On Semantic Gamification

Ignacio Ojea Quintana

Columbia University

Abstract. The purpose of this essay is to study the extent in which the semantics for different logical systems can be represented game theoretically. I will begin by considering different definitions of what it means to *gamify* a semantics, and show completeness and limitative results. In particular, I will argue that under a proper definition of gamification, all finitely algebraizable logics can be gamified, as well as some infinitely algebraizable ones (like ukasiewicz) and some non-algebraizable (like intuitionistic and van Fraassen supervaluation logic).

1 Introduction

1.1 Logic gamification

The present work builds on the well established work on game semantics for classical logic developed by Jaakko Hintikka [5, 6], and independently by Rohit Parikh [13]. It is embedded in a research line that seeks to build formal connections between logic and game theory, systematically developed in van Benthem [18]. Its contribution amounts extending some of those results to non-classical logics, and to provide an answer to the general question of which semantics can be represented game-theoretically.

In the past few years there have been several developments in game semantics for many valued logics, for example by Fermüller [2, 3]. In particular there are several well studied applications on games in Łukasiewicz-style (fuzzy) logics; Mundici [11] provides an alternative semantics for finite-valued Łukasiewicz logics in terms of Ulam's games and Cintula & Majer [1] develop an approach similar to what is going to be done here. Here I will briefly discuss the differences between my approach and Fermüller's, and provide a justification for my account.

But the central issue in this essay is to clearly define and discuss what it means for a semantics to be *gamifiable*, and to show that under an appropriate definition (a) all finitely-algebraizable logics are gamifiable, and (b) some non-algebraizable logics are gamifiable.

1.2 The basic case

Perfect information games in extensive form are trees whose nodes are possible states and turns for the players, arrows from a node to its children represent the available moves or actions that the player has at that node. A strategy for a player i is a function that assigns a move at each node corresponding to a turn for player i. In terminal nodes payoffs are assigned for the players.

In Evaluation Games for Classical Propositional logic, two players **V** and **F** (for *Verifier* and *Falsifier*) dispute over the truth value of a formula ϕ of a language $\mathcal{L}$ in some model $\mathbf{M}$. To avoid unnecesary complications, I assume that $\mathcal{L}$ is Classical a Propositional language. It is possible to assign a game $G_\phi^\mathbf{M}$ to each pair $< \phi, \mathbf{M} >$ of formulas and Classical Propositional models in the following way:

- If $\phi = p$ for atomic sentence p, then $G_p^\mathbf{M}$ is a single (terminal) node tree in which **V** wins if $\mathbf{M} \vDash p$ and **F** wins otherwise.
- If $\phi = \neg\alpha$, then $G_\phi^\mathbf{M}$ is $G_\alpha^\mathbf{M}$, with turns and win-lose markings reversed.
- If $\phi = \alpha \vee \beta$, then $G_\phi^\mathbf{M}$ is a tree that starts with a node which has $G_\alpha^\mathbf{M}$ and $G_\beta^\mathbf{M}$ as its only children and that it is a turn for **V**. The basic idea is that she decides with which subformula to continue the game.
- Finally, if $\phi = \alpha \wedge \beta$, then $G_\phi^\mathbf{M}$ is a tree that starts with a node which has $G_\alpha^\mathbf{M}$ and $G_\beta^\mathbf{M}$ as its only children and that it is a turn for **F**. The basic idea is that he decides with which subformula to continue the game.

The point of such assignment is that the following bridging result holds:

Proposition 1 (Success for Classical Propositional Logic). *For all formulas ϕ and Propositional models $\mathbf{M}$: $\mathbf{V}$ has a winning strategy in $G_\phi^\mathbf{M}$ if and only if $\mathbf{M} \vDash \phi$.*

The well-known result is due to Hintikka and it generalizes to first order logic. A first intuitive definition of what it means for a semantic to be *gamifiable* can be generalized from this result.

Definition 1 (Semantic gamification - intuitive).

We start with a logic $\mathbf{L}$ and a semantics $\mathcal{S}$ for that logic that assigns truth values to the formulas in the language of $\mathbf{L}$. We say $\mathcal{S}$ is intuitively gamifiable *if there is a game theoretic representation $G^\mathcal{S}$ and a game-theoretic condition (expressed using a solution concept) $\mathcal{C}$ such that for all formulas ϕ in the language, $\mathcal{S}$ assigns certain truth v to ϕ if and only if the condition $\mathcal{C}$ applies to the game theoretic representation of the formula $G_\phi^\mathcal{S}$.*

1.3 Structure of the essay

In the next section I will discuss gamification for finitely algebraizable logics. In particular, I will present a hierarchy of notions of *gamification* and show the extent to which those logics can be gamified. Also, I will discuss some of the philosophical and technical aspects of those definitions.

In section three I will discuss non-finitely algebraizable logics, in particular intuitionistic and supervaluationist. The purpose of this section is to show that game semantics can be viewed as more general than the standard approaches to logics for semantics.

The last section includes concluding remarks on the significance of the results.

2 Finitely algebraizable logics

2.1 Logical Matrices

Logical Matrices were first introduced by Łukasiewicz and Tarski [9] in the 1920's as a general concept that was implicitly used in the work of other logicians. The reader can refer to [4, 16] for a more advanced treatment than the one given here. The basic idea is a generalization of the Boolean Algebra underlying truth values in Classical Logic. Formulas are assigned truth values in the domain of the algebra and connectives are interpreted as the algebraic operations over those truth values.

Given a propositional language $\mathcal{L}$, a $\mathcal{L}-$ **matrix** is a pair $<\mathbf{A}, F>$ where $\mathbf{A}$ is an algebra of type-$\mathcal{L}$ with universe $\mathcal{A}$, and $F \subseteq \mathcal{A}$; where F is the set of designated values. An assignment h is an homomorphism from the algebra of formulas $\mathbf{Fm}$ to the algebra $\mathbf{A}$ of the same $\mathcal{L}$-type $[h \in Hom(\mathbf{Fm}, \mathbf{A})]$. Here the elements of the algebra serve as the truth-values of the semantics. One of the key features here is compositionality. h starts by assigning elements of $\mathcal{A}$ to the set *Var* of propositional variables and can be extended to all of $\mathcal{L}$ by interpreting operations in the language as operations in the algebra:

- $h(p_i) = a_i$, where $p_i \in Var$ and $a_i \in \mathcal{A}$.
- $h(\neg\phi) = \neg^{\mathbf{A}} h(\phi)$.
- $h(\phi * \psi) = h(\phi) *^{\mathbf{A}} h(\psi)$, for any diadic connective $*$.

