Towards an Implementation Theory via a Game Logic Approach

Davi Romero de Vasconcelos[1] and Edward Hermann Haeusler[2]

[1] Universidade Federal do Ceará, Quixadá, Brazil
daviromero@ufc.br
[2] Pontifícia Universidade Católica do Rio de Janeiro, Brazil
hermann@inf.puc-rio.br

Abstract. Recently, the general approach of Game Theory, which is to find the set of outcomes (the equilibria) of a solution concept in a game, has been modeled by means of Game Logics, in which a game and a solution concept are related to models and formulas of a Game Logic. Thus, the general problem can be stated as a model-checking problem. Moreover, the set of solutions can be found automatically by using model-checkers of the respective Game Logic. The implementation problem in Game Theory focus on the inverse problem, i.e., finding a game that yields a given set of outcomes as equilibria of a solution concept. In this work, we relate the implementation problem to a model checking problem of an adequate Game Logic. Specifically, we illustrate the relationship between the implementation problem of extensive games with perfect information and the model checking problem of GAL (Game Analysis Logic). As a consequence, we can benefit of the use of model-checkers in order to solve the implementation problem as well. It is worth mentioning that the approach used in this article seems to be adequate for other Game Logics as well.

1 Introduction

The general approach of Game Theory [7] is to find the set of outcomes (the equilibria) of a solution concept in a game. On the other hand, Implementation Theory focus on the inverse problem, i.e., a planner looks for a game that yields a given set of outcomes as equilibria of a solution concept. The planner can design the structure of the game, but he or she cannot control the players' preferences and actions. Typical examples of this approach come from mechanism design and auctions in which a planner has to set the rules of the game, and the players take them literally.

The general approach of Game Theory has been modeled by means of Game Logics [9, 2, 1, 5, 6, 10, 11, 8, 3], in which games and solution concepts are related to as models and formulas of the Game Logics (see Figure 1). So, the general problem is related to a model-checking problem of the Game Logics. Since most of the general approach of Game Theory have been stated as model-checking problems, we can drawn the following conjecture.

Conjecture 1. Let $\mathcal{S}$ be a solution concept of a game G.

$$\textbf{s is in the solution set of } \mathcal{S} \textbf{ of the game G iff } \mathbf{\Gamma_G} \models \alpha_{\mathcal{S}}(\mathbf{s}),$$

where Γ_G and $\alpha_{\mathcal{S}}(s)$ are, respectively, a model for the game G and a formula for the solution concept $\mathcal{S}$ within the solution s, expressed in an adequate Game Logic.

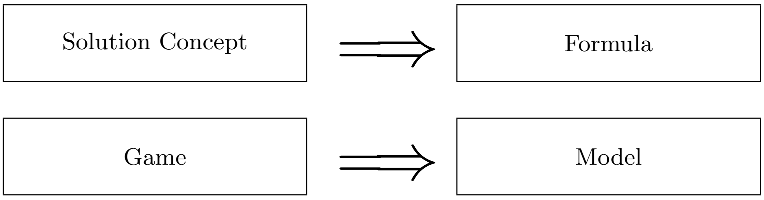


Fig. 1. Relationship between Game Theory and Game Logics.

In [11], we use GAL (Game Analysis Logic) to express the standard games and solution concepts of Game Theory. GAL is a first-order modal logic which is based on the standard logic CTL [4]. The standard models of Game Theory (strategic games, extensive games and coalition games) and their solution concepts (Nash equilibrium, subgame perfect equilibrium and *core*), respectively, are express as models of GAL and formulas of GAL. Moreover, a GAL model-checker is used to find these solution concepts automatically.

In this work we show a correspondence between the implementation problem and the model-checking problem. We restrict attention to the implement problems of extensive games; however, a general approach seems to follow the same baseline.

This work is divided into 4 sections: Section 2 introduces GAL briefly; The correspondence of the general approach and GAL is presented in Section 3; Section 4 states the correspondence between the implementation problem and the model-checking problem; and, Section 5 concludes this work.

2 Game Analysis Logic (GAL)

GAL is a many-sorted modal first-order logic language that is a logic based on the standard Computation Tree Logic (CTL) [4]. A game is a model of GAL, called game analysis logic structure, and an analysis is a formula of GAL.

The *games* that we model are represented by a set of states $\mathcal{S}E$ and a set of actions $\mathcal{CA}$.

A *state* is defined by both a first-order interpretation and a set of players, where: 1- The first-order interpretation is used to represent the choices and the consequences of the players' decisions. For example, we can use a list to represent the history of the players' choices until certain state; 2- The set of players represents the players that have to decide simultaneously at a state. This set must be a subset of the players' set of the game. The other players cannot make a choice at this state. For instance, we can model games such as auction games, where all players are in all states, or even games as Chess or turn-based synchronous game structure, where only a single player has to make a choice at each state. Notice that we may even have some states where none of the players can make a decision that can be seen as states of the nature.

An *action* is a relation between two states e_1 and e_2, where all players in the state e_1 have committed themselves to move to the state e_2. Note that this is an extensional view of how the players committed themselves to take a joint action.

