



**HAL**  
open science

# Epistemic Reasoning About Rationality and Bids in Auctions

Munyque Mittelman, Andreas Herzig, Laurent Perrussel

► **To cite this version:**

Munyque Mittelman, Andreas Herzig, Laurent Perrussel. Epistemic Reasoning About Rationality and Bids in Auctions. *Logics in Artificial Intelligence: 17th European Conference, JELIA 2021, Virtual Event, May 17–20, 2021, Proceedings*, 12678, Springer International Publishing, pp.116-130, 2021, Lecture Notes in Computer Science book series, 978-3-030-75774-8. 10.1007/978-3-030-75775-5\_9. hal-03573581

**HAL Id: hal-03573581**

**<https://hal.science/hal-03573581>**

Submitted on 16 Feb 2022

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Epistemic Reasoning About Rationality and Bids in Auctions

Munyque Mittelmann<sup>1(✉)</sup>, Andreas Herzig<sup>2</sup>, and Laurent Perrussel<sup>1</sup>

<sup>1</sup> Université de Toulouse - IRIT, Toulouse, France

{munyque.mittelmann, andreas.herzig, laurent.perrussel}@irit.fr

<sup>2</sup> Université de Toulouse - IRIT-CNRS, Toulouse, France

**Abstract.** In this paper, we investigate strategic reasoning in the context of auctions. More precisely, we establish an explicit link between bidding actions and bounded rationality. To do so, we extend the Auction Description Language with an epistemic operator and an action choice operator and use it to represent a classical auction where agents have imperfect information about other agents' valuations. We formalize bounded rationality concepts in iterative protocols and show how to use them to reason about the players' actions. Finally, we provide a model checking algorithm.

**Keywords:** Logics for multi-agent systems · Game description language · Bounded rationality · Auction-based markets

## 1 Introduction

Building a General Auction Player is similar to the General Game Playing (GGP) challenge [9], it aims at designing an agent that can participate in an auction while discovering the rules governing it. As for games, there is a wide variety of auction-based markets. Auctions may differ in the participants' type (e.g., only buyers, both buyers and sellers, ...), the kind and amount of goods being auctioned, the bidding protocol, and the allocation and payment rules [13].

Inspired by the Game Description Language (GDL), which is a logic programming language for representing and reasoning about game rules [9], we defined a general language to describe auction-based markets from the auctioneer perspective [15]: *Auction Description Language* (ADL). In this paper, we consider the player's perspective and our goal is to show how an agent may reason about the rules governing an auction and also about their knowledge of other agents' valuations for eliciting her bid. More precisely, we show that computing a rational bid requires to assume that other agents are also bidding rationally. Following [2], we understand 'rational' as 'not playing dominated actions'.

This research is supported by the ANR project AGAPE ANR-18-CE23-0013 and by the EU project TAILOR (EU Horizon 2020 program, GA No 952215).

[https://doi.org/10.1007/978-3-030-75775-5\\_9](https://doi.org/10.1007/978-3-030-75775-5_9)

Our contribution is twofold. We first extend ADL with knowledge operators from Epistemic GDL [12] and the action modality from the GDL variant proposed in [21]. This extension aims at providing the ground for the design of General Auction Players. Second, we characterize rationality along two dimensions: (i) the impact of the level of higher-order knowledge about other agents and (ii) the impact of looking-ahead beyond the next action to be executed. We also explore the complexity of model-checking for evaluating rationality.

*Related Work.* To the best of our knowledge, there is no contribution that focuses on the strategic dimension of auctions through a logical perspective. However, numerous contributions define logical systems for representing games and representing strategic reasoning. GGP uses the Game Description Language (GDL) [9] for representing games. The Auction Description Language (ADL) [15] extends GDL by handling numerical variables, a key feature for representing an auction mechanism with its allocation and payment rules.

Alternating-time Temporal Logic (ATL) [1] provides a logic-based analysis of strategic decisions. Strategy Logic (SL) generalizes ATL with first-order quantifications over strategies [4]. These approaches cannot model the internal structures of strategies, which makes it difficult to easily design strategies aiming to achieve a goal state. A logic for reasoning about composite strategies in turn-based games is introduced in [17], where strategies are treated as programs that are combined by PDL-like connectives. Zhang and Thielscher [22] present a variant of GDL to describe game strategies, where strategies can be understood as moves for a player. However, their work can only model turn-based games.

To incorporate imperfect information games, GDL has been extended to GDL-II [18] and GDL-III [19]. GDL-II and GDL-III aim at describing the rules of an imperfect information game, but do not provide tools for reasoning about how a player infers information based on these rules. All these logics face decidability and tractability issues: their expressive power prevents them from being implemented realistically in an artificial agent. Jiang et al. [12] propose an epistemic extension of GDL (EGDL) to represent and reason about imperfect information games. Their language allows us to represent the rules in the imperfect information setting. A key characteristic of EGDL is that it manages the balance between expressiveness and computational complexity of model checking ( $\Delta_2^P$ ).

Epistemic Game Theory (EGT) considers strategic reasoning with uncertain information and is about the interplay between knowledge, belief and rationality [3, 14]. More precisely, EGT shows how dominated strategies may be eliminated in an iterative manner [2]. These contributions however require perfect reasoners, who can reason about higher-order knowledge at arbitrary depth, which is unrealistic. In [5], the authors abandon this hypothesis but do not propose a full logic detailing the impact of bounded rationality.

*Structure of the Paper.* The remainder of the paper proceeds as follows. In Sect. 2, we define the models of E-ADL in terms of State-Transition structures. In Sect. 3 we present the language and its semantics and illustrate our approach

by describing a Dutch auction. In Sect. 4 we show how to express bounded rationality with higher-order knowledge. In Sect. 5 we present the model-checking algorithm. Section 6 concludes the paper.

## 2 Auctions as State-Transition Models

In this section, we introduce a logical framework for reasoning about auction protocols while considering imperfect information. The framework is based on ADL [15] and Epistemic GDL [11]. We call the framework *Epistemic Auction Description Language*, denoted by E-ADL.

**Definition 1.** *An auction signature  $\mathcal{S}$  is a tuple  $(\mathbb{N}, \mathbb{V}, \mathcal{A}, \Phi, \mathbb{Y})$ , where: (i)  $\mathbb{N} = \{1, 2, \dots, n\}$  is a nonempty finite set of agents; (ii)  $\mathbb{V} \subset \mathbb{Z}$  is a finite subset of integer numbers representing the range of valuations, bids and payments; (iii)  $\mathcal{A} = \bigcup_{r \in \mathbb{N}} A^r$ , where each  $A^r$  consists of a nonempty finite set of actions performed by agent  $r \in \mathbb{N}$  and  $A^r \cap A^s = \emptyset$  if  $r \neq s$ . For convenience, we may write  $a^r$  for denoting an action in  $A^r$ ; (iv)  $\Phi = \{p, q, \dots\}$  is a finite set of atomic propositions for specifying individual features of a state; (v)  $\mathbb{Y} = \{y_1, y_2, \dots\}$  is a finite set of numerical variables for specifying numerical features of a state.*

We assume a total order among the agents in  $\mathbb{N}$ , denoted by  $\prec$ , where  $r \prec i$  means that agent  $r$  precedes agent  $i$  in  $\prec$ ; it will be used to break ties in winner determination. Throughout the rest of the paper, we fix an auction signature  $\mathcal{S}$  and all concepts will be based on this signature, except if stated otherwise. We adopt a semantics based on state-transition models. This is more suitable for describing the dynamics than stable models that were initially considered for GDL and GGP [9].

