The Verifier-Falsifier Games with Restrictions on Computational Complexity of Strategies

WT-6 Stockholm Workshop 2022

Sergei Soloviev, IRIT, France

21 May 2022

Verifier-Falsifier Games and Game Semantics

- Orginally, the Game Theoretic Semantics (GTS) was developed as a variant of *verification procedure* for existing logical semantics. This semantics could be classical, constructive etc.
- References related to this talk: Thierry Coquand (1995), Denis Bonnay (2004), Boyer and Sandu (2012), Odintsov, Speranski, Shevchenko (2018).
- In this talk I will first outline the bases of the game theoretic approach. I will also consider briefly some variants of this approach and problems studied in the literature.
- I plan to explore (or, rather, to start the exploration) of **how the GTS may be modified, or even “perversed”, if there are significant differences in the computational power of players.**

Basic definition

- Semantical games are played with first-order sentences in a given model M which interprets the function and relation symbols of the relevant formal language.
- The truth (satisfaction) in M of an *atomic* formula is supposed to be fixed.
- The two players, Verifier and Falsifier play to establish the truth (falsity) of a given (compound) sentence in M.
- $\vee$-move (Verifier), $\wedge$-move (Falsifier) - choice of disjunct (conjunct).
- $\exists$-move (Verifier), $\forall$-move (Falsifier) - choice of individual $\in M$
- **Remark.** Verfier and Falsifier sometimes are called $\exists$loise and $\forall$belard.
- The players move along the syntactic tree of a given formula A. The play is always finite since an atomic subformula will be reached after a finite number of moves.

Verifier-Falsifier Games and Game Semantics

- If A is true, the play is a win for Verifier; if false, it is a win for Falsifier.
- Truth of a formula is equated with existence of a winning strategy for the Verifier, that is, a set of instructions which give Verifier a win no matter what Falsifier does. Falsity is defined analogously.

A standard example.

- Consider $A = \exists x_0 \forall x_1 . x_0 \leq x_1$.
- Consider the game played on the standard model N.
- The collection of strategies of Verifier consists of individuals (natural numbers).
- One, 0, is winning: for any number n selected by Falsifier $0 \leq n$.

Verifier-Falsifier Games and Game Semantics

- In Hintikka's semantical games, the strategies of Verifier are Skolem functions, and those of Falsifier are Kreisel's counter-examples. (Boyer/Sandu)
- The works that I cited consider *GTS* for *classical* logic, so implication is not treated, negation can be moved to atomic formulas etc. Since the aim is exploration of influence of asymmetry between players it is not a principal point.
- There are of course other works concerning *GTS* for other logics, including intuitionistic/constructive.
- Also, exploring connections with realizability:
- S. Odintsov, S. Speranski, I. Shevchenko. Hintikka's Independence-Friendly Logic Meets Nelson's Realizability. Studia Logica, 2018.

Verifier-Falsifier Games and Game Semantics

- The authors of *GTS* understood the drawbacks of the definition outlined above. Most basic: the proof in a first-order system is an effective notion, whereas truth is not. Hintikka (1996):
- **The demand of playabilty might seem to imply that the set of the initial Verifier's strategies must be restricted. For it does not seem to make any sense to think of any actual player as following nonconstructive (nonrecursive) strategy.**
- A possible solution: restrict semantical games to the games played only on recursive structures with recursive strategies.
- *CGTS′*-truth (computable game-theoretic semantics truth): **a sentence ϕ is *GTCS*-true on recursive model M exactly when there is a computable winning strategy for Verifier in the semantic game played with ϕ on M** (Boyer/Sandu)
- With free variables this is relativized to an assignment.