We refer to $(A_k)_{k\in K}$ as a sequence of A_k's with the index $k \in K$. Sometimes we will use more than one index as in the example $(A_{k,l})_{k,l\in K\times L}$. We can also use $(A_k, B_l)_{k\in K,l\in L}$ to denote the sequence of $(A_k)_{k\in K}$ followed by the sequence $(B_l)_{l\in L}$. Throughout of this article, when the sets of indexes are clear in the context, we will omit them.

A *path* $\pi(e)$ is a sequence of states (finite or infinite) that could be reached through the set of actions from a given state e that has the following properties: 1- The first element of the sequence is e; 2- If the sequence is infinite $\pi(e) = (e_k)_{k\in\mathbb{N}}$, then $\forall k \geq 0$ we have $\langle e_k, e_{k+1}\rangle \in \mathcal{CA}$; 3- If the sequence is finite $\pi(e) = (e_0, \ldots, e_l)$, then $\forall k$ such that $0 \leq k < l$ we have $\langle e_k, e_{k+1}\rangle \in \mathcal{CA}$ and there is no e' such that $\langle e_l, e'\rangle \in \mathcal{CA}$. The game behavior is characterized by its paths that can be finite or infinite. Finite paths end in a state where the game is over, while infinite ones represent a game that will never end.

Below we present the formal syntax and semantics of GAL. As usual, we call the sets of sorts S, predicate symbols P, function symbols F and players N as a non-logic language in contrast to the logic language that contains the quantifiers and the connectives. We define a term of a sort in a standard way. We denote a term t of sort s as t_s. The modalities can be read as follows.

- $[EX]\alpha$ - 'exists a path α in the next state'
- $[AX]\alpha$ - 'for all paths α in the next state'
- $[EF]\alpha$ - 'exists a path α in the future'
- $[AF]\alpha$ - 'for all paths α in the future'
- $[EG]\alpha$ - 'exists a path α globally'
- $[AG]\alpha$ - 'for all paths α globally'
- $E(\alpha\mathcal{U}\beta)$ - 'exists a path α until β'
- $A(\alpha\mathcal{U}\beta)$ - 'for all paths α until β'

Definition 1 (Syntax of GAL). *Let $\langle S, F, P, N\rangle$ be a non-logic language, and $t^1_{s_1}, ..., t^n_{s_n}$ be terms, and t'_{s_1} be a term, and $P : s_1...s_n$ be a predicate symbol, and i be a player, and x_s be a variable of sort s. The* ***logic language of GAL*** *is generated by the following BNF definition:*

$$\Phi ::= \top \mid i \mid P(t^1_{s_1}, \ldots, t^n_{s_n}) \mid (t^1_{s_1} \approx t'_{s_1}) \mid (\neg\Phi) \mid (\Phi \to \Phi) \mid \exists x_s \Phi \mid [AX]\Phi \mid E(\Phi\,\mathcal{U}\,\Phi) \mid A(\Phi\,\mathcal{U}\,\Phi)$$

It is well-known that the operators $\wedge, \vee, \bot, [EX], [AF], [EF], [AG], [EG]$ and $\forall x$ can be given by the following usual abbreviations.

- $\alpha \wedge \beta \iff \neg(\alpha \to \neg\beta)$
- $\alpha \vee \beta \iff (\neg\alpha \to \beta)$
- $\bot \iff \neg\top$
- $[EX]\alpha \iff \neg[AX]\neg\alpha$
- $[AF]\alpha \iff A(\top\,\mathcal{U}\,\alpha)$
- $[EF]\alpha \iff E(\top\,\mathcal{U}\,\alpha)$
- $[AG]\alpha \iff \neg E(\top\,\mathcal{U}\,\neg\alpha)$
- $[EG]\alpha \iff \neg A(\top\,\mathcal{U}\,\neg\alpha)$
- $\forall x\alpha(x) \iff \neg\exists x\neg\alpha(x)$

Definition 2 (Structure of GAL). *Let $\langle S, F, P, N\rangle$ be a non-logic language of GAL. A* ***Game Analysis Logic Structure*** *for this non-logic language is a tuple $\mathcal{G} = \langle \mathcal{SE}, \mathcal{SE}_o, \mathcal{CA}, (\mathcal{D}_s), (\mathcal{F}_{f,e}), (\mathcal{P}_{p,e}), (N_e)\rangle$ such that:*

- *$\mathcal{SE}$ is a non-empty set, called the set of states.*
- *$\mathcal{SE}_o$ is a set of initial states, where $\mathcal{SE}_o \subseteq \mathcal{SE}$.*
- *For each state $e \in \mathcal{SE}$, N_e is a subset of N.*
- *$\mathcal{CA} \subseteq \mathcal{SE} \times \mathcal{SE}$, called the set of actions of the game[3], in which if there is at least one player in the state e_1, then exists a state e_2 such that $\langle e_1, e_2\rangle \in \mathcal{CA}$.*
- *For each sort $s \in S$, $\mathcal{D}_s$ is a non-empty set, called the domain of sort s[4].*
- *For each function symbol $f : s_1 \times \ldots \times s_n \to s$ of F and each state $e \in \mathcal{SE}$, $\mathcal{F}_{f,e}$ is a function such that $\mathcal{F}_{f,e} : \mathcal{D}_{s_1} \times \ldots \times \mathcal{D}_{s_n} \to \mathcal{D}_s$.*
- *For each predicate symbol $p : s_1 \times \ldots \times s_n$ of P and state $e \in \mathcal{SE}$, $\mathcal{P}_{p,e}$ is a relation such that $\mathcal{P}_{p,e} \subseteq \mathcal{D}_{s_1} \times \ldots \times \mathcal{D}_{s_n}$.*

A **function or predicate is rigidly interpreted** if its interpretation is the same for every state. A **GAL-structure is finite** if the set of states $\mathcal{SE}$ and each set of domains

[3] This relation is not required to be total as in the CTL case. The idea is because we have finite games.