The notion of **model** is the same as before. A logic $\mathbf{L}$ in the language $\mathcal{L}$ is said to be *complete relative to a class of* $\mathbf{L}$*-matrices* $\mathbf{M}$, if all the elements of $\mathbf{M}$ are models of $\mathbf{L}$ and for every $\Gamma \cup \{\phi\} \subseteq Fm$ such that $\Gamma \nvdash_{\mathbf{L}} \phi$ there is a matrix $<\mathbf{A}, F> \in \mathbf{M}$ and $h \in Hom(\mathbf{Fm}, \mathbf{A})$ such that $h[\Gamma] \subseteq F$ but $h(\phi) \notin F$. If this is the case, then it is said that $\mathbf{M}$ is a *matrix semantics for* $\mathbf{L}$, or that $\mathbf{M}$ is *strongly characteristic for* $\mathbf{L}$. In particular, if $\mathbf{M}$ is a singleton with matrix M, then M is the characteristic matrix of $\mathcal{S}$.

Definition 2 (Finitely algebraizable).
A logic $\mathbf{L}$ *is finitely algebraizable if it is complete relative to a class of* finite $\mathbf{L}$*-matrices* $\mathbf{M}$.

2.2 Games

In this work we are interested in a very restricted class of games: two player perfect information extensive games of finite depth [and in almost all cases, strictly competitive or zero-sum games]. As before the players are $\mathbf{V}$ and $\mathbf{F}$. We will introduce the basic notions following [10, 12]; where the reader should turn for a more elaborate presentation.

An *extensive game model* is a tree $G =< S, R, turn, \mathcal{V} >$ with a set of state-nodes S and a family R of binary transition relations for the available moves, pointing from parent to daughter nodes. R is assumed here to be well-founded[1] in

[1] Since R is well founded, branches of the trees have only finite depth.

that there is no infinite sequence $< a_1, a_2, ... >$ of nodes such that $< a_i, a_{i+1} >\in M$ for all $i \in N$. *turn* is a function that assigns players to non-terminal nodes, indicating the player whose turn it is. $\mathcal{V}$ is a function that assigns utility values for players at all terminal nodes, but possibly also to any other node.[2]

A *strategy* for player i is a function s_i that assigns at each of i's turns one of the available actions. A *mixed strategy* for a player i is a function $\sigma_i : S_i \to [0, 1]$ which assigns a probability $\sigma_i(s_i) \geqslant 0$ to each pure strategy $s_i \in S_i$, satisfying that $\sum_{s_i \in S_i} \sigma_i(s_i) = 1$.

Given a set of players $I = \{1, ..., n\}$, a *pure strategy profile* is an n-tuple $< s_1, ..., s_n >$ where each s_i is a pure strategy for player i. Each pure strategy profile is associated with a terminal node in the game model, the one that would be reached if players played the strategy in the profile. Furthermore, given a pure strategy profile $< s_1, ..., s_n >$, $\mathcal{V}_i(< s_1, ..., s_n >) =_{df} \mathcal{V}_i(a)$, where a is the terminal node of that strategy profile and $\mathcal{V}_i$ is the utility for any player i. The payoff of a (possibly mixed) strategy profile $< \sigma_1, ..., \sigma_n >$, $\mathcal{V}_i(< \sigma_1, ..., \sigma_n >) = \sum_{<s>\in S}[\sigma_1(s_1)...\sigma_1(s_n)]\mathcal{V}_i(< s_1, ..., s_n >)$.

The solution concept that we will be using in almost all cases is that of Nash Equilibrium: A strategy profile $< \sigma_1, ..., \sigma_n >$ is a *Nash equilibrium* if and only if for any player $i \in \{1, ..., n\}$ and any strategy $\sigma'_i \neq \sigma_i$ for that player, $\mathcal{V}_i(< \sigma_1, ..., \sigma_i, ..., \sigma_n >) \geqslant \mathcal{V}_i(< \sigma_1, ..., \sigma'_i, ..., \sigma_n >)$. The insight behind Nash Equilibrium is that unilateral deviation is not profitable. Once the strategy profile is reached, no player has an incentive to change strategies given the other player's strategic choices are fixed. Yet a particular subset of the Nash Equilibra will be used here, namely those obtained by the Backward Induction procedure.

2.3 Strong gamification

When evaluation games for Classical Propositional logic were introduced before, there was an implicit function **game** that assigned extensive game trees of the ones just presented to formulas in $\mathcal{L}$ in some model $\mathbf{M}$; so that ϕ in $\mathbf{M}$ got assigned to $G^{\mathbf{M}}_{\phi}$. This way of presenting the evaluation games followed van Benthem in [17] and Parikh in [13, 14].

The simple generalization proposed here requires us to drop the model dependence of the function, so that each formula $\phi \in \mathcal{L}$ gets an *game form*, a tree G_ϕ in all which terminal nodes $< p_i >$ corresponding to atomic sentences p_i have no assigned payoffs for the players. We later define $\mathcal{V}$ in a way that assigns members of the relevant (non-Classical) matrix to *terminal* nodes - but can be extended to other nodes. In general, given a game G, $\mathcal{V}(G)$ is the payoff that *Verifier* gets in the (relevant) equilibria of G.[3]

In detail, state-nodes of the trees are members of S and are denoted here with tuples $< \phi >$, where $\phi \in \mathcal{L}$. It is useful to reformulate the definition of $\mathbf{game}(\phi) = G_\phi$:

[2] A further assumptions is that there is complete and perfect information.