**Definition 2.** *A state transition ST-model  $M$  is a tuple  $(W, I, T, \{R_r\}_{r \in \mathbb{N}}, U, \pi_\Phi, \pi_Y)$ , where: (i)  $W$  is a finite nonempty set of states; (ii)  $I \subseteq W$  is a set of initial states; (iii)  $T \subseteq W \setminus I$  is a set of terminal states; (iv)  $R_r \subseteq W \times W$  is an equivalence relation for agent  $r$ , indicating the states that are indistinguishable for  $r$ ; (v)  $U : W \times (\prod_{r \in \mathbb{N}} A^r) \rightarrow W$  is an update function, specifying the transitions for each combination of joint actions; (vi)  $\pi_\Phi : W \rightarrow 2^\Phi$  is the valuation function for the state propositions; and (vii)  $\pi_Y : W \times \mathbb{Y} \rightarrow \mathbb{V}$  is the valuation function for the numerical variables.*

For a group of agents  $G \in 2^{\mathbb{N}} \setminus \{\emptyset\}$ , we write  $d^G \in \prod_{r \in G} A^r$  to denote a joint action of the agents in  $G$ . We denote by  $d^r$  the individual action for agent  $r \in G$  in the joint action  $d^G$ . When  $G = \mathbb{N}$  then we omit  $\mathbb{N}$  and simply write  $d$  instead of  $d^{\mathbb{N}}$ . Let  $R_r(w)$  denote the set of all states that agent  $r$  cannot distinguish from  $w$ , i.e.,  $R_r(w) = \{u \in W : wR_r u\}$ .

For every  $w \in W$  and  $d \in \prod_{r \in \mathbb{N}} A^r$ , we call  $(w, d)$  a *move*. Given a group of agents  $G \in 2^{\mathbb{N}} \setminus \{\emptyset\}$ , we write  $(w, \langle d^G, d^{-G} \rangle)$  instead of  $(w, d)$  when we want to talk about  $G$ 's part in  $(w, d)$ , where  $d^{-G} \in \prod_{s \in \mathbb{N} \setminus G} A^s$  denotes the actions of all the agents except those in  $G$  in the joint action  $d$ . Our notion of move resembles the turn-based definition proposed in [21] and [22].

**Definition 3.** Two moves  $(w, d)$  and  $(u, e)$  are equivalent for agent  $r$ , written  $(w, d) \approx_r (u, e)$ , iff  $wR_r u$  and  $d^r = e^r$ .

Clearly relation  $\approx_r$  is reflexive, transitive and symmetric. Differently from standard GDL, our semantics is based on moves instead of paths. This allows the agent to reason about the effects of actions without exploring all ways the game could proceed (i.e., all the reachable states in each complete path where she takes this action). In E-ADL, we define the action execution modality in games with synchronous moves. The idea of move-based semantics and action modalities stems from [21]. Their approach is restricted to turn-based games, where only one action can be performed at a given state.

### 3 Epistemic Auction Description Language

The *Epistemic Auction Description Language* (E-ADL) is a framework to allow epistemic reasoning for auction players. First, we introduce the syntax.

#### 3.1 Syntax

Let  $z \in \mathcal{L}_z$  be a numerical term defined as follows:  $z ::= t \mid add(z, z) \mid sub(z, z) \mid min(z, z) \mid max(z, z) \mid times(z, z) \mid y$ , where  $t \in \mathbb{V}$ ,  $y \in Y$ . The meaning of numerical terms is the natural one; for instance, the term  $min(z_1, z_2)$  specifies the minimum value between  $z_1$  and  $z_2$ . Finally,  $y$  denotes the value of the variable  $y \in Y$  in the current state.

A formula in E-ADL, denoted  $\varphi \in \mathcal{L}_{\text{E-ADL}}$ , is defined by the following BNF:

$$\varphi ::= p \mid z \otimes z \mid r \prec r \mid \text{initial} \mid \text{terminal} \mid \text{does}(a^r) \mid \neg\varphi \mid \varphi \wedge \varphi \mid K_r\varphi \mid [d^G]\varphi$$

where  $p \in \Phi$ ,  $r \in \mathbb{N}$ ,  $\otimes \in \{>, <, =\}$ ,  $a^r \in \mathcal{A}$ ,  $G \in 2^{\mathbb{N}} \setminus \{\emptyset\}$ ,  $d^G \in \prod_{r \in G} A^r$  and  $z \in \mathcal{L}_z$ . Other connectives  $\vee, \rightarrow, \leftrightarrow, \top$  and  $\perp$  are defined by  $\neg$  and  $\wedge$  in the standard way. The comparison operators  $\leq, \geq$  and  $\neq$  are defined by  $\vee, >, <$  and  $=$ . The extension of the operators  $>, <$  and  $=$  and numerical terms  $max(z_1, z_2), min(z_1, z_2), add(z_1, z_2)$  to multiple arguments is straightforward. The formula  $r_1 \prec r_2$  denotes the tie-breaking priority of  $r_1$  over  $r_2$ .

Intuitively, *initial* and *terminal* specify the initial and the terminal states, respectively; *does*( $a^r$ ) asserts that agent  $r$  takes action  $a^r$  at the current move. The epistemic operator  $K_r$  is taken from the Epistemic Logic [7]. The formula  $K_r\varphi$  is read as “agent  $r$  knows that  $\varphi$ ”. The action execution operator comes from the GDL variant with action modalities [21] and the formula  $[d^G]\varphi$  means that if joint action  $d^G$  is executed,  $\varphi$  will be true next. The abbreviation *does*( $d^G$ ) specifies that each agent in  $G$  performs her respective action in  $d^G$ , that is,  $\text{does}(d^G) \stackrel{\text{def}}{=} \bigwedge_{r \in G} \text{does}(d^r)$ . As in [21], we use the action modality to define the temporal operator  $\bigcirc$ :

$$\bigcirc\varphi =_{\text{def}} \bigvee_{d \in \prod_{r \in \mathbb{N}} A^r} (\text{does}(d) \wedge [d]\varphi)$$

The formula  $\bigcirc\varphi$  reads “ $\varphi$  will be true next”. We also use the following abbreviation from Epistemic Logic:  $\widehat{K}_r\varphi =_{def} \neg K_r\neg\varphi$  where  $\widehat{K}_r\varphi$  represents that “ $\varphi$  is compatible with agent  $r$ ’s knowledge”. Given  $j > 0$  and  $G \in 2^N \setminus \{\emptyset\}$ , we write  $\sigma^G = (\prod_{r \in G} A^r)^j$  for a sequence of joint actions for  $G$ . The  $i$ -th joint action in  $\sigma^G$  is noted  $\sigma_i^G$ . Finally, define  $[\sigma^G]^j\varphi$ , for  $|\sigma^G| = j$  by induction of  $j$ :

$$\begin{aligned} [\sigma^G]^1\varphi &\stackrel{def}{=} [\sigma^G]\varphi \\ [\sigma^G]^{j+1}\varphi &\stackrel{def}{=} [\sigma^G][\sigma_j^G]\varphi \end{aligned}$$

The formula  $[\sigma^G]^j\varphi$  means that if the group  $G$  followed the sequence of joint actions described by  $\sigma^G$  for the next  $j$  stages, then  $\varphi$  would hold.