[4] In algebraic terminology $\mathcal{D}_s$ is a carrier for the sort s.

D_s are finite. Otherwise, it is infinite. Note that even when a GAL-structure is finite we might have infinite paths.

In order to provide the semantics of GAL, we define a valuation function as a mapping σ_s that assigns to each free variable x_s of sort s some member $\sigma_s(x_s)$ of domain $\mathcal{D}_s$. As we use terms, we extend every function σ_s to a function $\bar{\sigma}_s$ from state and term to element of sort s that is done in a standard way. When the valuation functions are not necessary, we will omit them.

Definition 3 (Semantics of GAL). *Let $\mathcal{G} = \langle \mathcal{SE}, \mathcal{SE}_o, \mathcal{CA}, (\mathcal{D}_s), (\mathcal{F}_{f,e}), (\mathcal{P}_{p,e}), (N_e)\rangle$ be a GAL-structure, and (σ_s) be valuation functions, and α be a GAL-formula, where $s \in S, f \in F, p \in P$ and $e \in \mathcal{SE}$.* ***We write $\mathcal{G}, (\sigma_s) \models_e \alpha$ to indicate that the state e satisfies the formula α in the structure $\mathcal{G}$ with valuation functions (σ_s).*** *The formal definition of satisfaction $\models$ proceeds as follows:*

- $\mathcal{G}, (\sigma_s) \models_e \top$.
- $\mathcal{G}, (\sigma_s) \models_e i \iff i \in N_e$
- $\mathcal{G}, (\sigma_s) \models_e p(t^1_{s_1}, ..., t^n_{s_n}) \iff \langle \bar{\sigma}_{s_1}(e, t^1_{s_1}), ..., \bar{\sigma}_{s_n}(e, t^n_{s_n})\rangle \in \mathcal{P}_{p,e}$
- $\mathcal{G}, (\sigma_s) \models_e (t^1_{s_1} \approx t'_{s_1}) \iff \bar{\sigma}_{s_1}(e, t^1_{s_1}) = \bar{\sigma}_{s_1}(e, t'_{s_1})$
- $\mathcal{G}, (\sigma_s) \models_e \neg\alpha \iff$ *NOT* $\mathcal{G}, (\sigma_s) \models_e \alpha$
- $\mathcal{G}, (\sigma_s) \models_e (\alpha \to \beta) \iff$ *IF* $\mathcal{G}, (\sigma_s) \models_e \alpha$ *THEN* $\mathcal{G}, (\sigma_s) \models_e \beta$
- $\mathcal{G}, (\sigma_s) \models_e [AX]\alpha \iff \forall e' \in \mathcal{SE}$ *such that* $\langle e, e'\rangle \in \mathcal{CA}$ *we have* $\mathcal{G}, (\sigma_s) \models_{e'} \alpha$ *(see Figure 2.a).*
- $\mathcal{G}, (\sigma_s) \models_e E(\alpha\ \mathcal{U}\ \beta) \iff$ *exists a finite (or infinite) path* $\pi(e) = (e_0 e_1 e_2 ... e_i)$, *such that exists a k where $k \geq 0$, and $\mathcal{G}, (\sigma_s) \models_{e_k} \beta$, and for all j where $0 \leq j < k$ we have $\mathcal{G}, (\sigma_s) \models_{e_j} \alpha$ (see Figure 2.b).*
- $\mathcal{G}, (\sigma_s) \models_e A(\alpha\ \mathcal{U}\ \beta) \iff$ *for all finite (and infinite) paths such that* $\pi(e) = (e_0 e_1 e_2 ... e_i)$, *exists a k where $k \geq 0$, and $\mathcal{G}, (\sigma_s) \models_{e_k} \beta$, and for all j where $0 \leq j < k$ we have $\mathcal{G}, (\sigma_s) \models_{e_j} \alpha$ (see Figure 2.c).*
- $\mathcal{G}, (\sigma_s, \sigma_{s_k}) \models_e \exists x_{s_k}\alpha \iff$ *exists* $d \in \mathcal{D}_{s_k}$ *such that* $\mathcal{G}, (\sigma_s, \sigma_{s_k}(x_{s_k}|d)) \models_e \alpha$, *where $\sigma_{s_k}(x_{s_k}|d)$ is the function which is exactly like σ_{s_k} except for one thing: At the variable x_{s_k} it assumes the value d. This can be expressed by the equation:*

$$\sigma_s(x_{s_k}|d)(y) = \begin{cases} \sigma_s(y), & \text{if } y \neq x_{s_k} \\ d, & \text{if } y = x_{s_k} \end{cases}$$