[3] For the games considered, it is not hard to show existence of equilibria as well as uniqueness of payoff under all equilibria.

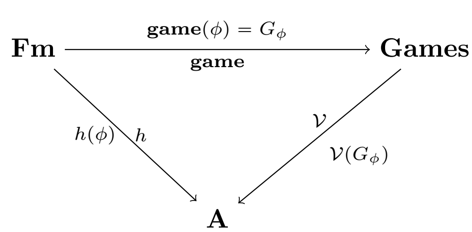
– G_{p_i} is a single node tree $< p_i >$, which can be seen as a test or payoff gaining game.
– $G_{\neg\phi}$ is G_ϕ with turns reversed, replacing each terminal atomic node $< p_i >$ by a node $< \overline{p_i} >$ *and vice versa*. Also, formulas $< \phi >$ in game nodes are syntactically dualized, interchanging conjunctions and disjunctions.
– $G_{\phi\vee\psi}$ is the disjoint union of two game trees G_ϕ and G_ψ put under a common root node $< \phi \vee \psi >$ that is a turn for **V**.
– $G_{\phi\wedge\psi}$ is the disjoint union of two game trees G_ϕ and G_ψ put under a common root node $< \phi \wedge \psi >$ that is a turn for **F**.

It is worth noticing that in this definition G_ϕ is generated solely from the syntactic structure of ϕ.

Let us go back briefly to the *alethic* or Model-theoretic approach to logic. Part of the gist of it is that we are able to model our natural or intuitive understanding of the connectives that appear in the formula algebra **Fm** with operations in our modeling algebra **A**. This was captured by the fact that for any assignment (homomorphism) $h : \mathbf{Fm} \to \mathbf{A}$, $h(\neg\phi) = \neg^{\mathbf{A}} h(\phi)$ and $h(\phi * \psi) = h(\phi) *^{\mathbf{A}} h(\psi)$. Thus the algebra can successfully represent the alethic structure that we want it to embody. In the *pragmatic* or game-theoretic approach to logic, we want to have a relation of the same sort between the games and some underlying algebra. This will be captured by analogous principles:

– $\mathcal{V}(G_{p_i}) = a_i$, where g_i is an atomic game and $a_i \in \mathcal{A}$.
– $\mathcal{V}(G_{\neg\phi}) = \neg^{\mathbf{A}} \mathcal{V}(G_\phi)$.
– $\mathcal{V}(G_{\phi*\psi}) = \mathcal{V}(G_\phi) *^{\mathbf{A}} \mathcal{V}(G_\psi)$ for any diadic connective $*$.

Furthermore, in principle nothing ensures that the algebraic operation will coincide with our strategic intuitions and theories about how games are resolved (i.e. its equilibria). Conversely, it should not be clear *prima facie* that concepts in game theory and game structures function the same way as algebraic transformations. Yet, at least for some algebraic structures we know that the relation holds. The overall project is then:



Given a formula algebra **Fm**, an underlying algebra **A** and $h \in Hom(\mathbf{Fm}, \mathbf{A})$, the central purpose of evaluation games is to provide a translation function **game** and a payoff function $\mathcal{V}_h$ so that for any formula $\phi \in \mathbf{Fm}$:

$$h(\phi) = \mathcal{V}^h \circ \mathbf{game}(\phi)$$

Definition 3 (Strong Semantic Gamification).

Begin with a logic **L** *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* strongly gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is (a) a game theoretic translation* G^{ϕ}*, (b) a payoff assignment* $\mathcal{V}_h$ *to* G^{ϕ} *that is defined in terms of* h *and (c) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to the game* G^{ϕ} *with payoffs determined by* $\mathcal{V}_h$*.*

I will now overview a few results that show that some many valued logics are *strongly* gamifiable. Since the precise cases considered here are not the main focus of the present essay, I will only provide a superficial presentation of each case. Nevertheless, the reader is invited to read some of the proofs in the appendix to get a gist of the basic techniques used here.

2.4 Strong Kleene

We start with Kleene's 3-valued system developed in [7, 8] because generalizing evaluation games for it is straightforward. The set of truth values is $\mathcal{K} = \{1, \frac{1}{2}, 0\}$, where "1" codes truth, "0" codes falsity and "$\frac{1}{2}$" codes undefined. The operations $\neg^{\mathbf{K_3}}$, $\vee^{\mathbf{K_3}}$, $\wedge^{\mathbf{K_3}}$ are defined in analogy to Classical logic: (a) $\neg^{\mathbf{K_3}} x = 1 - x$, (b) $x \vee^{\mathbf{K_3}} y = max\{x, y\}$ and (c) $x \wedge^{\mathbf{K_3}} y = min\{x, y\}$.

In order to develop evaluation games for Kleene's 3-valued system I need, given a homomorphism h, a translation function G (or **game**) and an evaluation function $\mathcal{V}^h$ that assigns payoffs to terminal nodes of those trees and values to complex game trees using a solution concept. The translation function is the same as for the classical case. The valuation function $\mathcal{V}^h$ needs first to assign members of $\mathcal{K} = \{1, \frac{1}{2}, 0\}$ to the terminal nodes $< p_i >$ and $\overline{< p_i >}$ such that the payoff of *both* players are specified. $\mathcal{V}^h(G_\phi)$ is the payoff that *Verifier* gets in the equilibria of the game G_ϕ; and since I am considering strictly competitive games, the payoff that *Falsifier* gets will be 1-$\mathcal{V}^h(G_{p_i})$.

The valuation for terminal nodes is:

- $\mathcal{V}^h(G_{p_i}) = h(p_i)$. Hence *Verifier* gets $h(p_i)$ and *Falsifier* gets 1-$h(p_i)$.
- $\mathcal{V}^h(\overline{G_{p_i}}) = 1 - h(p_i)$. Hence *Verifier* gets $1 - h(p_i)$ and *Falsifier* gets $h(p_i)$.

