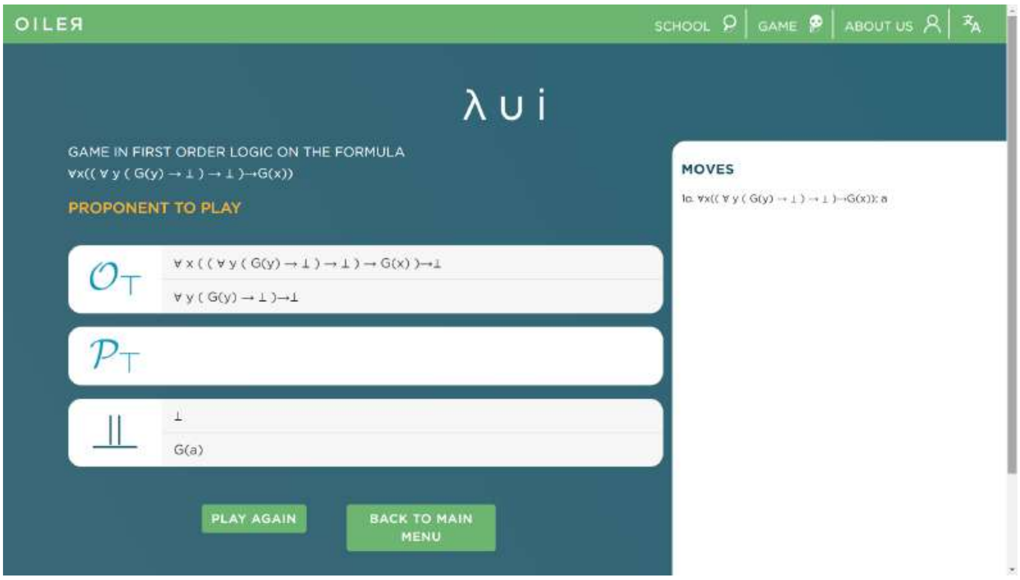

Figure 5.11.: **P** plays $\forall y(G(y) \to \bot) \to \bot$.

invite **O** to find a number k such that $13 \times k = 221$. Clearly, here too, **O** is able to identify the correct witness k. In the end, **P** can do nothing but repeat the same moves, and the game will result in an infinite loop. In other words, **O** will never fall into contradiction.

We conclude the section with a lemma that ensures the two game modes presented are equivalent.

Lemma 23. *The SHORTCUT MODE and the STANDARD MODE are equivalent up to winning strategy for* **P**.

Proof. The SHORTCUT MODE introduces two simplifications: first, whenever a true equality appears in $\perp\!\!\!\perp$, **P** can point it out and win the game; second, to play a formula $F = \forall \vec{x}\,(F_1, \dots, F_n \to t_1(\vec{x}) = t_2(\vec{x}))$ contained in $\mathbf{O}_\top$, it is sufficient that, after the instantiation of the variables $\vec{x}$, $t_1 = t_2$ turns out to be false, regardless of whether it belongs to $\perp\!\!\!\perp$ or not.

Regarding the first simplification, since sums and products are managed by the computer in both modes, it is enough to note that any true equality is of the type $n = n$ for some natural number n. Therefore, if the equality $n = n$ appears in $\perp\!\!\!\perp$, the game is easily won in both modes: in the SHORTCUT MODE, it suffices to click on the equality in question, while in the STANDARD MODE, it is sufficient to play the axiom EQ1 $= \forall x(x = x)$, choosing the constant n.

Regarding the second simplification, we need to show that—in the STANDARD MODE—it is always possible for **P** to make **O** admit a false equality. To do this, note that, since sums and products are managed by the computer, a false equality is always of the type $n = m$ with n and m natural numbers different from each other. Furthermore, thanks to the axiom EQ2 (symmetry of equality), we can always assume $n < m$. Thus, to make a false equality of one's choice appear in $\perp\!\!\!\perp$, it is sufficient to first play