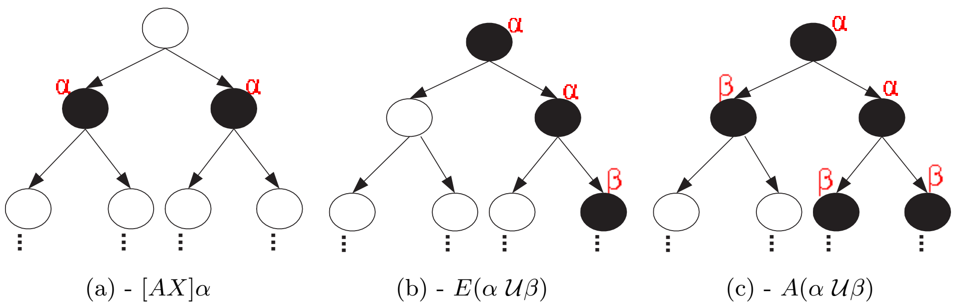


Fig. 2. Modal Connectives of GAL.

3 Game Theory in Game Analysis Logic

In the sequel we write down the definitions of Game Theory (for more details see [7]).

Definition 4. ***An extensive game*** *is a tuple* $\langle N, H, P, g, (\succeq_i)\rangle$, *where*

- N *is a finite set, called the set of* ***players***.
- H *is a set of sequences of actions (finite or infinite), called the set of* ***histories***, *that satisfies the following properties*
  - *the empty sequence is a history, i.e.* $\emptyset \in H$
  - *if* $(a_k)_{k\in K} \in H$ *where* $K \subseteq \mathbb{N}$ *and for all* $l \leq |K|$, *then* $(a_k)_{k=0,\ldots,l} \in H$
  - *if* $(a_0 \ldots a_k) \in H$ *for all* $k \in \mathbb{N}$, *then the infinite sequence* $(a_0 a_1 \ldots) \in H$.

  A history h *is* ***terminal*** *if it is infinite or it has no action* a *such that* $(h, a) \in H$. *We refer to a set of terminals as* $\boldsymbol{T}$.
- $P : H\backslash T \to N$ *is a* ***player function*** *that assigns a player for each nonterminal history.*
- $g : T \to \mathcal{O}$ *is an* ***outcome function*** *that assigns a consequence for each terminal history.*
- *For each player* $i \in N$, *a preference relation* $\succeq_i$ *on* T. *We denote* $\succeq = (\succeq_i)$ *as a* ***preference profile***.

We refer to a tuple $\langle N, H, P, g\rangle$, *which components satisfy the first four conditions in the definition above, as a* ***game form***.

Example 1. An example of a two-player extensive game $\langle N, H, P, g, (\succeq_i)\rangle$, where:

- $N = \{1, 2\}$;
- $H = \{\emptyset, (A), (B), (A, L), (A, R)\}$;
- $P(\emptyset) = 1$ and $P((A)) = 2$;
- $g((B)) = o_1,\ g((A, L)) = o_2,\ g((A, R)) = o_3$.
- $o_2 \succ_1 o_3 \succ_1 o_1$ and $o_3 \succ_2 o_2 \succ_2 o_1$.

A **strategy of player i** is a function that assigns an action for each non-terminal history for each $P(h) = i$. For the purpose of this article, we represent a strategy as a tuple. In order to avoid confusing when we refer to the strategies or the histories, we use '$\langle$' and '$\rangle$' to the strategies and '(' and ')' to the histories. In Example 1, Player 1 has to make a decision only after the initial state and he or she has two strategies $\langle A\rangle$ and $\langle B\rangle$. Player 2 has to make a decision after the history (A) and he or she has two strategies $\langle L\rangle$ and $\langle R\rangle$. We denote $\mathbf{S_i}$ as the set of player i's strategies. We denote $\mathbf{s} = (\mathbf{s_i})$ as a **strategy profile**. We refer to $\mathbf{O}(\mathbf{s_1}, \ldots, \mathbf{s_n})$ as an outcome that is the terminal history when each player follows his or her strategy s_i. In Example 1, $\langle\langle B\rangle, \langle L\rangle\rangle$ is a strategy profile in which Player 1 chooses B after the initial state and Player 2 chooses L after the history (A), and $O(\langle B\rangle, \langle L\rangle)$ is the outcome (B). In a similar way, we refer to $\mathbf{O_h}(\mathbf{h}, \mathbf{s_1}, \ldots, \mathbf{s_n})$ as the outcome when each player follows his or her strategy s_i from history h. In Example 1, $O_h((A), \langle B\rangle, \langle L\rangle)$ is the outcome (A, L) and $g(O_h((A), \langle B\rangle, \langle L\rangle)) = g((A, L)) = o_1$.