### 3.2 Semantics

The semantics for E-ADL is given in two steps. First, function  $f$  interprets the meaning of numerical terms  $z \in \mathcal{L}_z$ . Next, a formula  $\varphi \in \mathcal{L}_{E-ADL}$  is interpreted with respect to a move. In Definition 4, we specify function  $f$  to evaluate the meaning of any  $z \in \mathcal{L}_z$  in a move.

**Definition 4.** Let  $M$  be an ST-Model. Define Function  $f : W \times (\prod_{r \in N} A^r) \times \mathcal{L}_z \rightarrow \mathbb{Z}$ , assigning any  $w \in W$ ,  $d \in \prod_{r \in N} A^r$ , and  $z \in \mathcal{L}_z$  to a number in  $\mathbb{Z}$ :

If  $z$  is on the form  $add(z', z'')$ ,  $sub(z', z'')$ ,  $min(z', z'')$ ,  $max(z', z'')$  or  $times(z', z'')$ , then  $f(w, d, z)$  is defined through the application of the corresponding mathematical operators and functions over  $f(w, d, z')$  and  $f(w, d, z'')$ . Otherwise,  $f(w, d, z) = z$  if  $z \in V$  and  $f(w, d, z) = \pi_Y(w, z)$  if  $z \in Y$ .

**Definition 5.** Let  $M$  be an ST-Model. Given a move  $(w, d)$ , where  $w \in W$  and  $d \in \prod_{r \in N} A^r$ , and a formula  $\varphi \in \mathcal{L}_{ADL}$ , we say that  $\varphi$  is true in the move  $(w, d)$  under  $M$ , denoted by  $M \models_{(w,d)} \varphi$ , according to the following rules:

$$\begin{aligned} M \models_{(w,d)} p &\quad \text{iff } p \in \pi_\Phi(w) \\ M \models_{(w,d)} \neg\varphi &\quad \text{iff } M \not\models_{(w,d)} \varphi \\ M \models_{(w,d)} \varphi_1 \wedge \varphi_2 &\quad \text{iff } M \models_{(w,d)} \varphi_1 \text{ and } M \models_{(w,d)} \varphi_2 \\ M \models_{(w,d)} \text{initial} &\quad \text{iff } w \in I \\ M \models_{(w,d)} \text{terminal} &\quad \text{iff } w \in T \\ M \models_{(w,d)} r_1 \prec r_2 &\quad \text{iff } r_1 \prec r_2 \\ M \models_{(w,d)} \text{does}(a^r) &\quad \text{iff } d^r = a^r \\ M \models_{(w,d)} z_1 \otimes z_2 &\quad \text{iff } f(w, d, z_1) \otimes f(w, d, z_2), \text{ where } \otimes \in \{>, <, =\} \\ M \models_{(w,d)} K_r\varphi &\quad \text{iff for every } u \in W \text{ and } e \in \prod_{s \in N} A^s, \text{ if } (w, d) \approx_r (u, e), \\ &\quad \text{then } M \models_{(u,e)} \varphi \\ M \models_{(w,d)} [b^G]\varphi &\quad \text{iff } M \models_{(U(w,e),c)} \varphi, \text{ where } e = \langle b^G, d^{-G} \rangle, \text{ for every} \\ &\quad c \in \prod_{r \in N} A^r \end{aligned}$$

A formula  $\varphi$  is *globally true* in an ST-Model  $M$ , written  $M \models \varphi$ , if  $M \models_{(w,d)} \varphi$  for all  $w \in W$  and  $d \in \prod_{r \in N} A^r$ . Finally, let  $\Sigma$  be a set of formulas in  $\mathcal{L}_{E-ADL}$ , then  $M$  is a *model* of  $\Sigma$  if  $M \models \varphi$  for all  $\varphi \in \Sigma$ .

Each  $K_r$  is a normal modal operator. It satisfies that if all  $r$ -accessible worlds agree on  $\varphi$  then  $r$  knows either  $\varphi$  or  $\neg\varphi$ . If  $\varphi$  is true then  $r$  knows that  $\varphi$ .



**Proposition 1.** *Let  $M$  be an ST-Model,  $r \in N$  be an agent and  $\varphi \in \mathcal{L}_{E-ADL}$  be a formula, then  $M \models \varphi \rightarrow K_r \varphi$  if and only if for all  $w, u \in W$  and all  $d, e \in \prod_{r \in N} A^r$  such that  $(w, d) \approx_r (u, e)$ ,  $M \models_{(w,d)} \varphi$  iff  $M \models_{(u,e)} \varphi$ .*

It follows from the equivalence relation  $\approx_r$  that agent  $r$  knows the actions she performs. This is similar to the uniform strategies in Alternating-time Temporal Epistemic Logic [10] and Dynamic Epistemic Logic [20].

**Proposition 2.** *For any agent  $r \in N$ , action  $a^r \in A^r$ , formula  $\varphi \in \mathcal{L}_{E-ADL}$ , number of steps  $j > 0$ , group of agents  $G \in 2^N \setminus \{\emptyset\}$  and  $\sigma^r \in (\prod_{r \in G} A^r)^j$ :*

1.  $M \models \text{does}(a^r) \rightarrow K_r \text{does}(a^r)$
2. If  $M \models [\sigma^G]^j \varphi$  then  $M \models K_r [\sigma^G]^j \varphi$
3. If  $M \models [\sigma^G]^j K_r \varphi$  then  $M \models K_r [\sigma^G]^j \varphi$

Let us now illustrate how to represent an auction-based protocol in E-ADL, namely, a Dutch auction. First, we show the syntactical representation through E-ADL-formulas. Later, we address the semantical representation.

### 3.3 Running Example: Dutch Auction

In a Dutch auction, the auctioneer starts by proposing a high asking price. The price is decreased until it reaches a predefined reserve price or some bidder shows interest at purchasing the good. The auction then ends and the object is sold at the given price to the bidder who signaled her interest [13].

Let  $\mathcal{S}_{\text{dut}}$  be an auction signature and  $\text{starting}, \text{reserve} \in \mathbb{N}$ ,  $\text{dec}, \mathbf{n} \in \mathbb{N} \setminus \{0\}$  be constant values. The constants  $\text{starting}, \text{reserve}, \text{dec}, \mathbf{n}$  represent the starting and reserve prices, the decrement in each round and the number of agents, respectively. The auction signature is defined as follows:  $\mathcal{S}_{\text{dut}} = (\mathbb{N}_{\text{dut}}, \mathbb{V}_{\text{dut}}, \mathcal{A}_{\text{dut}}, \Phi_{\text{dut}}, \mathbb{Y}_{\text{dut}})$ , where  $\mathbb{N}_{\text{dut}} = \{1, \dots, \mathbf{n}\}$ ,  $\mathbb{V}_{\text{dut}} = \{0, \dots, \text{starting}\}$ ,  $\mathcal{A}_{\text{dut}} = \{\text{bid}_r, \text{wait}_r : r \in \mathbb{N}_{\text{dut}}\}$ ,  $\Phi_{\text{dut}} = \{\text{winner}_r : r \in \mathbb{N}\}$  and  $\mathbb{Y}_{\text{dut}} = \{\text{payment}_r, \vartheta_r : r \in \mathbb{N}\}$ . The numerical variables  $\text{payment}_r$  and  $\vartheta_r$  specify the payment and the private valuation for an agent  $r$ .