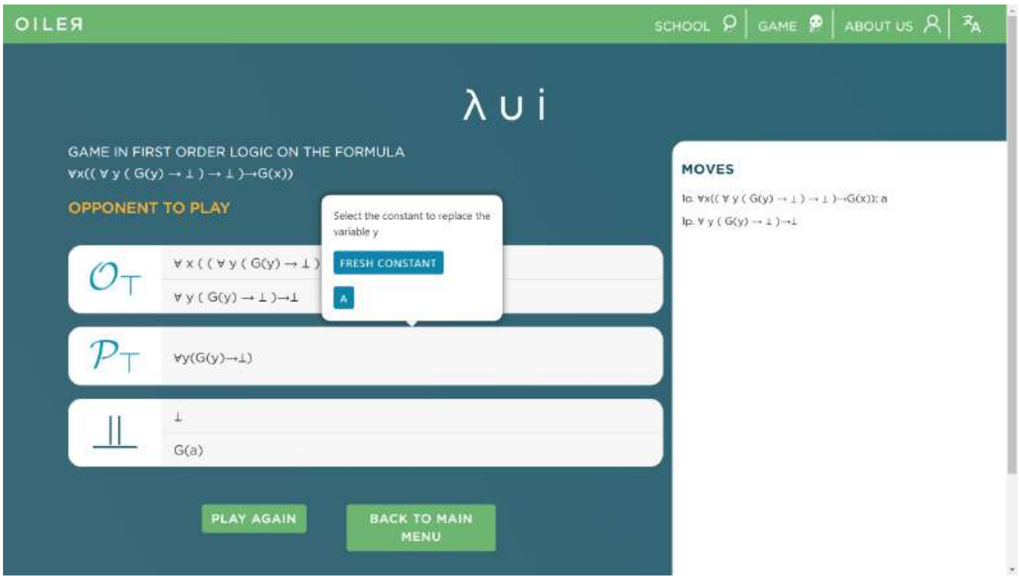

Figure 5.12.: **O** decides on the closed term.

the axiom PA1 with $k = m - n$ and then repeatedly play the axiom PA2 until the desired equality is obtained.

As can be noted, throughout the proof, the axiom EQ3 (not present in the SHORTCUT MODE) was not useful in proving the equivalence of the two modes. Indeed, EQ3 is not useful even in the STANDARD MODE and could be safely removed from $\mathbb{T}$ like the axioms PA3, PA4, PA5, and PA6. □

5.2.3.4. Differences with the Formal Game: On the Induction Principle and True Formulas in $\mathbb{N}$

The version of PA we've defined doesn't precisely mirror the theoretical game for two distinct reasons: firstly, it's not true that infinitely many constants are fresh for $\mathbb{T}$; and secondly the Induction Principle is not included in the theory $\mathbb{T}$.

Concerning the first discrepancy, players can only use numbers in the game, each described by the theory. This means that winning strategies in λuì may not correspond to proofs in PA. When discussing a game on formula F, there is a possibility that while having *for each possible play a winning strategy*, one might not have *a unique winning strategy for every possible play.*

However, we can agree that the condition "for each possible play there is a winning strategy" is a necessary condition for "there exists a winning strategy for every possible play". Once a student has for every play a winning strategy, they can be encouraged to generalize the reasoning, explaining why they are sure to win, no matter which numbers **O** will play. Specifically, by using variables in the Opponent's moves instead of constants.

Relating to the second reason why λuì doesn't perfectly mirror the theoretical game, it's worth noting that the Induction Principle (IP) is not included in $\mathbb{T}$. This does not

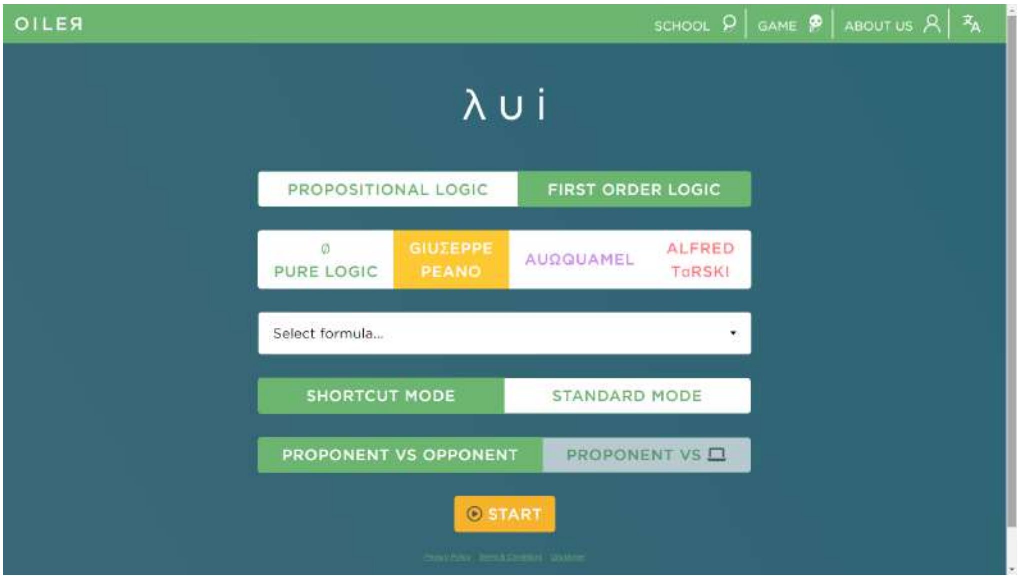

Figure 5.13.: Users set up their game within the GIUSEPPE PEANO mode.

change the potential of having a winning strategy since IP is valid in $\mathbb{N}$. What changes is that the Induction Principle provides the ability to generalize reasoning. Without it, winning strategies might not correspond to derivations. IP can indeed be presented from this perspective, namely as a tool for generalizing reasoning. We plan to add the possibility to play with the IP soon.