We can model an extensive game $G = \langle N, H, P, g, (\succeq_i)\rangle$ as a GAL-structure in the following way. Each history $h \in H$ (from the extensive game) is represented by a state, in which a 0-ary symbol h designates a history of G (the one that the state is coming from), so h is a non-rigid designator. The set of the actions of the GAL-structure is determined by the set of actions of each history, i.e., given a history $h \in H$ and an action a such that $(h,a) \in H$, then the states namely h and (h,a) are in the set of actions of the GAL-structure, i.e. $\langle h,(h,a)\rangle \in \mathcal{CA}$. Function P determines the player that has to make a choice at every state, i.e. $N_h = \{P(h)\}$. The preference profiles and the outcome function g are rigidly defined as in the extensive game. The initial state is the state represented by the initial history of the extensive game, i.e. $H_o = \{\emptyset\}$. Sorts H and T are interpreted as the histories and terminal histories of the extensive game, respectively, i.e., $\mathcal{D}_H = H$ and $\mathcal{D}_T = T$. Sort $\mathcal{O}$ represents the possible outcomes of the extensive game[5], and, for each outcome $o \in \mathcal{O}$ (of the game G), we add a symbol $o :\to \mathcal{O}$, which is rigidly interpreted as the outcome o. In order to define the solution concepts of extensive games that we will define below, we add to this structure the sets of players' strategies ($\mathcal{D}_{S_i}$), functions O and O_h, and a predicate (of type $\in: H \times H$) that states if a history h precedes a history (h,a). To summarize, a **GAL-structure for an extensive game with perfect information** $G = \langle N,P,H,g,(\succeq_i)\rangle$ is the tuple $\langle H, H_o, \mathcal{CA}, (\mathcal{D}_H,\mathcal{D}_T,\mathcal{D}_{S_i},\mathcal{D}_\mathcal{O}), (o_k, g, h_h, O, O_h), \ (\succeq_i, \in), (N_h)\rangle$ with non-logic language $\langle (H,T,S_i,\mathcal{O}), (o_k :\to \mathcal{O}, g : T \to \mathcal{O}, h :\to H, O : S \to T, O_h : H \times S \to T), (\succeq_i\colon \mathcal{O}\times\mathcal{O}, \in\colon H \times H), N\rangle$, where $i \in \{1,\ldots,n\}$ (n is the number of players), $h \in H$, and $k \in \{1,\ldots,K\}$ (K is the number of outcomes). The example below is the GAL-structure (see Figure 3.b) of Example 1 (see Figure 3.a).

Example 2. The GAL-structure of Example 1 is $\langle H, H_o, \mathcal{CA}, (\mathcal{D}_H,\mathcal{D}_T,\mathcal{D}_{S_1},\mathcal{D}_{S_2},\mathcal{D}_\mathcal{O}),$ $(o_1,o_2,o_3,h_h,g,O,O_h), \ (\succeq_1,\succeq_2,\in) \ , (N_h)\rangle$ with non-logic language $\langle (H,T,S_1,S_2,\mathcal{O}),$ $(o_1 :\to \mathcal{O}, o_2 :\to \mathcal{O}, o_3 :\to \mathcal{O}, \ h :\to H, g : T \to \mathcal{O}, O : S_1 \times S_2 \to T, O_h : H \times S_1 \times S_2 \to T), (\succeq_1\colon \mathcal{O}\times\mathcal{O}, \ \succeq_2\colon \mathcal{O}\times\mathcal{O}, \in\colon H\times H), \{1,2\}\rangle$ where

- $H = \{\emptyset, \ (A), \ (B), \ (A,L), \ (A,R)\}$ and $H_o = \{\emptyset\}$.
- $\mathcal{CA} = \{\langle\emptyset, \ (A)\rangle, \ \langle\emptyset,(B)\rangle, \ \langle(A),(A,L)\rangle, \ \langle(A),(A,R)\rangle\}$.
- $\mathcal{D}_{S_1} = \{\langle A\rangle, \langle B\rangle\}$, $\mathcal{D}_{S_2} = \{\langle L\rangle, \langle R\rangle\}$ and $\mathcal{D}_\mathcal{O} = \{o_1,o_2,o_3\}$.
- $\mathcal{D}_H = \{\emptyset, \ (A), \ (B), \ (A,L), \ (A,R)\}$ and $\mathcal{D}_T = \{(B), \ (A,L), \ (A,R)\}$.
- $h_\emptyset = \emptyset$, $h_{(A)} = (A)$, $h_{(B)} = (B)$, $h_{(A,L)} = (A,L)$, $h_{(A,R)} = (A,R)$.
- $N_\emptyset = \{1\}$, $N_{(A)} = \{2\}$, $N_{(B)} = N_{(A,L)} = N_{(A,R)} = \{\}$.
- Functions O, O_h, and g are rigidly defined as in the extensive game.
- $\emptyset \in (A)$, $\emptyset \in (B)$, $(A) \in (A,L)$ and $(A) \in (A,R)$.