**Syntactical Representation.** The rules of the Dutch auction are formulated by E-ADL-formulas as shown in Fig. 1.

In an initial state, the price starts at  $\text{starting}$  and there is no winner (Rule 1). If an agent is a winner, she pays the current price. Otherwise, she does not pay anything (Rules 2 and 3). The terminal state is reached when it is not possible to decrease the price anymore or there is a winner (Rule 4). While not in the terminal state, the price either decreases if no agent bids or the price is settled if some agent accepted to purchase the good (Rules 5 and 6). If only one agent accepts, she is marked as the winner. In case two or more agents bid, the winner is assigned according to the tie-breaking rule. Rules 7 and 8 ensure no proposition or numerical variable change its value after a terminal state. Finally, Rule 9 specifies that each agent is aware of how much she values the good. Let  $\Sigma_{\text{dut}}$  be the set of Rules 1–9.

1.  $initial \leftrightarrow price = starting \wedge \bigwedge_{r \in N_{dut}} \neg winner_r$
2.  $winner_r \rightarrow payment_r = price$ , for each  $r \in N_{dut}$
3.  $\neg winner_r \rightarrow payment_r = 0$ , for each  $r \in N_{dut}$
4.  $terminal \leftrightarrow sub(price, dec) < reserve \vee \bigvee_{r \in N_{dut}} winner_r$
5.  $\neg terminal \wedge price = x \wedge \bigwedge_{r \in N_{dut}} does(wait_r) \rightarrow \bigcirc(price = sub(price, dec) \wedge \bigwedge_{r \in N_{dut}} \neg winner_r)$ , for each  $x \in V_{dut}$
6.  $\neg terminal \wedge price = x \wedge does(bid_r) \wedge \bigwedge_{s \neq r, s \in N_{dut}} (\neg does(bid_s) \vee r \prec s) \rightarrow \bigcirc(winner_r \wedge \bigwedge_{s \neq r, s \in N_{dut}} \neg winner_s)$ , for each  $x \in V_{dut}$  and each  $r \in N_{dut}$
7.  $terminal \wedge y = x \rightarrow \bigcirc y = x$ , for each  $y \in Y_{dut}$  and each  $x \in V_{dut}$
8.  $terminal \wedge win \rightarrow \bigcirc win$ , for each  $win \in \{winner_r, \neg winner_r : r \in N_{dut}\}$
9.  $K_r(\vartheta_r = x) \vee K_r \neg(\vartheta_r = x)$ , for each  $x \in V_{dut}$  and  $r \in N_{dut}$

**Fig. 1.** Dutch auction represented by  $\Sigma_{dut}$

**Model Representation.** Let us address the model representation of the Dutch auction. Let us define  $\mathcal{M}_{dut}$  as the class of models  $M_{dut}$  defined for a signature  $\mathcal{S}_{dut}$  and the constants  $starting, reserve, dec$  and  $n$ . Each  $M_{dut} = (W_{dut}, I_{dut}, T_{dut}, \{R_{r,dut}\}_{r \in N}, U_{dut}, \pi_{\Phi,dut}, \pi_{Y,dut})$  is defined as follows:

- $W_{dut} = \{\langle pr, buyer, val_1, \dots, val_n \rangle : 0 \leq pr \leq starting \ \& \ buyer \in N_{dut} \cup \{none\} \ \& \ 0 \leq val_r \leq starting \text{ for each } r \in N_{dut}\}$ ;
- $I_{dut} = \{\langle starting, none, val_1, \dots, val_n \rangle : 0 \leq val_r \leq starting \text{ for each } r \in N_{dut}\}$ ;
- $T_{dut} = \{\langle pr, buyer, val_1, \dots, val_n \rangle : 0 \leq pr \leq starting \ \& \ buyer \in N_{dut} \ \& \ 0 \leq val_r \leq starting \text{ for each } r \in N_{dut}\} \cup \{\langle pr, buyer, val_1, \dots, val_n \rangle : pr - dec < reserve \ \& \ buyer \in N_{dut} \cup \{none\} \ \& \ 0 \leq val_r \leq starting \text{ for each } r \in N_{dut}\}$ ;
- For each agent  $r \in N_{dut}$  and for any two states  $w = \langle pr, buyer, val_1, \dots, val_n \rangle$  and  $u = \langle pr', buyer', val'_1, \dots, val'_n \rangle$  in  $W_{dut}$ , the relation  $R_{r,dut}$  is defined as follows:  $wR_{r,dut}u$  iff (i)  $pr = pr'$ ; (ii)  $buyer = buyer'$ ; and (iii)  $val_r = val'_r$ .
- For all states  $w = \langle pr, buyer, val_1, \dots, val_n \rangle$  and all joint actions  $d = (a^r)_{r \in N_{dut}}$ , such that  $w \in W_{dut}$  and  $a^r \in \{bid_r, wait_r\}$ , we define  $U_{dut}$  as follows:
  - If  $w \notin T_{dut}$ , then  $U_{dut}(w, d) = \langle pr', buyer', val_1, \dots, val_n \rangle$ , such that the components  $pr'$  and  $buyer'$  are defined as follows: (i)  $pr' = pr - dec$  if  $a^r = wait_r$ , for all  $r \in N_{dut}$ ; otherwise  $pr' = pr$ ; (ii)  $buyer' = r$  if  $a^r = bid_r$  for some  $r \in N$  and for all  $s \in N_{dut}$  such that  $s \neq r$ , either  $a^s = wait_s$  or  $r \prec s$ ; otherwise,  $buyer' = none$ ;
  - Otherwise,  $U_{dut}(w, d) = w$ .
- Finally, for each state  $w = \langle pr, buyer, val_1, \dots, val_n \rangle$ , such that  $w \in W_{dut}$ , let  $\pi_{\Phi,dut}(w) = \{winner_r : buyer = r \ \& \ r \in N_{dut}\}$ ;  $\pi_{Y,dut}(w, price) = pr$ . For each agent  $r \in N_{dut}$ , let  $\pi_{Y,dut}(w, \vartheta_r) = val_r$  and  $\pi_{Y,dut}(w, payment_r) = pr$  if  $buyer = r$ . Otherwise,  $\pi_{Y,dut}(w, payment_r) = 0$ .

Let us assume a model  $M_{dut} \in \mathcal{M}_{dut}$  and  $\Sigma_{dut}$  for some  $\mathcal{S}_{dut}$  and the constants  $starting, reserve \in \mathbb{N}$ ,  $dec, n \in \mathbb{N} \setminus \{0\}$ .