5.2.4. Auoquamel

5.2.4.1. Real Numbers

There are various axiomatizations for real numbers, and we find it interesting to mention two qualitatively different approaches here. In 1936, Tarski proposed an elegant second-order axiomatization, which allows for the discussion of completeness (*i.e.*, every non-empty set that is bounded above has a least upper bound), but is clearly unsuitable for our purposes due to its second-order nature. It's worth noting that, in any case, any course in mathematical analysis implicitly considers a second-order structure. On the other hand—as far as first-order is concerned—the theory of *real closed fields* is usually considered. This theory has as models all those that are elementarily equivalent to the real numbers using the standard language (*i.e.*, those models that satisfy all and only the first-order formulas satisfied by the real numbers).

More specifically, a real closed field F is a totally ordered field where every positive element of F has a square root in F, and any polynomial of odd degree with coefficients in F has at least one root in F. This theory was proven to be decidable by A. Tarski.

An example of a model of this theory is the set of algebraic numbers, which could theoretically be utilized in the $\mathbf{T}\mathscr{UVA}$ game. In this game, playing a constant means selecting a specific polynomial and specifying (with the order relation) which root

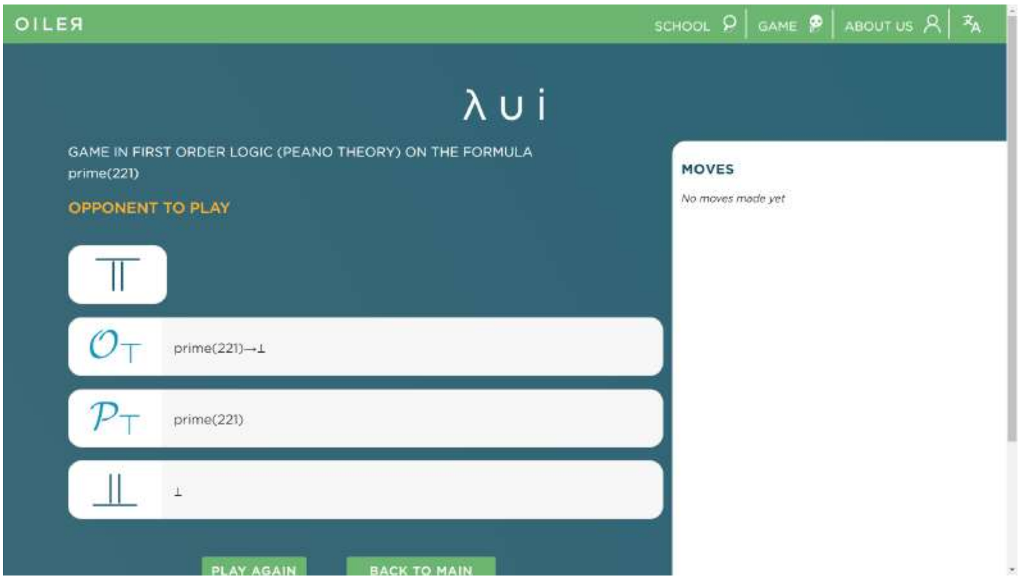

Figure 5.14.: The user sets their game within the GIUSEPPE PEANO mode.

to consider. However, despite being theoretically feasible, this approach would be impractical and artificial in a real-world setting. Additionally, it would be unsatisfactory as it would not allow for the play of commonly used constants like π or e. From a theoretical perspective, the real closed field of computable reals is more intriguing.

Computable numbers are real numbers that can be computed to any desired precision using a finite, terminating algorithm. Emile Borel introduced the concept of a computable real number in 1912, based on the intuitive notion of computability available at the time.

A real number a is considered computable if it can be approximated by a computable function

$$f : \mathbb{N} \to \mathbb{Z}$$

in the following way: for any given positive integer n, the function produces an integer $f(n)$ such that:

$$\frac{f(n)-1}{n} \leq a \leq \frac{f(n)+1}{n}.$$

The fact that computable real numbers form a field was first proved by Henry Gordon Rice in 1954 (Rice 1954).