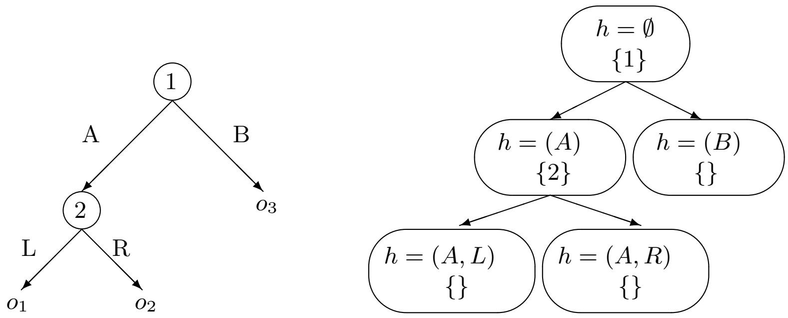
The most used solution concepts for extensive games are Nash equilibrium (NE) and subgame perfect equilibrium (SPE). The solution concept of NE requires that each player's strategy be optimal, given the other players' strategies. And, the solution concept of SPE requires that the action prescribed by each player's strategy be optimal, given the other players' strategies, after every history. In SPE concept, the structure of the extensive game is taken into account explicitly, while, in the solution concept of NE, the structure is taken into account only implicity in the definition of the strategies. Below we present the SPE definition in a standard way. The NE definition below regards to the structure of an extensive game, yet is an equivalent one to the standard.

Definition 5. *A* ***subgame perfect equilibrium (SPE)*** *of an extensive game with perfect information $\langle N,H,P,g,(\succeq_i)\rangle$ is a strategy profile $s^* = \langle s_1^*,\ldots,s_n^*\rangle$ such that for every player $i \in N$ and every history $h \in H$ for which $P(h) = i$ we have*

$$g(O_h(h,s_1^*,\ldots,s_n^*)) \succeq_i g(O_h(h,s_1^*,\ldots,s_i,\ldots,s_n^*)),$$

for every strategy $s_i \in S_i$.

[5] Note that this set is finite if the game is finite.



(a) - Extensive form representation (b) - A GAL representation

Fig. 3. *Mapping an extensive game into a GAL model.*

Definition 6. *A **Nash equilibrium (NE)** of an extensive game with perfect information $\langle N, H, P, g, (\succeq_i)\rangle$ is a strategy profile $s^* = \langle s_1^*, \ldots, s_n^*\rangle$ such that for every player $i \in N$ and every history on the path of the strategy profile s^* (i.e. $h \in O(s^*)$) for which $P(h) = i$ we have*

$$g(O_h(h, s_1^*, \ldots, s_n^*)) \succeq_i g(O_h(h, s_1^*, \ldots, s_i, \ldots, s_n^*)),$$

for every strategy $s_i \in S_i$.

We invite the reader to verify that the strategy profiles $\langle\langle A\rangle, \langle R\rangle\rangle$ and $\langle\langle B\rangle, \langle L\rangle\rangle$ are the Nash equilibria in Example 1. Game theorists can argue that the solution $\langle\langle B\rangle, \langle L\rangle\rangle$ is not reasonable when the players regard to the sequence of the actions. To see that the reader must observe that after the history (A) there is no way for Player 2 commit himself or herself to choose L instead of R since he or she will be better off choosing R (he or she prefers o_2 instead of o_1). Thus, Player 2 has an incentive to deviate from the equilibrium, so this solution is not a subgame perfect equilibrium. On the other hand, we invite the reader to verify that the solution $\langle\langle A\rangle, \langle R\rangle\rangle$ is the only subgame perfect equilibrium.

Consider Formulas 1 and 2 as expressing subgame perfect equilibrium definition 5 and Nash equilibrium definition 6, respectively. A strategy profile $s^* = \langle s_1^*, \ldots, s_n^*\rangle$ is a SPE (or NE) if and only if Formula 1 (or Formula 2) holds at the initial state $\emptyset$, where each $\sigma_{S_i}(v_{s_i}^*) = s_i^*$.

$$[AG]\left(\bigwedge_{i\in N} i \rightarrow \forall v_{s_i}\left(g(O_h(h, v_{s_1}^*, \ldots, v_{s_n}^*)) \succeq_i g(O_h(h, v_{s_1}^*, \ldots, v_{s_i}, \ldots, v_{s_n}^*))\right)\right) \tag{1}$$

$$[EG]\left(\begin{array}{c} h \in O(v_{s_1}^*, \ldots, v_{s_n}^*) \quad \wedge \\ \left(\bigwedge_{i\in N} i \rightarrow \forall v_{s_i}\left(g(O_h(h, v_{s_1}^*, \ldots, v_{s_n}^*)) \succeq_i g(O_h(h, (v_{s_1}^*, \ldots, v_{s_i}, \ldots, v_{s_n}^*)))\right)\right)\end{array}\right) \tag{2}$$

In order to guarantee the correctness of the representation of both subgame perfect equilibrium and Nash equilibrium, we state the theorem below. For the proof of theorem below, see [11, 12]. Thus, the general problem of finding the equilibria of an extensive game according to NE and SPE can be stated as a model-checking problem of GAL. Moreover, the set of equilibria of a finite extensive game can be automatically found by using a model-checker such as the GALV model-checker.

Theorem 1. *Let G be an extensive game, and Γ_G be a GAL-structure for Γ, and α be a subgame perfect equilibrium formula as defined in Equation 1, and β be a Nash equilibrium formula as defined in Equation 2, and (s_i^*) be a strategy profile, and (σ_{S_i}) be valuations functions for sorts (S_i).*

- *A strategy profile (s_i^*) is a SPE of $G \iff \Gamma_G, (\sigma_{S_i}) \models_\emptyset \alpha$, where each $\sigma_{S_i}(v_{s_i}^*) = s_i^*$*
- *A strategy profile (s_i^*) is a NE of $G \iff \Gamma_G, (\sigma_{S_i}) \models_\emptyset \beta$, where each $\sigma_{S_i}(v_{s_i}^*) = s_i^*$*

4 Implementation Theory in Game Analysis Logic

In the sequel we write down the definitions of Implementation Theory (for more details, see [7]), provide an example and state the correspondence of the implementation problem and the model-checking problem in GAL.