Verifier-Falsifier Games and Game Semantics

- Boyer and Sandu then consider the case when the structure M is N, since N is the only recursive structure of PA (up to isomorphism), by Tennenbaum's theorem.
- So they consider effective winning strategies for Verifier in semantic games played on N.
- **Example.** On the standard structure N the Verifier has a computable winning strategy for the sentence $\forall x_0 \exists x_1.(x_0 \geq x_1)$ iff there is a recursive function $f : N \to N$ such that for all $n \in N$, $n \geq f(n)$. That is, $N| =_{CGTS} \forall x_0 \exists x_1.(x_0 \geq x_1)$
- More generally, for any binary predicate $F(x, y)$, $N| =_{CGTS} \forall x \exists ! y.F(x, y) \iff F(x, y)$ *defines a total recursive function*.

Verifier-Falsifier Games and Game Semantics

Two questions:

1. Do proofs in *PA* yeld *CGTS*-truth:

   $$PA \vdash \phi \Rightarrow PA| =_{CGTS} \phi?$$

   Here $\Gamma| =_{CGTS} \phi$ is defined by the condition that in all recursive models M: if for all $\psi \in \Gamma$, $M| =_{CGTS} \psi$, then $M| =_{CGTS} \phi$.

2. Can the *CGTS*-truth of a sentence be always interpreted as given by a proof?

   $$PA| =_{CGTS} \phi \Rightarrow PA \vdash \phi?$$

   And, thus $PA| =_{CGTS} \phi \iff PA \vdash \phi$?

The answer to both questions is negative.

Verifier-Falsifier Games and Game Semantics

- To (1) a standard counterexample is given by the sentence

$$\forall x_1 \forall x_2 \exists y \forall z.(Halt(x_1, x_2, y) \vee \neg Halt(x_1, x_2, z)).$$

  Here $H(x_1, x_2, z)$ is the predicate that represents the "halting" of the Turing mashine encoded by x_1 on x_2 after z steps. There is no recursive winning strategy for Verifier on N since othewise the halting problem would be decidable.
- But the sentence is provable in *PA*.

Verifier-Falsifier Games and Game Semantics

- The answer to (2) is negative as well.
- Consider ϕ of the form $\forall x \exists ! y . F(x, y)$.

$$N| =_{CGTS} \forall x \exists ! y . F(x, y) \iff F(x, y)$$

defines a total recursive function.

- Taking into account the Tennenbaum's theorem we would have

$$PA \vdash \forall x \exists ! y . F(x, y) \iff F(x, y)$$

- but then the set of total recursive functions would be recursively enumerable.

The Games with Backward Moves

- To obtain a positive answer at least to (1) several authors modify the notion of semantic game.
- Coquand (1995), Krivine (2003), Bonnay (2004).
- They admit an important asymmetry: one of the players (in their work the Verifier) is permitted to go back and change a move.
- They introduce the games with backward moves.

The Games with Backward Moves

The Main Differences:

- Whenever its turn to move, Verifier can return to any one of its earlier decision points and remake the choice; the play then continues as in the standard game -
- even if the false atomic formula is reached (win for Falsifier in the standard game) return to one of the earlier decision points for Verifier is permitted (and the play then continues as in the standard game)
- Verifier wins a play if it is finite and it ends with a true atomic formula, otherwise Falsifier wins (in case of infinite play that is now possible as well).
- Now both players may have more strategies that in standard games. It has important consequences.

The Games with Backward Moves

- **Example** (Coquand). Consider the backward game for $(\exists m \forall x. x \leq m) \vee (\forall n \exists y. n < y)$ played on N. The Verifier has the following winning strategy (not in the standard game):
- V. chooses the right disjunct;
- F. chooses a value n_0 for n;
- V. goes back, chooses the left disjunct and n_0 for m
- F. chooses some x_0 for x.
- Now, if $x_0 \leq n_0 = m$, then V. wins.
- Otherwise, if $x_0 > n_0 = m$, V. goes back to its choice of disjunct, and chooses instead to continue on the basis of the right dijunct again after the choice made by F.,
- that is, where $n = n_0$, and V. may choose $y = x_0$ and win the play.