Therefore, playing a constant in this model would mean selecting the index of the computable function. However, despite this scenario being theoretically feasible, an actual game is impossible.

5.2.4.2. On the theory used in the game

In the online game, only limited decimals are used, meaning those with finitely many digits after the decimal point in base 10. More specifically, decimals with a fixed maximum length are used. The relations in our language are >, <, and =. As for functions,

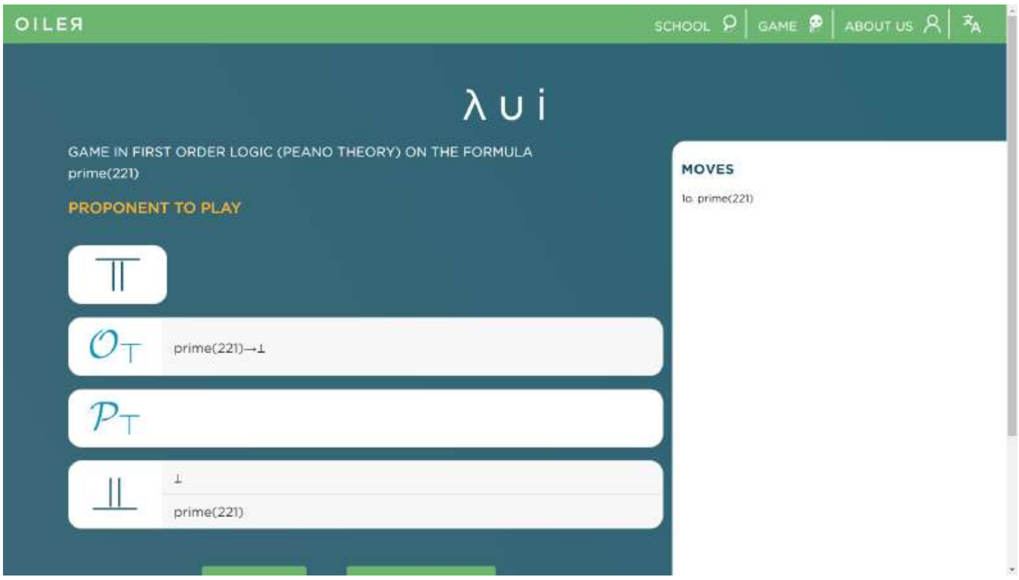

Figure 5.15.: The game is played on the formula PRIME(221).

we consider all the commonly used functions in analysis: $+$, $-$ (binary), $-$ (unary), $/$, $\times$, pow, log, sin, cos, tan, $\sqrt{x}$, $\sqrt[3]{x}$, absolute value, π, e, and ϕ. Clearly, there are formulas that are true in $\mathbb{R}$ but false when restricted to our model. Furthermore, the relations of $>$, $<$, and $=$ turn out to be decidable in our model, whereas they are only semidecidable in computable reals[8]. However, this does not impose a pedagogical limitation on our game, because the formulas from which players can choose are almost always formulas with the same truth value in both models. And when this is not the case, interesting educational insights can be drawn from the discrepancy. Furthermore, we believe that the set of limited decimal numbers is sufficient at an educational level to provide the students with the necessary intuitions for understanding real numbers. The model of limited decimal numbers captures the underlying dynamics and conveys that every real number can be approximated with reasonable accuracy by a limited decimal, an accuracy that clearly increases with the number of significant digits available.

We emphasize that the functions on closed terms are automatically calculated by the computer[9], and so is the truth value of closed relations: in other words, the SHORTCUT mode is always active for every predicate. Consequently, since there are no other predicates defined from the basic predicates, **T** turns out to be empty, and therefore not present during the game.

To conclude this section, we highlight a fact of crucial importance: unlike what

[8] In computable reals, if two numbers are different, the computation will eventually identify this difference. However, in general, it cannot determine if two reals are equal. Conversely, if $>$ and $<$ are satisfied, the computation will eventually realize it, but if they are not, it might never become aware of this.

[9] Reconstructing the logical steps necessary to justify, for example, the sum of real numbers every time would be excessively demanding and not in line with the goals of the game.