**Proposition 3.**  $M_{dut}$  is an ST-Model and  $M_{dut} \models \Sigma_{dut}$ , i.e.,  $M_{dut}$  is a model of  $\Sigma_{dut}$ .



That is,  $M_{\text{dut}}$  is a sound representation of  $\Sigma_{\text{dut}}$ . Notice that as  $M_{\text{dut}}$  is not the unique model for  $\Sigma_{\text{dut}}$ , thereby, the completeness does not hold. It follows from Proposition 1 and 3 that each agent knows the auction rules denoted by  $\Sigma_{\text{dut}}$ , that is,  $M_{\text{dut}} \models \bigwedge_{r \in \mathbb{N}} (\mathbf{K}_r \Sigma_{\text{dut}})$ . In the next section, we define rationality in E-ADL.

## 4 Rationality in Auctions

To characterize rationality of auction players, we assume  $\{\vartheta_r, \text{payment}_r : r \in \mathbb{N}\} \subseteq Y$  and  $\{\text{winner}_r : r \in \mathbb{N}\} \subseteq \Phi$ , where  $\vartheta_r, \text{payment}_r$  and  $\text{winner}_r$  specify the agents valuation, payment and whether she won the auction, resp. Let  $ut \in V$ , we denote whether the utility of agent  $r \in \mathbb{N}$  is equal to  $ut$  in a single good and unit auction according to the truth value of the following formula:

$$\text{utility}_r = ut \stackrel{\text{def}}{=} (ut = \text{sub}(\vartheta_r, \text{payment}_r) \wedge \text{winner}_r) \vee (ut = -\text{payment}_r \wedge \neg \text{winner}_r)$$

Note that we can extend the notion of utility to multiple units and goods by including numerical variables representing the agents' allocations and their valuations for such allocations. In this work, we focus on epistemic reasoning about action choice and rationality of auction players. For a discussion on expressivity and hierarchy of valuations functions, the reader may refer to Feige et al. [8].

Similar to the strong strategy dominance (see [14]), we say an action  $a^r$  of an agent  $r$  is a *strongly dominated action* if and only if, there exists another action  $b^r$  of  $r$  such that, for all actions  $a^{-r}$  of the other agents, playing  $b^r$  while others play  $a^{-r}$  leads to a better utility than playing  $a^r$  while others play  $a^{-r}$ . In E-ADL, the agents' utility is captured in a move of a model and the action choice operator allows us to compare what would have happened if a group of agents took a given joint action.

### 4.1 Rationality

We adapt the weak rationality formalization from [14] to E-ADL formulas. Different from his approach, we consider levels of rationality instead of common knowledge. Our notion of  $k$ -order rationality is based on [6]: an agent is  $k$ -order rational if she is weakly rational and knows all agents are  $(k - 1)$ -order rational.

GDL-based languages explicit the stages of a game execution through paths (or runs). The game starts from an initial state and the succeeding states are defined according to the agents' joint actions. Since GDL agents choose “on-the-fly strategies” during the game, the players should be able to evaluate the current state of the game and to decide which action they will execute.

Adopting these features from GDL in E-ADL allows us to explicitly model information feedback, which is a key feature in the design of iterative auctions [16]. For instance, in E-ADL, we can describe auctions where the agents are assigned to allocations and payments at any stage, which may be different from

their final assignments in the terminal state. For this reason, instead of defining utilities as a function to strategy profiles as in ATL [1], we model the agents' utility as being dependent on the current state of the auction.

We rephrase the rationality notions from [6,14] by, at first, considering  $k$ -order of knowledge and, second, by taking into account state-based utilities and exploring bounded sequences of actions. A rational agent plays according to her utility after performing an action. When reasoning about iterative auctions, the agent considers her utility after playing according to a sequence of  $j$  actions. Since most auction-based markets are finite (in the sense that the auction finishes eventually), it is reasonable to assume the agents only need to include in their reasoning which actions may occur in the next  $j$  steps. Given a fixed number of steps  $j > 0$ , we inductively define that an agent is  $k$ -order rational, for  $k \leq j$ . The base case is that any agent is 0-order rational, that is,  $\text{Rat}(r, 0, j) \stackrel{\text{def}}{=} \top$ . For all  $k > 0$ , we define:

$$\text{Rat}(r, k + 1, j) \stackrel{\text{def}}{=} \text{WR}(r, j) \wedge \text{K}_r \left( \bigwedge_{s \in \mathbb{N}} \text{Rat}(s, k, j) \right)$$

That is, an agent is  $(k + 1)$ -order rational if she is weakly rational when looking  $j$  stages ahead and knows every other agent is  $k$  rational. Weak rationality is defined by:

$$\text{WR}(r, j) \stackrel{\text{def}}{=} \bigwedge_{a^r \in \mathbb{A}^r} (\text{does}(a^r) \rightarrow \bigvee_{\rho^r \in (\mathbb{A}^r)^{j-1}} \text{WRAction}(r, (a^r, \rho^r), j))$$

where

$$\begin{aligned} \text{WRAction}(r, \sigma^r, j) =_{\text{def}} & \bigwedge_{\chi^r \in (\mathbb{A}^r)^j} \left( \bigvee_{\sigma^{-r} \in (\prod_{s \neq r} \mathbb{A}^s)^j} (\widehat{\text{K}}_r \text{does}(\sigma_1^{-r}) \wedge \right. \\ & \left. \bigvee_{ut, ut' \in \mathbb{V}} ([\chi^r, \sigma^{-r}]^j \text{utility}_r = ut' \wedge [\sigma^r, \sigma^{-r}]^j \text{utility}_r = ut \wedge ut' \leq ut)) \right) \end{aligned}$$

An agent  $a^r$  is weakly rational when reasoning  $j$  stages ahead if when she performs an action  $a^r$ , there exists a sequence of  $j$  actions starting by  $a^r$  that is weakly rational for her to follow over  $j$  stages. Finally, it is weakly rational for agent  $r$  to follow a sequence of actions  $\sigma^r$  for  $j$  steps, noted  $\text{WRAction}(r, \sigma^r, j)$ , if for every other sequence of actions  $\chi^r$  there exists a sequence of joint actions  $\sigma^{-r}$  that  $r$  considers possible to be executed such that her utility after following  $\sigma^r$  for  $j$  steps is at least as good as her utility after following  $\chi^r$ .

Notice that if  $j$  is large enough to reach terminal states, the state-based utilities represent strategy-based utility functions. Our definition of rationality requires to assume that all agents are rational: as soon as one is known to be non-rational, it is no longer possible to be  $k$ -order rational, for  $k > 1$ . This requirement entails that looking ahead without considering knowledge leads to consider all actions as rational:

**Proposition 4.** *For every ST-Model  $M$ , state  $w \in W$ , joint action  $d \in \prod_{r \in \mathbb{N}} \mathbb{A}^r$ , agent  $r \in \mathbb{N}$  and  $j > 0$ , it holds that  $M \models_{(w,d)} \text{does}(d^r) \wedge \text{Rat}(r, 0, j)$ .*

Next, considering higher-order knowledge enables us to eliminate strongly dominated actions.