The Games with Backward Moves

- For the formula

$$\forall x_1 \forall x_2 \exists y \forall z.(Halt(x_1, x_2, y) \vee \neg Halt(x_1, x_2, z))$$

  the Verifier has now a winning strategy as well!
- In the beginning F chooses $x_1 = m_1$ an $x_2 = m_2$;
- V has to choose some $y = n$;
- now it is F's turn, it has to choose $z = p$;
- V may choose a disjunct; but it looks first what the value of disjuncts is: if $Halt(m_1, m_2, n)$ is true it chooses this disjunct;
- if it is false and $\neg Halt(m_1, m_2, p)$ is false then $Halt(m_1, m_2, p)$ is true; V goes backwards, chooses $y = p$ and (after any choice of $z = p'$ by F) chooses the left disjunct. And **wins**.

The Games with Backward Moves

- There are two theorems proved by Denis Bonnay.
- The first one speaks about any strategies, not only computable.
- **Theorem 1.** For any first order formula ϕ, structure M and assignment g, Verifier (Falsifier) has a winning strategy in the standard semantical game $G(M, \phi, g)$ iff it has a winning strategy in the corresponding game with backward moves.
- **Theorem 2.** If M is a recursive model, π is a proof (in classical logic) of $\Gamma \vdash \phi$ and recursive winning strategies $\{f_i\}_{i \in \Gamma}$ for Verifier are given for each game $G^*(M, \phi_i, \emptyset)$ with backward moves, with $\phi_i \in \Gamma$, then π yelds a recursive winning strategy for Verifier in $G^*(M, \phi, \emptyset)$.

The Games with Backward Moves

- So, Bonnay's theorem 2 says that **if ϕ is provable classically, then there is a winning strategy for Verifier.** This gives a positive answer (for games with backward moves) to the first question mentioned above.
- The answer to the second question, **whether the existence of winning strategy for V implies provability, remains negative.**
- The price of this one positive answer is introduction of an important asymmetry between players.
- The asymmetry is not in computation power, but it so to say "opens the way".
- And the fact that the answer to (2) remains negative makes us to ask, **what strange formulas may be "proved by winning"?**

Computational Asymmetry

- **Example. Let** $\phi = \exists x \forall y.(y \leq \mathcal{A}(x))$**. Let here** $\mathcal{A}$ **be the Ackermann's function**, and let the class of strategies of Falsifier be limited to **primitive recursive functions.**
- The strategies of Verifier are just natural numbers (values of x). If f is some strategy of Falsifier, its answer is $f(x)$. The formula is *false* on N, but there is no winning strategy for Falsifier because $\mathcal{A}$ grows faster than any PR function.
- The games themselves are yet symmetric (no backward moves), we can consider $\psi = \forall x \exists y.(\mathcal{A}(x) < y)$ (which is *true*), and here the Verifier will have no winning strategy if its strategies are PR.
- In fact, both don't have winning strategy in my example. So, it is not yet an example when the more powerful player can completely "perverse" the semantics. **However:**

Computational Asymmetry

- Assume that V can compute any general recursive function and knows (and can compute) a universal function $U(x, y)$ for the strategies f of F, i.e., every $f = U(k, -)$ for some k.
- Assume that if V knows the strategy of F it can win. That is, V can compute another function $W(x, y)$ such that $v_k = W(k, -)$ wins against $f_k = U(k, -)$.
- Here $x \in N$ but we may assume that $y \in N$ are the codes of partial plays (including backward moves).
- **Remark.** In our work Falsifier can (in its strategy) take into account the backward moves. But is does not change main result.