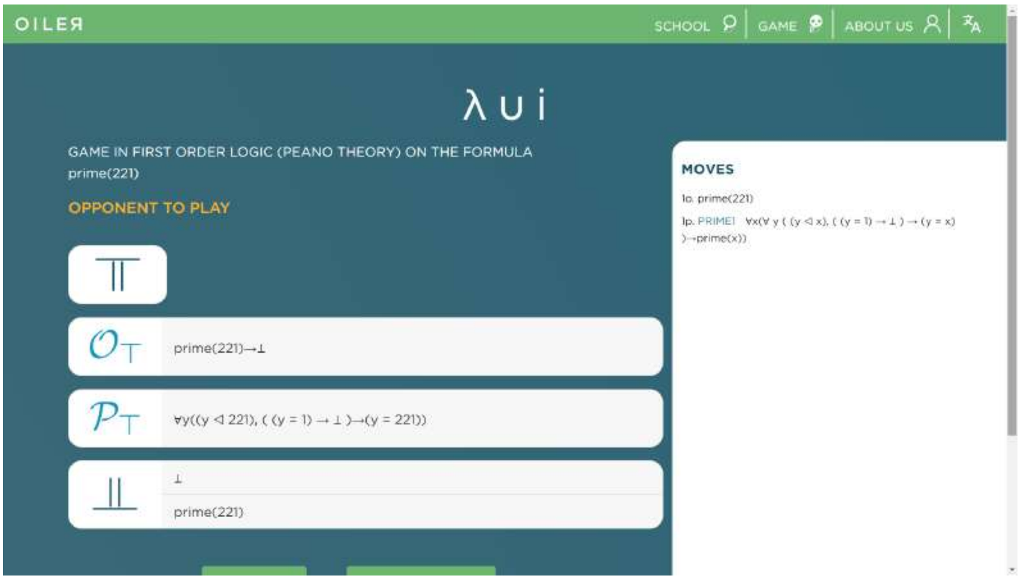

Figure 5.16.: **O** must provide a number y that satisfies the formula $\forall y(y \lhd 221,\ (y = 1 \to \bot) \to y = 221)$.

happens in PA, the domain of many functions is not the entire set of $\mathbb{R}$. The most straightforward solution is that, as soon as the computer gives a domain error, the last player who made a move is asked to change the numbers they have chosen. Unfortunately, this is not a valid solution: consider, for example, $log(x \times y)$ and suppose that at some point in the game a player chose $x = 0$. The player subsequently called to choose y cannot select any value due to the previous choice of x. In other words, although not responsible for the domain error, the second player bears its consequences.

The solution we have found is twofold: on one hand, in every playable formula, the domain is always precisely specified (by including the domain conditions as hypotheses in the formula). On the other hand, when the computer returns a domain error, the function is simply not computed, but the game continues with the unprocessed expression: the player responsible for the domain error will lose because the number they chose does not meet the pre-established domain conditions.

5.2.4.3. The Game

In the AUOQUAMEL game mode, users choose which formula to play from 21 available options. The rules of the game are identical to those of the others modalities, with the only exception being that the closed terms players can play are indeed limited decimals. The chosen formulas allow for a gradual approach to the concept of limits, with the number of quantifiers and connectives progressively increasing. The first six formulas pertain to the concept of bounded and unbounded functions. Clearly, some are true while others are not. It's interesting to note that a function f is upper-bounded if $\exists x \forall y(f(y) < x)$, while it is unbounded above if it satisfies $\forall x \exists y(f(y) > x)$, which

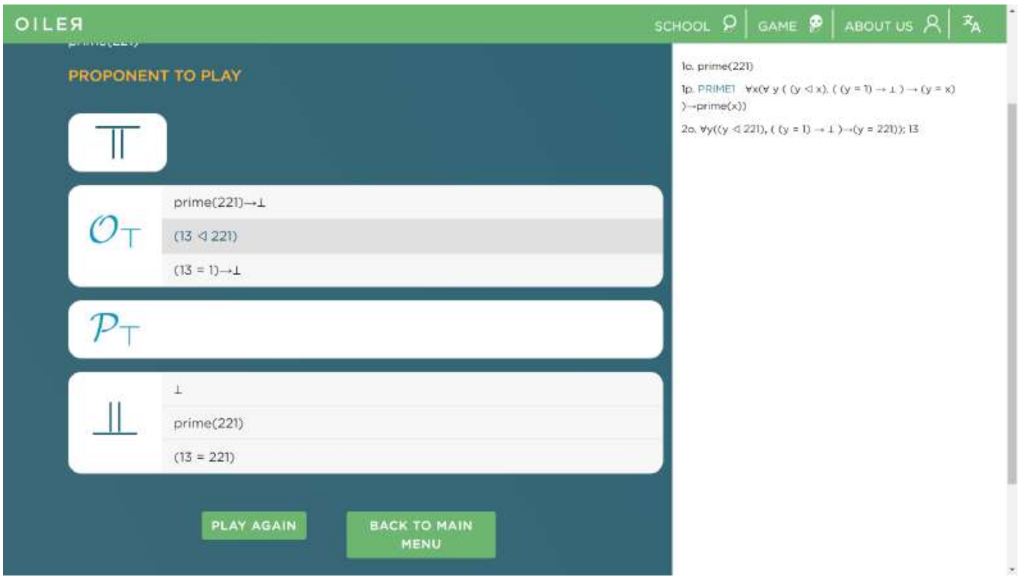

Figure 5.17.: **O** claims that 13 divides 221 even though 13 = 1 and 13 = 221 are both false.