Definition 7. ***An environment*** *is a tuple $\langle N, \mathcal{O}, \mathcal{P}, \mathcal{G}\rangle$, where*

- *N is a finite set of* ***players;***
- *$\mathcal{O}$ is a set of* ***feasible outcomes;***
- *$\mathcal{P}$ is a set of* ***preference profiles;***
- *$\mathcal{G}$ is a set of* ***extensive game forms*** *with outcomes in $\mathcal{O}$.*

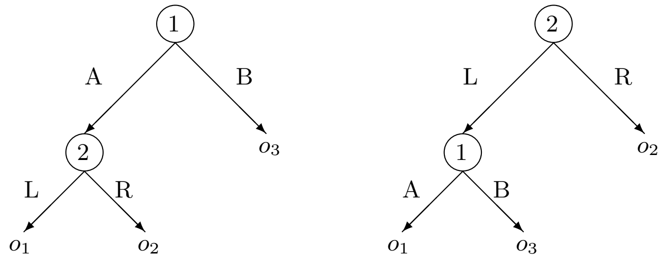
Definition 8. *A* ***choice rule*** *$f : \mathcal{P} \to 2^{\mathcal{O}}$ is a function that assigns a subset of $\mathcal{O}$ to each preference profile $\succeq \in \mathcal{P}$.*

Definition 9 (The Implementation Problem). *A planner has to provide a game form $G \in \mathcal{G}$ for an environment $\mathcal{E} = \langle N, \mathcal{O}, \mathcal{P}, \mathcal{G}\rangle$ and a choice rule f such that, for every preference profile $\succeq \in \mathcal{P}$, the outcomes of a solution concept $\mathcal{S}$ of game $\langle G, \succeq\rangle$ is the set $f(\succeq)$. In this case the game form G is said to* ***$\mathcal{S}$-implement*** *the choice rule f for the environment $\mathcal{E}$. The choice rule f is said* ***$\mathcal{S}$-implementable*** *in $\mathcal{E}$.*

Example 3. A simple example of an environment is a tuple $\langle N, \mathcal{O}, \mathcal{P}, \mathcal{G}\rangle$, where

- $N = \{1, 2\}$.
- $\mathcal{O} = \{o_1, o_2, o_3\}$.
- $\mathcal{P} = \{\succeq\}$, where $o_2 \succ_1 o_3 \succ_1 o_1$ and $o_3 \succ_2 o_2 \succ_2 o_1$.
- $\mathcal{G} = \{G_1, G_2\}$, where G_1 and G_2 are depicted in Figure 4.a and Figure 4.b, respectively.

In the above example, since outcome o_2 is the only outcome of subgame perfect equilibrium (SPE) of game $\langle G_1, \succeq\rangle$, game G_1 is said to SPE-implement the choice rule $f'(\succeq) = \{o_2\}$ for this environment. Thus, this choice rule is SPE-implementable in this environment. On the other hand, game G_2 does not SPE-implement this choice rule in this environment. As another example, games G_1 and G_2 are said to NE-implement the choice rule $f''(\succeq) = \{o_2, o_3\}$.



(a) - Extensive form G_1 (b) - Extensive form G_2

Fig. 4. *Extensive Forms.*

Theorem 2. *Let $\mathcal{E} = \langle N, \mathcal{O}, \mathcal{P}, \mathcal{G}\rangle$ be an environment, G be a game form in $\mathcal{G}$, f be a choice rule, and $\mathcal{S}$ be Nash (or subgame perfect) equilibrium solution concept. The following assertions are equivalent each other.*

1. *A game form G is said to $\mathcal{S}$-implement the choice rule f for the environment $\mathcal{E}$.*
2. *For every $\succeq \in \mathcal{P}$ and for every outcome $o \in \mathcal{O}$ we have*

$$o \in f(\succeq) \Longleftrightarrow \Gamma_{\langle G,\succeq\rangle} \models_\emptyset \exists v^*_{s_1}, \dots, \exists v^*_{s_n} \left(\alpha_{\mathcal{S}} \wedge O(v^*_{s_1}, \dots, v^*_{s_n}) \approx o\right),$$

where $\Gamma_{\langle G,\succeq\rangle}$ is the GAL-model for the game $\langle G, \succeq\rangle$, and $\alpha_{\mathcal{S}}$ is the GAL-formula for the solution concept $\mathcal{S}$.

Proof. *A game form G is said to $\mathcal{S}$-implement the choice rule f for the environment $\mathcal{E}$.*
$\Longleftrightarrow_{def}$ *For every preference profile $\succeq \in \mathcal{P}$,*

the outcomes of a solution concept $\mathcal{S}$ of game $\langle G, \succeq\rangle$ is the set $f(\succeq)$.