Proposition 2 (Success for Strong Kleene). *Given the matrix* $\mathbf{K_3}$ *and arbitrary assignment* h*, for all formulas* $\phi \in \mathcal{L}_{\mathcal{K}}$*:* $h(\phi) = x$ *if and only if* $\mathcal{V}^h(G_\phi) = x$ *[i.e. in all the Nash Equilibria in* G_ϕ Verifier *gets a payoff of* x*].*

The proof of the proposition is included in the appendix. Furthermore, the observant reader might have noticed that there is nothing essential in the fact that only *three* truth-values were considered. What is crucial is that the truth values are linearly ordered and the (Kleene) operations correspond to *max*, *min* and dualization. Then any logic of this form, with finite or infinite truth values, can be modeled analogously with a strictly-competitive two player game.

2.5 Gamification and some results

A natural question is whether all finitely algebraizable logics can be strongly gamified. I do not have an answer to this. Nevertheless, it is possible to show that all finitely algebraizable logics can be gamified, under a weaker notion of gamification.

Definition 4 (Semantic Gamification).
Begin with a logic **L** *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is a game* G_h^{ϕ} *whose* structure *and payoffs* depend on h *and (b) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to the game* G_h^{ϕ}.

The crucial difference here is that formulas are *not* mapped to game *forms*, but rather the mapping goes from formulas and assignments to completely specified games. In particular, the same formula can be mapped to different games under different assignments (and of course different matrices).

The strategy adopted here to show that all finitely algebraizable logics can be gamified is indirect. The first step consists in showing that Post logics are gamifiable. The second, to argue that this is sufficient given the truth-functional completeness of those logics.

2.6 Post and truth-functional completeness

In 1921 [15] Emil Post presented a finitely many valued logic and *showed that it is truth-functionally complete* i.e. that all truth functions $f : \mathcal{A}^n \rightarrow \mathcal{A}$ are expressible in terms of the truth functions corresponding to the connectives provided by that logic. Post's intepretation of the disjunction and conjunction is similar to that of Strong Kleene and Łukasiewicz, *max* and *min* respectively. The most salient feature of Post logic is Post's negation $\sim$; so let $\mathcal{L}_{\sim}$ by $\mathcal{L}$ augmented with that connective. Its interpretation -when $\mathcal{A} = \{0, ..., n\}$ - is the following:

– $h(\sim \phi) = h(\phi) - 1(mod\ n + 1)$.

Proposition 3 (Success for Post). *For every formula* ϕ *in the language of Post, truth value* v *and assignment h:* $h(\phi) = v$ *if and only if in all the Nash Equilibria in* G_{ϕ}^{h} Verifier *gets a payoff of* v*.*

The proof of this result, although innelegant and tedious, is included in the Appendix.

Any matrix $\mathbf{M} =< \mathbf{A}, \mathbf{F} >$ with finite universe $\mathcal{A}$ can be represented in a Post matrix of size $|\mathcal{A}|$, making use of the fact that it is truth-functionally complete. This is done in two steps. First, by corresponding each truth value in the matrix with a truth value in the Post logic. Second, the interpretation that matrix **M** gives to each connective is nothing more than a truth function; which by truth functional completeness can be captured in the Post logic by some composition of Post connectives. In a nutshell, providing a game semantic for Post logic is virtually the same as providing a game semantic for any finite matrix.

A similar result was given by Fermüller [3] in 2013, but with a different approach. There Fermüller associates games with *signed* formulas, which capture the idea that Verifier asserts a certain truth value for the formula at hand. For example, the expression '$v : \phi$' stands for Verifier's claim that the formula ϕ has truth value v in the relevant assignment. His basic idea is to have win-lose games in which Verifier makes the assertion that ϕ has certain truth value and Falsifier contests that assertion. In this way, his result are also expressed in terms of winning strategies, rather than Nash Equilibria or Backwards Induction solutions. This is, h assigns v to ϕ if and only if Verifier has a winning strategy in the game corresponding to $v : \phi$.

3 General Gamification

So far the focus of the paper has been on finitely-algebraizable logics, but what about other kinds of logics? Allow me to slightly generalize the definition of semantic gamification so that formulas in a semantics are represented by a *set* of games, rather than a single game.

Definition 5 (General Semantic Gamification).
Begin with a logic $\mathbf{L}$ *and a semantics* $\mathcal{S}$ *for that logic. We say* $\mathcal{S}$ *is* gamifiable *if for each formula* ϕ *in the language and each assignment* $h_{\mathcal{S}}$ *there is a* set *of games* G_h^{ϕ}*, each of whose* structure *and payoffs* depend on h *and (b) a game-theoretic condition (solution concept)* $\mathcal{C}$ *such that: For all formulas* ϕ*, an assignment* $h_{\mathcal{S}}$ *assigns certain truth* v *to* ϕ *if and only if the condition* $\mathcal{C}$ *applies to all the games in* G_h^{ϕ}*.*

Under this simple generalization, it is not hard to show that some non-algebraizable logics, such as Intuitionistic and Supervaluationistic, are gamifiable in general.

3.1 Supervaluationist

Supervaluationist logic was developed by Van Fraassen [19, 20] to treat issues of vagueness while satisfying some classical logic principles. The basic idea to evaluate a formula is to start with a partial assignment with three truth values and consider all the classical extensions of that assignment. If the formula is true in all its classical extensions, then it is true in the supervaluation; similarly for falsehood. If the formula is true in some extensions and false in others, then it gets an intermediate value.

More formally, an *initial* truth-value assignment is any function h such that for $h(p_i) \in \{0, \frac{1}{2}, 1\}$ for all propositional variables p_i and that is extended to all formulas using the Strong Kleene compositional rules. A *classical extension* h' to a initial truth-value assignment h $[h \leqslant h']$ is a function such that (a) $h'(p_i) \in \{0, 1\}$ for all propositional variables p_i and extends to all formulas as expected, and (b) for all $p_i \in Var$, if $h(p_i) \in \{0, 1\}$, then $h(p_i) = h'(p_i)$. A supervaluation induced by an assignment h is a function f_h such that for all $\phi \in \mathcal{L}$: (a) $f_h(\phi) = 1$ if and only if for all classical extensions h' of h, $h'(\phi) = 1$; (b) $f_h(\phi) = 0$ if and only if for all classical extensions h' of h, $h'(\phi) = 0$; and (c) $f_h(\phi) = \frac{1}{2}$ otherwise. One interesting aspect of supervaluatinist logic is that it is not compositional. For example, if $h(\phi) = h(\psi) = \frac{1}{2}$, then $f_h(\phi \vee \psi) = 1$ if $\psi = \neg\phi$ but $f_h(\phi \vee \psi) = \frac{1}{2}$ if $\phi = p_1$ and $\psi = p_2$.