is the negation of the previous definition; logically, the difference manifests in the swapping of quantifiers. The next six formulas, on the other hand, refer to functions that are bounded and unbounded within an interval. Logically, this involves adding an implication compared to the previous formulas, in the hypotheses of which the interval is specified. The last 9 formulas concern limits, which require, on a logical level, the addition of a third quantifier.

Let's now show an example of a play on formula number 3, namely $\exists x \forall y(\sin(y) < x)$. In other words, the formula asserts that the sine function is bounded. The formula, written in normal form, is $(\forall x(\forall y(\sin(y) < x) \rightarrow \bot)) \rightarrow \bot$.

O begins by claiming that the sine function is actually unbounded: $\forall x(\forall y(\sin(y) < x) \rightarrow \bot)$ (Figure 5.18).

P claims that the sine function is indeed bounded, stating that **O** will not be able to find a y for which $\sin(y)$ will be greater than 3 (Figure 5.19).

O is thus called upon to find a value y for which $\sin(y) \geq 3$. As shown in Figure 5.20, **O** chooses the number 0.

P wins by pointing out that $0 < 3$ is, in fact, true (see Figure 5.21 and Figure 5.22). Let's remember that the only mode available for AUOQUAMEL is indeed the shortcut mode.

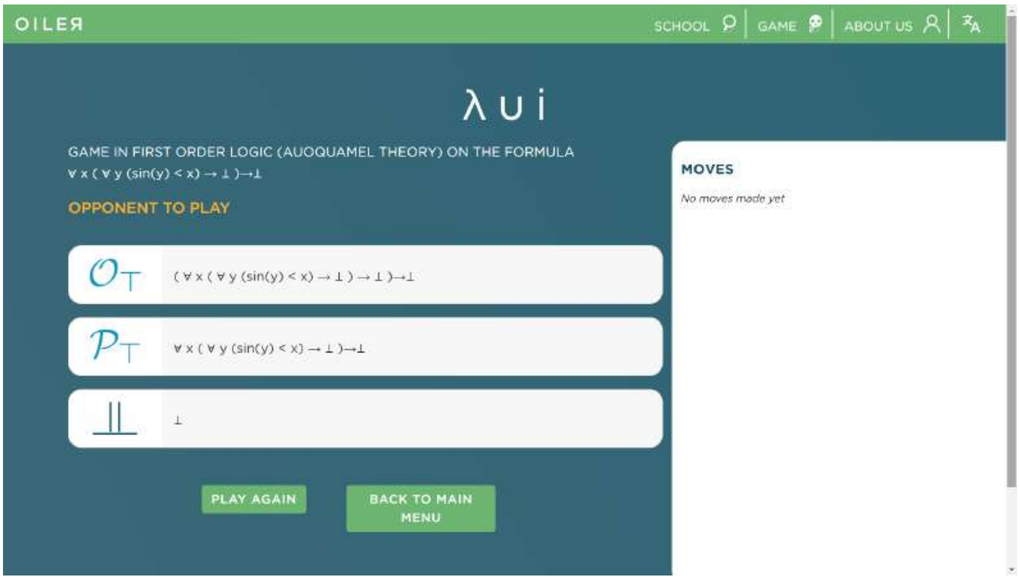

Figure 5.18.: **O** selects the only available formula in $\mathbf{P}_{\top}$.

5.3. Further Developments

5.3.1. Alfred Tarski

The reader may have noticed that we have not dealt with Euclidean geometry, as the software has not yet been implemented in this direction. We limit ourselves here to providing an intuition on how the game will be structured. Tarski's theory is a first-order formal theory for Euclidean geometry. The formalization, rather elegant, involves variables referring only to points, no function symbols (in particular, no constants) in the language, and the use of only two predicates besides equality: the ternary predicate *betweenness* $\beta(x, y, z)$, indicating that point y is aligned and lies between x and z, and the quaternary predicate *distance* $\delta(x, y, z, w)$, indicating that the distance between points x and y is equal to the distance between points z and w.