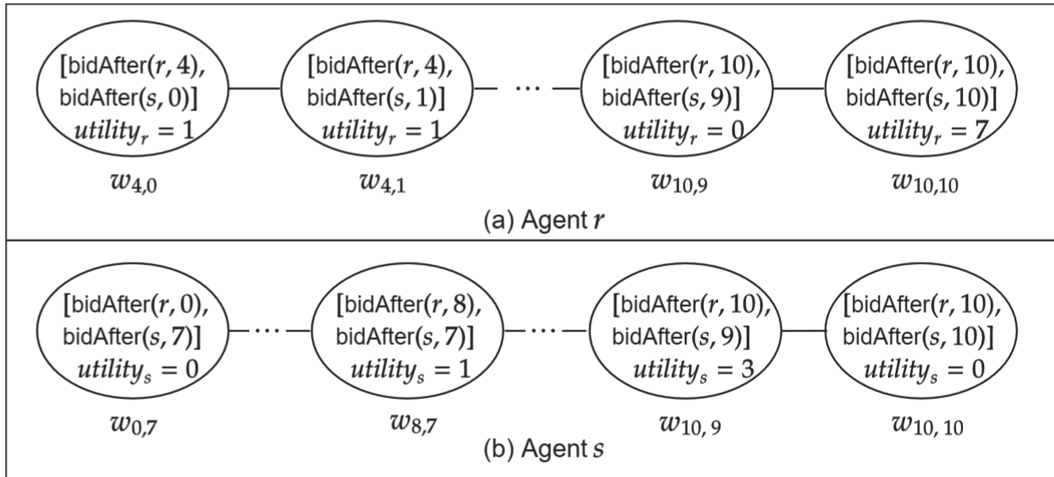
**Theorem 1.** *For any ST-Model  $M$ , state  $w \in W$ , joint action  $d \in \prod_{r \in N} A^r$ ,  $k > 0$ ,  $j > 0$ , agent  $r \in N$  and action  $a^r \in A^r$ , if  $M \models_{(w,d)} \text{does}(a^r) \wedge \text{Rat}(r, k, j)$  then  $M \models_{(w,d)} \text{does}(a^r) \wedge \text{Rat}(r, k - 1, j)$ .*

*Proof.* Assume  $M \models_{(w,d)} \text{does}(a^r) \wedge \text{Rat}(r, k, j)$ . Thus,  $M \models_{(w,d)} \text{does}(a^r) \wedge \text{WR}(r, j) \wedge K_r(\bigwedge_{s \in N} \text{Rat}(s, k - 1, j))$ . Since  $R_r$  is reflexive, it follows that  $M \models_{(w,d)} \text{does}(a^r) \wedge \text{Rat}(r, k - 1, j)$ .

Note that increasing  $j$  may not enable the elimination of actions. The larger  $j$ , the more stages will be considered. Ideally,  $j$  should be large enough to reach terminal states. However, termination may not be ensured in auction protocols and real world players usually have time restrictions to decide their actions.

## 4.2 Example: Rationality on the Dutch Auction

Let us consider the Dutch auction from Sect. 3.3. Consider a specific instance  $M_{\text{dut}}$  in  $\mathcal{M}_{\text{dut}}$ , such that there are only two players  $r$  and  $s$  whose valuation for the good being auctioned is 7 and 4, respectively. The auctioneer starts by proposing the price 10 and in each round the price is decreased by 1. Formally,  $N_{\text{dut}} = \{r, s\}$ ,  $V_{\text{dut}} = \{0, \dots, 10\}$ ,  $\mathcal{A}_{\text{dut}} = \{\text{bid}_r, \text{wait}_r, \text{bid}_s, \text{wait}_s\}$ ,  $\Phi_{\text{dut}} = \{\text{winner}_r, \text{winner}_s\}$  and  $Y_{\text{dut}} = \{\text{payment}_r, \vartheta_r, \text{payment}_s, \vartheta_s\}$ . Let  $M_{\text{dut}}$  be the model defined by the signature  $\mathcal{S}_{\text{dut}} = (N_{\text{dut}}, V_{\text{dut}}, \mathcal{A}_{\text{dut}}, \Phi_{\text{dut}}, Y_{\text{dut}})$  and the constants  $\text{starting} = 10$ ,  $\text{dec} = 1$ ,  $\text{reserve} = 0$  and  $n = 2$ . We consider the initial state  $w_0 \in I$ , such that  $\pi_Y(w_0, \vartheta_r) = 7$  and  $\pi_Y(w_0, \vartheta_s) = 4$ .

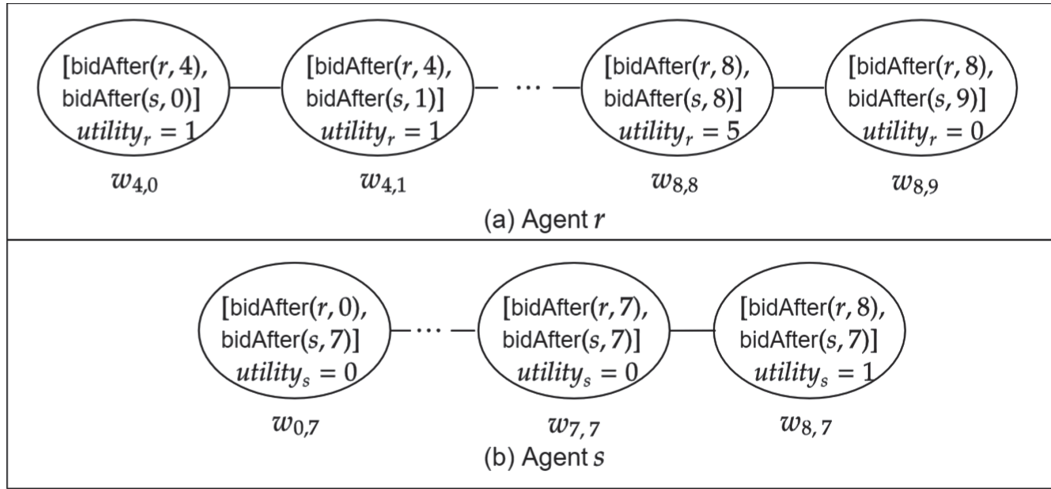


**Fig. 2.** The utilities agents  $r$  and  $s$  consider possible to obtain when they are 1st-order rational

Due to the starting price and the decrement, the auction is ensured to end after 10 stages. We therefore focus on the case  $j = 10$ . If the auction reaches

a terminal state before 10 stages, the update function ensures a loop in the terminal state. Since the auction ends at the first bid, we write  $\text{bidAfter}(r, m)$  as the sequence of actions  $\sigma^r$ , such that  $\sigma_i^r = \text{wait}_r$  for  $i < m \leq j$  and  $\sigma_i^r = \text{bid}_r$  for  $m \leq i \leq j$ . The sequence is read “ $r$  bids after  $m$  steps”. Let  $\text{onlywait}(r)$  be the sequence of  $j$  actions  $\text{wait}_r$ . We use a similar notation for expressing agent  $s$ ’s sequence of actions. Let  $d$  be a joint action, we will examine which sequences of actions are rational for each agent to follow. We assume the Dutch auction protocol  $\Sigma_{\text{dut}}$  and the tie-breaking ordering are common knowledge among the agents in  $\mathbb{N}_{\text{dut}}$ .