The basic idea of gamifying supervaluationist logic involves mapping each formula and assignment pair (ϕ, h) to a *set* of classical games, namely those classical games that correspond to the classical extensions of h.

Proposition 4 (Success for Supervaluation). *Given an arbitrary assignment h and a supervaluational semantics f_h, for all formulas ϕ: (a) $f_h(\phi) = 1$ if and only if* **V** *has a winning strategy in every game in G^h_ϕ, (b) $f_h(\phi) = 0$ if and only if* **F** *has a winning strategy in every game in G^h_ϕ, and (c) $f_h(\phi) = \frac{1}{2}$ otherwise.*

3.2 Intuitionistic Logic

Intuitionistic logic requires no introduction, and I will presume the reader is familiar with the Kripke semantics for intuitionistic logic. The only subtelty that is involved in gamifying the Kripke semantics for intuitionistic logic is that given a structure **K** of partially ordered nodes, the translation function associates to each formula-node pair (ϕ, k) a game $G_{(\phi,k)}$. Once again, the shape of the game depends on the structure provided by the Krike *frame*. The basic idea is that games represent what is for the formula to be true in that node. As an example, consider the usual clauses for the conditional and negation:

- A node k *forces* $\phi \rightarrow \psi$ if, for every $k' \geqslant k$, if k' *forces* ϕ then k' *forces* ψ.
- A node k *forces* $\neg\phi$ if, for **no** $k' \geqslant k$ does k' *forces* ϕ.

Then the translation functions are the following:

- The game corresponding to $(\phi \rightarrow \psi, k)$, $G_{(\phi\rightarrow\psi,k)}$ has a root node that is a move for Falsifier whose children are the games $G_{(\sim\phi\vee\psi,k')}$[4], for all $k' \geqslant k$.
- The game corresponding to $(\neg\phi, k)$, $G_{(\neg\phi,k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi,k')}}$, for all $k' \geqslant k$. Here $\overline{G_{(\phi,k')}}$ is just like $G_{(\phi,k')}$ but with roles and payoffs switched [just like in the classical negation clause].

The next obvious step is to match, for each formula ϕ the set of all games $G_{(\phi,k)}$ for all k in the Kripke structure **K**. In this way we obtain $G^{\mathbf{K}}_\phi$, the set of games corresponding to ϕ in **K**.

Proposition 5 (Success for Intuitionism). *Given an arbitrary Kripke structure* **K***, for all formulas ϕ: (a)* $\mathbf{K} = \mathbf{1}$ *if and only if* **V** *has a winning strategy in every game in $G^{\mathbf{K}}_\phi$.*

[4] Here $\sim$ is just classical negation.

As far as I know, Proposition 7 is new - although it is a natural application of dynamic reasoning.

4 Conclusion and discussion

To gamify a semantics means, intuitively, to provide a game-theoretic representation of it. The purpose of this essay was to clarify different notions of gamification and to study the extension to which different propositional logical systems can be gamifiable. I presented three notions of gamification - weak, basic and general. I argued that several finitely-algebraizable logics *strongly* gamifiable, but it is still open whether all of them are. In the next section I presented a result that shows that all finitely-algebraizable logics are gamifiable. The last section shows that even non-algebraizable and non-compositional logics are easily gamifiable if we relax the condition of uniqueness and allow formulas to be represented as sets of games.

So far, I have not provided any philosophical account of what we learn about a semantics by knowing if it is, or not, gamifiable. That was not the main purpose of the essay, but a few words are worth saying. Hintikka's original motivation to provide a game semantics had to do more with the pragmatic nature of assertion, or the meaning of conditionals, than with purely logical concerns. The purpose here was to advance an approach to logic that is neither semantic nor syntactic, but rather pragmatic. Valuation functions for formulas usually express the truth values that formulas have under some assignment or model *a la* Tarski, so that -in general- the truth value of a compound expression depends in some way on the truth value of its components. When providing game-theoretic semantics, I intended to avoid *alethic* considerations and ideas and substitute them by instrumental, pragmatic or operational concepts. The hope is that furthering this approach will provide us with more insights about the relation between Theoretical Reason - captured in our logical systems - and Practical Rationality - captured in our game and decision theoretic ideas. For a logical system to be gamifiable, then, means that its Theoretical import can be captured strategically.

To conclude, two questions and potential lines of research emerge from here. To begin, it would be interesting to answer whether all algebraizable logics can be strongly gamified. Furthermore, the converse problem for weak gamification is also interesting: Given a class of two-player games closed under some operations and with payoffs in a set V, whether there is a language $\mathcal{L}$ closed under some operations and a matrix-semantics $\mathbf{A}$ with assignment h such that (a) there is a function that translates games into formulas and (b) the (BI, Nash, etc.) solutions of the game correspond in some way to the value that its corresponding formula gets in h.[5]

[5] Notice here that nothing secures uniqueness of solutions for these games, so solving this problem might require generalizing the presented definition of matrix algebra in some way.