As can be immediately understood, the absence of closed terms makes the theory unsuitable for an immediate transposition into the game. However, the work of M. Beeson (2015) proves extremely useful, providing tools to make Tarski's theory constructive. An idea, evolving from Beeson's work, is that players can introduce constants to play with (*i.e.*, ordered pairs of decimal numbers) in a Cartesian environment, in the style of dynamic geometry software, with typical constructions that in these softwares are allowed.

5.3.2. AI and λuì

In our discussion, we have outlined several adjustments to enhance the game's suitability for human interaction. However, a computer engaged in strategy research is not subject to the boredom that we aimed to reduce with these modifications. Referring,

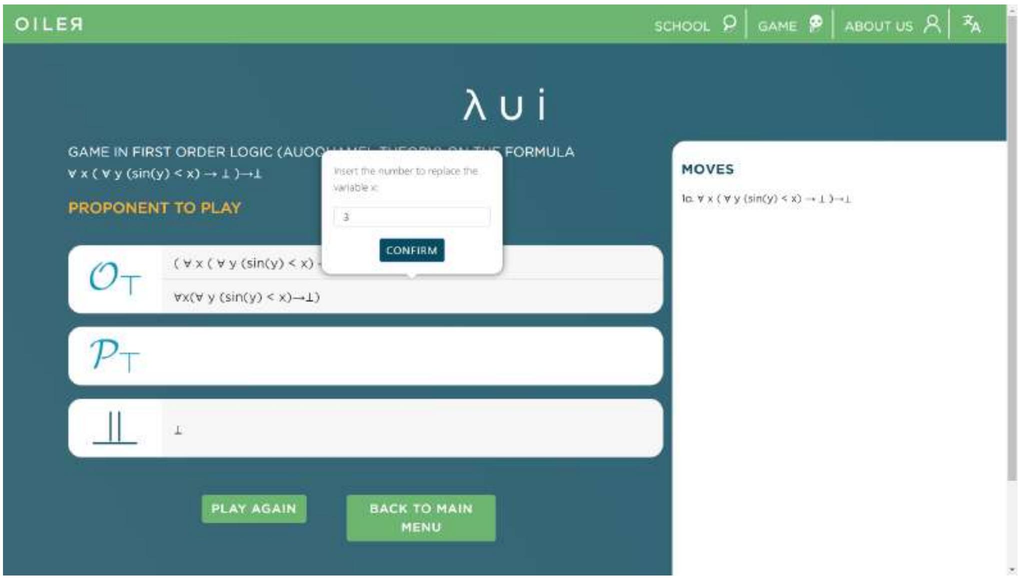

Figure 5.19.: **P** chooses the constant 3 to replace the variable x.

for example, to PA, the game can be implemented by requiring **O** to make exclusively generic moves, as described in 3, and by inserting all the axioms into $\mathbb{T}$, including the schema of the induction principle. At that point, a winning strategy for **P** would correspond to an actual proof, and we could view λuì as a proof-search program. In this direction, the possibility for users to create their own theories and formulas will be added.

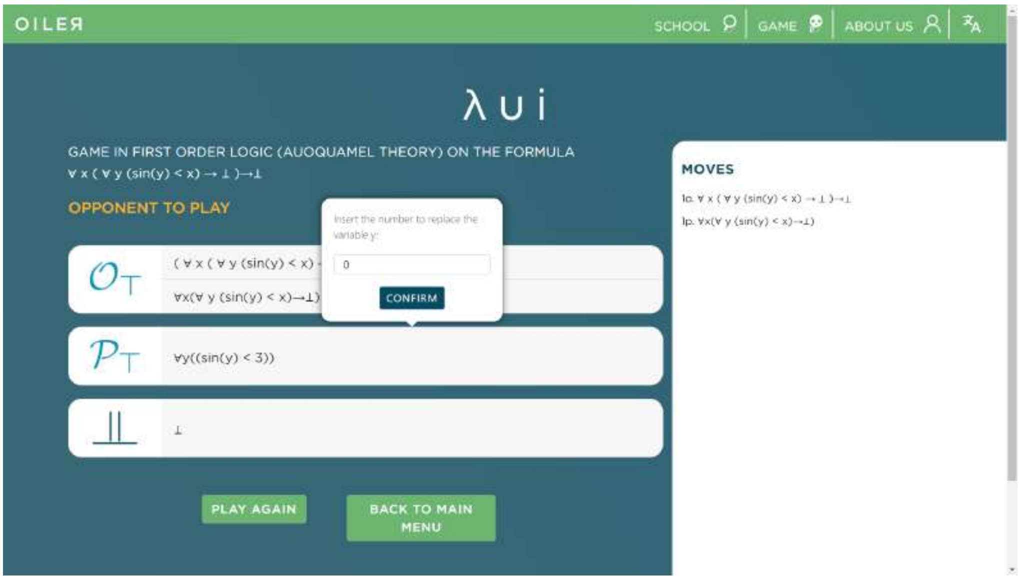

Figure 5.20.: **O** chooses the constant 0 to replace the variable y.