$\Longleftrightarrow$ *For every preference profile $\succeq \in \mathcal{P}$,*

$o \in f(\succeq) \Leftrightarrow o$ *is an outcome of the solution concept $\mathcal{S}$ of game $\langle G, \succeq\rangle$.*

$\Longleftrightarrow$ *For every preference profile $\succeq \in \mathcal{P}$,*

$o \in f(\succeq) \Leftrightarrow$ *a strategy profile $\langle s^*_1, \dots, s^*_n\rangle$ is a $\mathcal{S}$ of game $\langle G, \succeq\rangle$ and $o = O(s^*_1, \dots, s^*_n)$.*

By Theorem 1 and since the function O and the outcome o are rigidly interpreted in $\Gamma_{\langle G,\succeq\rangle}$ as in the game $\langle G, \succeq\rangle$, we have
$\Longleftrightarrow$ *For every preference profile $\succeq \in \mathcal{P}$,*

$o \in f(\succeq) \Leftrightarrow \Gamma_{\langle G,\succeq\rangle}, (\sigma_{S_i}) \models_\emptyset \alpha_{\mathcal{S}}$, *and* $\Gamma_{\langle G,\succeq\rangle}, (\sigma_{S_i}) \models_\emptyset O(v^*_{s_1}, \dots, v^*_{s_n}) \approx o$, *where each* $\sigma_{S_i}(v^*_{s_i}) = s^*_i$.

$\Longleftrightarrow$ *For every preference profile $\succeq \in \mathcal{P}$,*

$o \in f(\succeq) \Leftrightarrow \Gamma_{\langle G,\succeq\rangle}, (\sigma_{S_i}) \models_\emptyset \alpha_{\mathcal{S}} \wedge O(v^*_{s_1}, \dots, v^*_{s_n}) \approx o$,
where each $\sigma_{S_i}(v^*_{s_i}) = s^*_i$.

$\Longleftrightarrow$ *For every preference profile $\succeq \in \mathcal{P}$,*

$$o \in f(\succeq) \Leftrightarrow \Gamma_{\langle G,\succeq\rangle} \models_\emptyset \exists v^*_{s_1}, \dots, \exists v^*_{s_n} \left(\alpha_{\mathcal{S}} \wedge O(v^*_{s_1}, \dots, v^*_{s_n}) \approx o\right).$$

5 Conclusion

In this work, we have illustrated that the general approach of Game Theory can be rephrased as as model-checking problem of an adequate Game Logic. Specifically, we have stated the problem of finding the solution concepts of Nash equilibrium and subgame perfect equilibrium of extensive games as a model-checking problem of *Game Analysis Logic*. Moreover, we can also find these solutions automatically via a model-checker (e.g. the GALV model-checker). Nevertheless, the main contribution of this article is to relate the implementation problem with the model-checking problem. We have shown that the problem for extensive games can be stated as a model-checking problem of GAL as well. As a consequence, we can benefit of the use of model-checkers in order to solve the implementation problem as well. It is worth mentioning that the approach used in this article seems to be adequate for other Game Logics as well.

References

1. Thomas Ågotnes, Michael Wooldridge, and Wiebe van der Hoek. On the logic of coalitional games. In P. Stone and G. Weiss, editors, *AAMAS '06:Proceedings of the Fifth International Conference on Autonomous Agents and Multiagent Systems*, pages 153–160, Hakodate, Japan, May 2006. ACM Press.
2. Giacomo Bonanno. Modal logic and game theory: two alternative approaches. *Risk, Decision and Policy*, 7(03):309–324, November 2002.
3. Adam Brandenburger and Robert Aumann. Epistemic Conditions for Nash Equilibrium. *Econometrica*, 63:1161–1180, 1995.
4. Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching-time logic. In *Workshop on Logic of Programs*, pages 52–71. Springer, may 1981.
5. B. P. Harrenstein, W. van der Hoek, J-J. Ch. Meyer, and C. Witteveen. A modal interpretation of Nash-equilibria and some related concepts. In *Proceedings of the LOFT4 conference*, Torino, Italy, 2000.
6. Paul Harrenstein, John-Jules Meyer, W. van der Hoek, and Cees Witteveen. A modal characterization of Nash equilibrium. *Fundamenta Informaticae*, 57(2-4):281–321, 2003.
7. M. J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.
8. M. Pauly. *Logic for Social Software*. Illc dissertation series 2001-10, University of Amsterdam, 2001.
9. Wiebe van der Hoek, Wojciech Jamroga, and Michael Wooldridge. A logic for strategic reasoning. In *AAMAS '05: Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 157–164, New York, NY, USA, 2005. ACM Press.
10. Sieuwert van Otterloo, Wiebe van der Hoek, and Michael Wooldridge. Preferences in game logics. In *AAMAS '04: Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 152–159, Washington, DC, USA, 2004. IEEE Computer Society.
11. D. R. Vasconcelos. *Lógica Modal de Primeira-ordem para Raciocinar sobre Jogos (in Portuguese)*. PhD thesis, PUC-Rio, April 2007.
12. D. R. Vasconcelos and E. H. Haeusler. Reasoning about Games via a First-order Modal Model Checking Approach. In *SBMF'07 – 10th Brazilian Symposium on Formal Methods*, pages 144–159, Ouro Preto, MG, Brazil, 2007.