If the agents are 0-order rational, that is, if  $M_{\text{dut}} \models_{(w_0, d)} \text{Rat}(r, 0, j) \wedge \text{Rat}(s, 0, j)$ , then both agents consider possible that any sequence of joint actions will be taken. If we now consider 1st-order rationality for  $r$ , that is  $M_{\text{dut}} \models_{(w_0, d)} \text{Rat}(r, 1, j)$ , then  $r$  is not going to follow any sequence of actions that are strongly dominated in  $j$  steps. The weakly rational sequences of actions for  $r$  are those where she waits until the price is below her private valuation (e.g.,  $\text{bidAfter}(r, 4)$ ,  $\text{bidAfter}(r, 5)$ , and so on). The sequence of actions  $\text{onlywait}(r)$  is not rational for  $r$ . The weakly rational actions for agent  $s$  when  $M_{\text{dut}} \models_{(w_0, d)} \text{Rat}(s, 1, j)$  are defined similarly. Figure 2 illustrates the utilities each agent considers possible to achieve when playing a weakly rational sequence of actions.



**Fig. 3.** The utilities agents  $r$  and  $s$  consider possible to obtain when they are 7th-order rational and  $M_{\text{dut}} \models (2 \leq \vartheta_s \leq \text{starting}) \wedge (2 \leq \vartheta_r \leq \text{starting})$

For  $k > 1$ , which actions a  $k$ -order rational agent considers possible her opponents will take depends on her knowledge about their valuations. For instance, let us consider the case where it is common knowledge that  $(2 \leq \vartheta_s \leq \text{starting}) \wedge (2 \leq \vartheta_r \leq \text{starting})$ , i.e., we have  $M_{\text{dut}} \models (2 \leq \vartheta_s \leq \text{starting}) \wedge (2 \leq \vartheta_r \leq \text{starting})$ . By Proposition 1, both agents then know their opponent has a valuation between 2 and the starting price. If the agent  $s$  is 2nd-order rational, she will know the sequence of actions  $\text{onlywait}(r)$  is not weakly rational for  $r$ . Due to the tie-breaking rule, if both agents bid at the same stage, agent  $r$  wins. Thus, agent  $s$

cannot win by waiting for the price to reach zero and it is not weakly rational to perform  $\text{bidAfter}(s, 10)$ . If  $r$  is 3rd-order rational, she knows that  $s$  knows  $\text{onlywait}(r)$  is not rational for her and consequently, that it cannot be the case that  $s$  will  $\text{bidAfter}(s, 10)$ . If the agents are 4th-order rational, they will not consider possible that the good is not sold before the price be zero. Thus, a similar reasoning will happen due tie-breaking when the price is 1. Finally, Fig. 3 illustrates the utilities each agent considers possible when she is 7th-order rational. Since agents are uncertain about which value between 2 and starting represents the valuation of their opponents, raising the order of rationality beyond 7 would not modify the actions they consider possible to be taken by their opponent.

## 5 Model Checking

Now we examine the upper bound of the complexity of deciding whether an E-ADL formula is true with respect to a model and a move. To prove this bound, we provide a model-checking algorithm and analyze its complexity. Let  $\varphi \in \mathcal{L}_{\text{E-ADL}}$  be a formula and  $M = (W, I, T, \{R_r\}_{r \in \mathbb{N}}, U, \pi_\Phi, \pi_Y)$  be an ST-Model over  $\mathcal{S}$ . We say that  $\psi$  is a subformula of  $\varphi$  if either (i)  $\psi = \varphi$ ; (ii)  $\varphi$  is of the form  $\neg\phi$ ,  $K_r\phi$  or  $[d^G]\phi$  and  $\psi$  is a subformula of  $\phi$ ; or (iii)  $\varphi$  is of the form  $\phi \wedge \phi'$  and  $\psi$  is a subformula of either  $\phi$  or  $\phi'$ . Denote  $\text{Sub}(\varphi)$  as the set of all subformulas of  $\varphi$ .

---

### Algorithm 1. $\text{modelCheck}(M, w, d, \varphi)$

---

**Input:** an ST-model  $M = (W, I, T, \{R_r\}_{r \in \mathbb{N}}, U, \pi_\Phi, \pi_Y)$ , a state  $w$  of  $W$ , a joint action  $d \in \prod_{r \in \mathbb{N}} A^r$  and a formula  $\varphi \in \mathcal{L}_{\text{E-ADL}}$ .

**Output:** **true** if  $M \models_{(w,d)} \varphi$ , and **false** otherwise

```

1:  $S \leftarrow \text{Sub}(\varphi)$  ordered by ascending length
2: Let  $\text{isTrue}[1, \dots, |\varphi|]$  be a boolean array initiated with true values
3: for  $i \leftarrow 1$  to  $|\varphi|$  do
4:    $\phi \leftarrow S[i]$ 
5:   switch the formula type of  $\phi$  do
6:     case  $\phi$  is of the form  $\phi' \wedge \phi''$ 
7:        $\text{isTrue}[i] \leftarrow \text{isTrue}[\text{getIndex}(S, \phi')] \wedge \text{isTrue}[\text{getIndex}(S, \phi'')]$ 
8:     case  $\phi$  is of the form  $\neg\phi'$ 
9:        $\text{isTrue}[i] \leftarrow \neg\text{isTrue}[\text{getIndex}(S, \phi')]$ 
10:    case  $\phi$  is atomic
11:       $\text{isTrue}[i] \leftarrow M \models_{(w,d)} \phi$ 
12:    case  $\phi$  is of the form  $[b^G]\phi'$ 
13:       $e^G \leftarrow \langle b^G, d^G \rangle$ 
14:      for each  $c \in \prod_{r \in \mathbb{N}} A^r$  do
15:         $\text{isTrue}[i] \leftarrow \text{isTrue}[i] \wedge \text{modelCheck}(M, U(w, e), c, \phi')$ 
16:    case  $\phi$  is of the form  $K_r\phi'$ 
17:      for each  $u \in R_r(w)$  and each  $e \in \prod_{r \in \mathbb{N}} A^r$  with  $e^r = d^r$  do
18:         $\text{isTrue}[i] \leftarrow \text{isTrue}[i] \wedge \text{modelCheck}(M, u, e, \phi')$ 
19: return  $\text{isTrue}[|\varphi|]$ 

```

---

**Theorem 2.** *The following problem is in  $\mathcal{O}(|W| \times |\mathcal{A}|^m)$ , where  $m = |\mathcal{N}| \times |\varphi|$ : Given an ST-Model  $M$ , a state  $w \in W$ , a joint action  $d \in \prod_{r \in \mathcal{N}} A^r$  and a formula  $\varphi \in \mathcal{L}_{E-ADL}$ , determine whether  $M \models_{(w,d)} \varphi$  or not.*