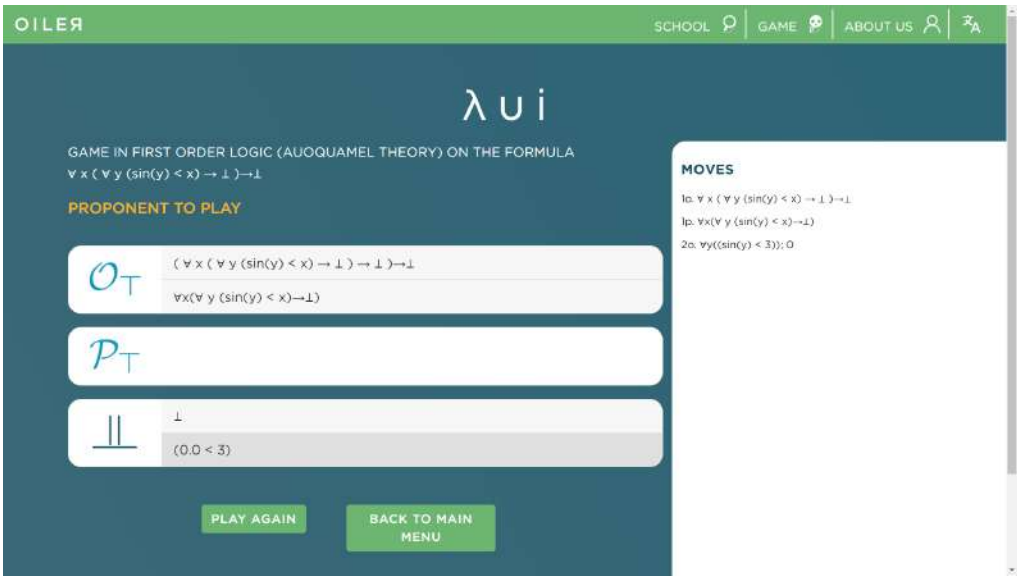

Figure 5.21.: **P** clicks on $0 < 3 \in \perp\!\!\!\perp$.

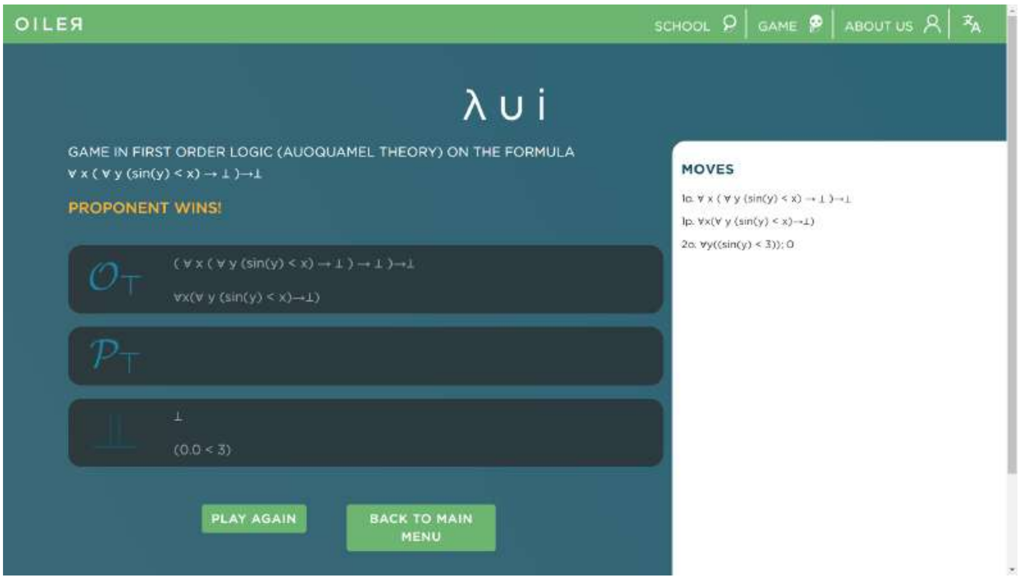

Figure 5.22.: **P** wins.