Computational Asymmetry

- **Theorem.** In the conditions listed above the Verifier has a recursive strategy that wins against any strategy of the Falsifier.
- *Proof.* The winning strategy of Verifier is constructed using “testing of hyptheses”. Initial hypothesis is that F uses the strategy $f_0 = U(0, -)$. When the current hypothesis has number k (that the strategy of F is $f_k = U(k, -)$), V plays using his strategy $v_k = W(k, -)$ while the moves of F are as predicted. If they are not then V returns to the initial position and passes from k-th hypothesis to the $k+1$-th.
- **Remark.** They can arrive to a position that is losing to V in the standard formulation of the game, but in the game with backwards moves V can backtrack. So this case is also included in the description of the strategy of V.
- V wins either when it arrives to the correct hypothesis *or* before. So the problem of “true” number of strategy remains indecidable.

Computational Asymmetry

Example with Ackermann function (continued)

- If the strategies of Falsifier do not take into account the backward moves, the application of the theorem is very simple.
- A strategy f of F is a PR function $f : N \to N$.
- Let $U(k, y)$ be a universal function for PR functions.
- The function $W(k)$ is $\mu x.(U(k, x) \leq \mathcal{A}(x))$ (the second argument is absent because we need only the intial value).
- It is general recursive (by classical results of Kleene).
- The winning strategy for V backtracks if $f(W(k)) > \mathcal{A}(W(k))$ and chooses $W(k + 1)$.
- **Remark.** Other solutions (not based on the theorem) are possible in this example. Say, V can just take the values $0, 1, ...$ for x (backtrack and choose $k + 1$ if $f(k) > \mathcal{A}(k)$).

Example with Ackermann function (continued)

- Let the strategies of Falsifier do take into account the backward moves. In this example there is only one backward move, so we can represent a partial play just by sequence of values of x (chosen by V) "mixed" with the values of $f(x)$.
- This sequence may be represented by its number (using enumeration of finite sequences of natural numbers).
- If y is a sequence, let $< y >$ be its number.
- Next move of a player is given by the value $v(< y >)$ $(f(< y >))$. Strictly speaking, we should distinguish whose move there is, but it can be done by appropriate convention.

Example with Ackermann function (continued)

- Notice that if f is PR, then $f(< y, x >)$ is PR on x when y is fixed.
- We may pose $W(k, y) = \mu x.(U(k, < y, x >) \leq \mathcal{A}(x))$.
- Again, by Kleene's results, it is general recursive.
- The rule proposed in the theorem (change k to $k + 1$ if $f(< y, W(k, y) >) \neq U(< y, W(k, y) >)$) defines a general recursive winning strategy of V.
- (The change is determined by y as well.)
- There are other winning strategies for V, not only based on the theorem.

Conclusion

- The context is rather that of scientific method, than purely mathematical.
- Often in everyday practice a massive computer based testing/verification/simulaton is used to complete/supplant proof.
- To rely on it, an absolute scientific integrity/honesty is expected. The biais maybe inconsious or even intended. And what can be opposed? It turns towards some sort of V/F game.
- However, while the asymmetry of the rules (like backward moves for **one** player) can be easily controlled, it is more difficult to detect and estimate the difference in computational power.
- There are works on GTS for Type Theory, e.g., Yamada (2018). It was used mostly for “full completeness” results. Does “distortion of truth” due to computational asymmetry have some independent interest there? It is to be explored.

THANKS FOR YOUR ATTENTION!

References

1. T. Coquand. A Semantics of Evidence for Classical Arithmetic. - J. of Symb. Logic, 60, 1, 325-337 (1995)
2. J.-L. Krivine. Dependent choice, ‘quote’ and the clock. TCS, 308, 1-3, 259-276 (2003).
3. D. Bonnay. Preuves et jeux sémantiques. Philosophia Scientiae, 8(2), 105-123 (2004).
4. J. Boyer, G. Sandu. Between Proof and Truth. Synthese, 187 , 821-832 (2012).
5. S.P. Odintsov, S.O. Speranski, I.Yu. Shevchenko. Hintikka’s Independence-Friendly Logic Meets Nelson’s Realizability. Studia Logica, 637-670 (2018).
6. N. Yamada. Game semantics of Martin-Lof type theory. Preprint submitted to RMS (2018)