A Appendix: Proofs

Success for Strong Kleene. The proof of this proposition is by induction, and it is analogous to the traditional proof of Proposition 1. The atomic case is trivial and guaranteed by the definition of $\mathcal{V}$ in labeled and unlabeled terminal nodes [i.e. literals]. For complex expressions, assuming by Induction Hypothesis that Proposition 2 holds for all the subformulas, we need to ensure that: (a) $\mathcal{V}(G_{\neg\phi}) = \neg^{\mathbf{K_3}}\mathcal{V}(G_\phi)$; (b) $\mathcal{V}(G_{\phi\vee\psi}) = \mathcal{V}(G_\phi) \vee^{\mathbf{K_3}} \mathcal{V}(G_\psi)$; and (c) $\mathcal{V}(G_{\phi\wedge\psi}) = \mathcal{V}(G_\phi) \wedge^{\mathbf{K_3}} \mathcal{V}(G_\psi)$.

In the case of binary connectives we get the identity easily, since the Nash Equilibria are obtained by the Backward Induction procedure and the player's payoffs are such that *Verifier* prefers maximizing between $\mathcal{V}(G_\phi)$ and $\mathcal{V}(G_\psi)$, and *Falsifier* prefers minimizing between those two alternatives. The case for negation requires an observation:

Observation 1 (Mirroring of Pure Strategies and NE).
If s_V is a pure strategy for Verifier *in G_ϕ, then s_V is a pure strategy for* Falsifier *in $G_{\neg\phi}$ [and vice versa]. Furthermore, given some payoff assignment $\mathcal{V}$ to the terminal nodes, if $< s_V, s_F >$ is a Nash Equilibrium in G_ϕ, then $< s_F, s_V >$ is a Nash Equilibrium in $G_{\neg\phi}$.*

A short proof of the observation is the following. Any pure strategy profile $< s_V, s_F >$ in G_ϕ is associated with a terminal node, the one that is reached by the path indicated by the strategies, with some payoffs $(x, 1-x)$ for *Verifier* and *Falsifier* respectively. Also, $< s_F, s_V >$ in $G_{\neg\phi}$ leads to the same node, but now with payoffs $(1-x, x)$. Notice that in $G_{\neg\phi}$ there was a *turn switch*; so $< s_F, s_V >$ is the profile where *Verifier* plays s_F and *Falsifier* plays s_V. If the *second* profile $< s_F, s_V >$ in game $G_{\neg\phi}$ is **not** a Nash Equilibrium, then at least one player, say *Falsifier*, can change the strategy to s'_V so that $< s_F, s'_V >$ terminates in a node with payoff $y > x$ for him [leaving s_F fixed]. But then *Verifier* can change her strategy in G_ϕ to also obtain a better payoff. Obviously, the argument is symmetric, and hence one profile is a Nash Equilibrium if and only if the other is.

With this observation, we get that $\mathcal{V}(G_{\neg\phi}) = 1 - \mathcal{V}(G_\phi)$, which is what we needed.

□

Success for Post. It is easy to see that Post's negation does not satisfy De Morgan's properties in general. Yet, a weaker (partial) form of De Morgan's is satisfied:

Observation 2 (Partial De Morgan's for Post's Logic).

- $h(\sim(\phi \vee \psi)) = \begin{cases} h(\sim\phi \wedge \sim\psi) & \textit{if } h(\phi) = 0 \neq h(\psi) \textit{ or } h(\phi) = 0 \neq h(\psi) \\ h(\sim\phi \vee \sim\psi) & \textit{if otherwise} \end{cases}$
- $h(\sim(\phi \wedge \psi)) = \begin{cases} h(\sim\phi \vee \sim\psi) & \textit{if } h(\phi) = 0 \neq h(\psi) \textit{ or } h(\phi) = 0 \neq h(\psi) \\ h(\sim\phi \wedge \sim\psi) & \textit{if otherwise} \end{cases}$

For simplicity, I avoid this proof here.

Providing a game semantics for Post requires a slight change in methodology. In all the cases presented here, the function **game** : $\mathcal{L}_{\sim} \rightarrow \mathcal{A}$ was defined *independently* of the *evaluation* function $\mathcal{V}$ for the games. In this way we had *game forms.* For disjunctions and conjunctions, nothing different is needed. Yet, in order to provide a game that corresponds to a formula that involves Post's negation, it is necessary to look at the truth values of the components. Notice then that for formulas ϕ *not* involving Post negation, we can rely in the success lemmas already shown, so that $h(\phi) = \mathcal{V} \circ \mathbf{game}(\phi)$. Hence in when defining **game** we can use this fact. Se we only need to show that the success holds for formulas that involve the negation. $\mathbf{game}(\sim \phi)$ is $\mathbf{game}(\phi)$ transformed inductively in the following way:

For terminal nodes, we need to generalize the $\overline{x}$ function that we had before because now 'bars' are cummulative. So we have a function in the exponent that tracks the amount of times terminal nodes were negated. Non negated terminal nodes $< p_i >$ are now replaced by $< p_i^0 >$. Give $\mathbf{game}(\phi)$, we start by replacing each terminal node $< p_i^k >$ with $< p_i^{k+1} >$. As expected, we stipulate that $\mathcal{V}(G_{p_i^k}) = [h(p_i) - k](mod\, n+1) = h(\sim ... \sim (p_i))$ where the negation is iterated k times. If $< \psi >$ is a non-terminal node and has children $< \alpha >$ and $< \beta >$, and if $h(\alpha) = n \neq h(\beta)$ or $h(\beta) = n \neq h(\alpha)$:

$$turn(< \psi >) = \begin{cases} \mathbf{V} & \text{if } turn(< \psi >) = \mathbf{F} \\ \mathbf{F} & \text{if } turn(< \psi >) = \mathbf{V} \end{cases}$$

Otherwise, turns are not changed.

The only thing that is really needed now is to show the case for Post-negated formulas, i.e. that $\mathcal{V}(\mathbf{game}(\sim \phi)) = [\mathcal{V}(\mathbf{game}(\phi)) - 1](mod\, n+1)$. In order to do this we need an observation. Notice that the tree corresponding to $\mathbf{game}(\sim \phi)$ and the tree coresponding to $\mathbf{game}(\phi)$ are the *same*, but the games are different in that turn assignments to non-terminal nodes and exponential markings in terminal nodes might have changed.