*Proof.* Algorithm 1, named *modelCheck*, works in the following way: first it gets all subformulas of  $\varphi$  and orders them in a vector  $S$  by ascending length. Thus,  $S(|\varphi|) = \varphi$ , i.e., the position  $|\varphi|$  in  $S$  corresponds to the formula  $\varphi$  itself, and if  $\phi_i$  is a subformula of  $\phi_j$  then  $i < j$ . An induction on  $S$  labels each subformula  $\phi_i$  depending on whether or not  $\phi_i$  is true in  $M$  at the move  $(w, d)$ . If  $\phi_i$  does not have any subformula, its truth value is obtained directly from the model. Since  $S$  is ordered by formulas length, if  $\phi_i$  is either of the form  $\phi' \wedge \phi''$  or  $\neg\phi'$  the algorithm labels  $\phi_i$  according to the label assigned to  $\phi'$  and/or  $\phi''$ . If  $\phi_i$  is of the form  $[b^G]\phi'$  then its label is recursively defined according to  $\phi'$  truth value in the updated state given the joint action  $\langle b^G, d^{-G} \rangle$ , for any joint action to be taken in the next move. Since we compare with every joint action, this is done in an exponential number of steps, based on the size of the set of agents (i.e., according to  $|\mathcal{A}|^n$ , where  $n = |\mathcal{N}|$ ). Finally, the case where  $\phi_i$  is in the form  $K_r\phi'$  is recursively defined according to the truth value of  $\phi'$  in all moves that are equivalent to  $(w, d)$ . Similar to the previous case, since we compare with all possible moves and all states in  $R_r(w) \subseteq W$ , this step is done in an exponential number of steps (i.e., according to  $|W| \times |\mathcal{A}|^n$ , where  $n = |\mathcal{N}|$ ). As Algorithm *modelCheck* visits each subformula at most once, and the number of subformulas is not greater than the size of  $\varphi$ , the algorithm can clearly be implemented in  $\mathcal{O}(|W| \times |\mathcal{A}|^m)$ , where  $m = |\mathcal{N}| \times |\varphi|$ .

It follows that checking agent rationality is exponential in the quantity of agents, the order of rationality and how many rounds are considered.

**Corollary 1.** *Given an ST-model  $M$ , a state  $w \in W$ , a joint action  $d \in \prod_{r \in \mathcal{N}} A^r$ , an agent  $r$ ,  $j > 0$  and  $k > 0$ , the problem of checking whether  $M \models_{(w,d)} \text{Rat}(r, k + 1, j)$  is in  $\mathcal{O}(|W| \times |\mathcal{A}|^{nkj})$ , where  $n = |\mathcal{N}|$ .*

## 6 Conclusion

In this paper, we present *Epistemic Auction Description Language* (E-ADL), a language to allow reasoning about knowledge and action choice in auctions. E-ADL extends ADL with epistemic operators and action modalities. Our goal is to provide the ground for the design of General Auction Players and the characterization of their rational behavior. As in the GGP competition, real world players may have time restrictions to decide their actions. For those scenarios, we explore bounded rationality in relation to the level of higher-order knowledge about other agents and bounded looking-ahead beyond the next state. For future work, we intend to investigate the interplay between agents' bounded rationality and the auctioneer revenue and to generalize the definitions to combinatorial auctions.



## References

1. Alur, R., Henzinger, T.A., Kupferman, O.: Alternating-time temporal logic. *J. ACM (JACM)* **49**(5), 672–713 (2002)
2. Aumann, R.: Backward induction and common knowledge of rationality. *Games Econ. Behav.* **8**, 6–19 (1995)
3. Bonanno, G.: Epistemic foundations of game theory. In: van Ditmarsch, H., Halpern, J.Y., van der Hoek, W., Kooi, B. (eds.) *Handbook of Logics for Knowledge and Belief*, chap. 9, pp. 411–450. College Publications (2015)
4. Chatterjee, K., Henzinger, T.A., Piterman, N.: Strategy logic. *Inf. Comput.* **208**(6), 677–693 (2010)
5. Chen, J., Micali, S.: Leveraging possibilistic beliefs in unrestricted combinatorial auctions. *Games* **7**(32), 83–101 (2016)
6. Chen, J., Micali, S., Pass, R.: Tight revenue bounds with possibilistic beliefs and level-k rationality. *Econometrica* **83**(4), 1619–1639 (2015)
7. Fagin, R., Moses, Y., Halpern, J.Y., Vardi, M.Y.: *Reasoning about Knowledge*. MIT Press, Cambridge (2003)
8. Feige, U., Feldman, M., Immorlica, N., Izsak, R., Lucier, B., Syrgkanis, V.: A unifying hierarchy of valuations with complements and substitutes. In: *Proceedings of AAAI 2015*, pp. 872–878. AAAI Press (2015)
9. Genesereth, M., Thielscher, M.: *General Game Playing*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan & Claypool Publishers, San Rafael (2014)
10. Jamroga, W., van der Hoek, W.: Agents that know how to play. *Fundamenta Informaticae* **63**(2–3), 185–219 (2004)
11. Jiang, G., Perrussel, L., Zhang, D.: On axiomatization of epistemic GDL. In: Baltag, A., Seligman, J., Yamada, T. (eds.) *LORI 2017*. LNCS, vol. 10455, pp. 598–613. Springer, Heidelberg (2017). [https://doi.org/10.1007/978-3-662-55665-8\\_41](https://doi.org/10.1007/978-3-662-55665-8_41)
12. Jiang, G., Zhang, D., Perrussel, L., Zhang, H.: Epistemic GDL: a logic for representing and reasoning about imperfect information games. In: *Proceedings of IJCAI-2016* (2016)
13. Krishna, V.: *Auction Theory*. Academic Press, San Diego (2009)
14. Lorini, E.: A minimal logic for interactive epistemology. *Synthese* **193**(3), 725–755 (2015). <https://doi.org/10.1007/s11229-015-0960-5>
15. Mittelman, M., Perrussel, L.: Auction description language (ADL): a general framework for representing auction-based markets. In: *ECAI 2020*. IOS Press, Santiago de Compostela (2020)
16. Parkes, D.C.: *Iterative Combinatorial Auctions*. Combinatorial Auctions. MIT Press, Cambridge (2006). <https://doi.org/10.7551/mitpress/9780262033428.003.0003>
17. Ramanujam, R., Simon, S.: Dynamic logic on games with structured strategies. In: *Proceedings of KR-2008*, pp. 49–58. AAAI Press (2008)
18. Thielscher, M.: A general game description language for incomplete information games. In: *Proceedings of AAAI 2010*, pp. 994–999 (2010)
19. Thielscher, M.: GDL-III: a description language for epistemic general game playing. In: *Proceedings of IJCAI-2017*, pp. 1276–1282 (2017)

20. Van Benthem, J.: Games in dynamic-epistemic logic. *Bull. Econ. Res.* **53**(4), 219–248 (2001). <https://doi.org/10.1111/1467-8586.00133>
21. Zhang, D., Thielscher, M.: A logic for reasoning about game strategies. In: *Proceedings of AAAI 2015*, pp. 1671–1677. AAAI Press (2015)
22. Zhang, D., Thielscher, M.: Representing and reasoning about game strategies. *J. Philos. Logic* **44**(2), 203–236 (2014). <https://doi.org/10.1007/s10992-014-9334-6>