Observation 3. *Given trees* $\mathbf{game}(\sim \phi)$ *and* $\mathbf{game}(\phi)$*, for any node* $< \psi >_{\neg\phi}$ *in the former corresponding to a node* $< \psi >_{\phi}$ *in the latter we have that* $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) - 1](mod\, n+1)$.

The proof is by (Backwards) Induction. If $< p_i^k >$ is terminal, then the observation follows by definition. Say $< \psi >$ is a non-terminal node and has children $< \alpha >$ and $< \beta >$ such that $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}) - 1](mod\, n+1)$ and $\mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1](mod\, n+1)$.

Case 1: $\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}) = n \neq \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})$ or $\mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi}) = n \neq \mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi})$. Suppose the former, the latter case is symmetrical. By inductive hypothesis, $\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}) = 0 \neq \mathcal{V} \circ \mathbf{game}(\beta_{\phi})$. If $turn(< \alpha >_{\phi}) = \mathbf{V}$, $\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) = max\{\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\phi})\} = \mathcal{V} \circ \mathbf{game}(\beta_{\phi})$. Also, by definition, $turn(< \alpha >_{\neg\phi}) = \mathbf{F}$, and then $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = min\{\mathcal{V} \circ \mathbf{game}(\alpha_{\neg\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\neg\phi})\} = min\{n, \mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1(mod\, n+1)\} = \mathcal{V} \circ \mathbf{game}(\beta_{\phi}) - 1(mod\, n+1)$. So $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = [\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) - 1](mod\, n+1)$. If $turn(< \alpha >_{\phi}) = \mathbf{F}$, $\mathcal{V} \circ \mathbf{game}(\psi_{\phi}) = min\{\mathcal{V} \circ \mathbf{game}(\alpha_{\phi}), \mathcal{V} \circ \mathbf{game}(\beta_{\phi})\} = 0 = \mathcal{V} \circ \mathbf{game}(\alpha_{\phi})$. Also, by definition, $turn(< \alpha >_{\neg\phi}) = \mathbf{V}$, and then $\mathcal{V} \circ \mathbf{game}(\psi_{\neg\phi}) = max\{\mathcal{V} \circ$

$\mathbf{game}(\alpha_{\neg\phi}), \mathcal{V}\circ\mathbf{game}(\beta_{\neg\phi})\} = max\{n, \mathcal{V}\circ\mathbf{game}(\beta_{\neg\phi})\} = n = \mathcal{V}\circ\mathbf{game}(\alpha_{\neg\phi}) = [\mathcal{V}\circ\mathbf{game}(\alpha_{\phi}) - 1](mod\, n+1)$.

Case 2: Otherwise. Either (i) $\mathcal{V}\circ\mathbf{game}(\alpha_{\neg\phi}) = \mathcal{V}\circ\mathbf{game}(\beta_{\neg\phi}) = n$ or (ii) $\mathcal{V}\circ\mathbf{game}(\alpha_{\neg\phi}) \neq n \neq \mathcal{V}\circ\mathbf{game}(\beta_{\neg\phi})$. This proof is left to the reader for its simplicity. □

Success for Supervaluationism. The main consideration here is to translate each formula and assignment pair (ϕ, h) to a *set* of classical games. This is done in two steps. First, map each formula ϕ to its game *form* without any specified payoffs, using the original translation method. Second, consider in order the propositional letters p_i that appears in the game form and the value that h assigns to p_i. If $h(p_i) \in \{0, 1\}$, then assign the corresponding payoff to p_i and move to the next. If $h(p_i) = \frac{1}{2}$, then split the game into two games, one in which the payoff of p_i is (1,0) and another in which it is (0,1). Continue the procedure with all the games that were generated in the steps before.

It should be clear to the reader that the set of games obtained are the games corresponding to all of the classical extensions of h. Then the proposition follows by mere definition.

□

Success for Intuitionism. Begin with a Kripke structure $\mathbf{K}$ of partially ordered nodes $\{k_i\}_{i\in I}$. Recall that this proof is just for the propositional case; nothing conceptually different is added for the first order case.

There is an atomic forcing relation defined for all nodes k such that for all propositional letters p_i, either k *forces* p_i [i.e. k makes p_i true, or the forcing relation is not defined for that node and propositional letter. This atomic forcing relation is subject to the constrain that if $k \leqslant k'$ and k forces p_i, then k' forces p_i. The extensio of the *forcing* relation to all formulas is the following: (a) A node k *forces* $\phi \wedge \psi$ if it forces ϕ and ψ; (b) A node k hforces $\phi \vee \psi$ if it forces ϕ or ψ; (c) A node k *forces* $\phi \rightarrow \psi$ if, for every $k' \geqslant k$, if k' *forces* ϕ then k' forces ψ; and (d) A node k *forces* $\neg\phi$ if, for **no** $k' \geqslant k$ does k' *forces* ϕ.

Given a formula and a node k in a Kripke structure $\mathbf{K}$, the translation functions are the following: (a) The game corresponding to (p_i, k) is a one node game with payoffs (1,0) if k forces p_i and (0,1) otherwise; (b) The game corresponding to $(\phi \wedge \psi, k)$, $G_{(\phi\wedge\psi,k)}$, consists of a root node for Falsifier with two subgames,$G_{(\phi,k)}$ and $G_{(\psi,k)}$; (c) The one corresponding to disjunction is as expected; (d) The game corresponding to $(\phi \rightarrow \psi, k)$, $G_{(\phi\rightarrow\psi,k)}$, has a root node that is a move for Falsifier whose children are the games $G_{(\sim\phi\vee\psi,k')}$, for all $k' \geqslant k$; and (e) The game corresponding to $(\neg\phi, k)$, $G_{(\neg\phi,k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi,k')}}$, for all $k' \geqslant k$. Here $\overline{G_{(\phi,k')}}$ is just like $G_{(\phi,k')}$ but with roles and payoffs switched [just like in the classical negation clause].

To show the result is sufficies to show that for any pair (ϕ, k), k forces ϕ if and only if Verifier has a winning strategy in $G_{(\phi,k)}$. The atomic case is trivial. So are the cases for conjunction, disjunction and *classical negation*. This is just the same proof as for classical logic. The case for the conditional is slightly more complicated. It is worth noticing is that here $\sim$ refers to classical negation, and hence $\sim \phi \vee \psi$ is just code for the material conditional. We begin with $(\phi \rightarrow \psi, k)$, and suppose k forces $\phi \rightarrow \psi$. Then, in a nutshell, for every $k' \geqslant k$, k' forces $\sim \phi \vee \psi$. But then, by inductive hypothesis, Verifier has a winning strategy in every such $G_{(\sim \phi \vee \psi, k')}$. So it has a winning strategy in $G_{(\phi \rightarrow \psi, k)}$. Suppose Verifier has a winning strategy in $G_{(\phi \rightarrow \psi, k)}$. Then whichever choice Falsifier makes at the root node, Verifier still has a winning strategy. That means that for all $k' \geqslant k$, Verifier has a winning strategy in $G_{(\sim \phi \vee \psi, k')}$. By inductive hypothesis this just means that for every $k' \geqslant k$, if k' *forces* ϕ then k' *forces* ψ. Consider $(\neg \phi, k)$. Suppose k forces $\neg \phi$. Then there is **no** $k' \geqslant k$ such that k' *forces* ϕ. The game $G_{(\neg \phi, k)}$ has a root node that is a move for Falsifier whose children are the games $\overline{G_{(\phi, k')}}$, for all $k' \geqslant k$. Now, by (a minor extension of) Observation 2 - Mirroring of pure strategies and Nash Equilibria -, we know that for all such k' Falsifier has a winning strategy in $\overline{G_{(\phi, k')}}$ if and only if Verifier has a winning strategy in $G_{(\phi, k')}$, and viceversa. We also know, by inductive hypothesis, that Verifier has a winning strategy in $G_{(\phi, k')}$ if and only if k' forces ϕ. Since k forces $\neg \phi$, there is no $k' \geqslant k$ such that Verifier has a winning strategy in $G_{(\phi, k')}$. But all of the $G_{(\phi, k')}$ are two-player perfect information games, and therefore are determined. *Ergo*, Falsifier has a winning strategy in all those $G_{(\phi, k')}$. By Observation 2, Verifier has a winning strategy in all the $\overline{G_{(\phi, k')}}$. Hence, whichever move Falsifier makes in the root node of $G_{(\neg \phi, k)}$, it leads to a game won by Verifier. To conclude, Verifier has a winning strategy for $G_{(\neg \phi, k)}$. Now for the converse. Suppose Verifier has a winning strategy for $G_{(\neg \phi, k)}$. This just means that whatever $\overline{G_{(\phi, k')}}$ Falsifier chooses at the root node, Verifier has a winning strategy there. Therefore, again by Observation 2, Falsifier has a winning strategy in all the $G_{(\phi, k')}$ with $k' \geqslant k$. By inductive hypothesis, this just means that there is no $k' \geqslant k$ such that k' forces ϕ.

□

References

1. Petr Cintula and Ondrej Majer. Towards Evaluation Games for Fuzzy Logics. In Ondrej Majer, Ahti-Veikko Pietarinen, and Tero Tulenheimo, editors, *Games: Unifying Logic, Language, and Philosophy*, volume 15 of *Logic, Epistemology, and the Unity of Science*, pages 117–138. Springer Netherlands.
2. Christian G. Fermüller. Dialogue Games for Many-Valued Logics - an Overview. *Studia Logica*, 90(1):43–68, 2008.
3. Christian G. Fermüller. On matrices, Nmatrices and games. *Journal of Logic and Computation*, 2013.
4. Reiner Hähnle. Advanced Many-Valued Logics. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2 of *Handbook of Philosophical Logic*, pages 297–395. Springer Netherlands.
5. Jaakko Hintikka. *Logic, Language Games, and Information.* Clarendon Press, 1973.
6. Jaakko Hintikka and Gabriel Sandu. Game-Theoretical Semantics, 1997.
7. S. C. Kleene. On Notation for Ordinal Numbers. *The Journal of Symbolic Logic*, 3(4):pp. 150–155, 1938.
8. Kleene S.C. *Introduction to Metamathematics.* Bibliotheca Mathematica. Wolters-Noordhoff, 1952.
9. Łukasiewicz J. and Borkowski L. *Selected works.* Studies in logic and the foundations of mathematics. North-Holland Pub. Co., 1970.
10. Mas-Colell A., Whinston M.D., and Green J.R. *Microeconomic Theory.* Oxford student edition. Oxford University Press, 1995.
11. Daniele Mundici. Ulam's games, Łucasiewicz logic, and AFC*-algebras. *Fundamenta Informaticae*, 18, 1993.
12. Martin J Osborne and Ariel Rubinstein. *A course in game theory.* The MIT press, 1994.
13. Rohit Parikh. D-structures and their semantics. *Notices of the AMS 19*, page A329, 1972.
14. Rohit Parikh. The Logic of Games and its Applications. In *Annals of Discrete Mathematics*, pages 111–140. Elsevier, 1985.
15. Emil L. Post. Introduction to a General Theory of Elementary Propositions. *American Journal of Mathematics*, 43(3):pp. 163–185, 1921.
16. Alasdair Urquhart. Basic Many-Valued Logic. In D.M. Gabbay and F. Guenthner, editors, *Handbook of Philosophical Logic*, volume 2 of *Handbook of Philosophical Logic*, pages 249–295. Springer Netherlands.
17. Johan van Benthem. Logic Games Are Complete for Game Logics. *Studia Logica: An International Journal for Symbolic Logic*, (75):183–203, 2003.
18. Johan van Benthem. *Logic in Games.* MIT Press, 2014.
19. B. C. van Fraassen. Presuppositions: Supervaluations and Free Logic. In K. Lambert, editor, *The Logical Way of Doing Things*, pages 67–92. Yale University Press, 1969.
20. Bas C. van Fraassen. Singular Terms, Truth-Value Gaps, and Free Logic. *The Journal of Philosophy*, 63(17):481–495, 1966.