ECAI 2020
G.D. Giacomo et al. (Eds.)

doi:10.3233/FAIA200172

Auction Description Language (ADL): A General Framework for Representing Auction-Based Markets

Munyque Mittelmann [1] and **Laurent Perrussel** [2]

Abstract. The goal of this paper is to propose a language for representing and reasoning about the rules governing an auction-based market. Such language is at first interest as long as we want to build up digital market places based on auction, a widely used framework for automated transactions. Auctions may differ in several aspects: single or double-side, ascending or descending, single or multi-unit, open cry or sealed-bid, and so on. This variety prevents an agent to easily switch between different (auction-based) markets. The first requirement for building such agents is to have a general language for describing auction-based markets. Second, this language should also allow the reasoning about the key issues of a specific market, namely the allocation and payment rules. To do so, we define a language in the spirit of the *Game Description Language* (GDL): the *Auction Description Language* (ADL) is the first language for describing auctions in a logical framework. In this paper, we illustrate this general dimension by representing two different types of well-known auctions: an English Auction and a Multi-Unit Vickrey Auction. We show the benefit of ADL by deriving properties about these two auction protocols. It also enables us to show in an explicit way what should be assumed about the behavior of a rational bidder.

1 INTRODUCTION

A huge volume of goods and services are sold through auctions [10]. Typically, an auction-based market is described by a set of rules stating what are the available actions to the participants, how the winner is determined, and what price should be paid by the winner. There are variants where multiple winners could be considered or payment may also concern the losers. Actually, an Auction protocol may differ in numerous aspects: single or double-side [14], ascending [4] or descending [9], single or multi-unit goods [1, 16], and so on.

This great variety of auction protocols prevents any autonomous agent to easily switch between different auction-based markets [19]. Having a language for describing auctions from a general perspective is then at first interest. Participants may be able to process the auction definition and, consequently, define their bids wrt. these rules.

The goal of the paper is to propose a general language to describe auction-based markets. The logic is based on the Game Description Language (GDL) which is a logic-based language for representing and reasoning about game rules; GDL is the official language for the *General Game Playing* challenge [6]. We revisit the GDL variant proposed in [15] and define the logical *Auction Description Language*: we allow numerical variables, comparison, and parameters at the opposite of GDL. Handling numerical values is critical for defining the payment and allocation rules.

[1] Université de Toulouse - IRIT, France, munyque.mittelmann@irit.fr
[2] Université de Toulouse - IRIT, France, laurent.perrussel@irit.fr

1.1 Related work

An auction protocol is an allocation procedure. Its main characteristics are (i) it is a centralized procedure, i.e. it has a central authority (the auctioneer), and (ii) it has monetary transfer. This is not always true for an allocation procedure. For instance, a negotiation protocol may be defined in a distributed (decentralized) approach, where the allocation is the result of a sequence of local negotiation steps [3]. On the other hand, a protocol for exchanging goods or services may not be dependant on monetary transfer.

To the best of our knowledge, almost all contributions on the computational representation of auction-based markets focus on their implementation. In [13], the authors propose an assertive and modular definition of an auction market by representing the market as a set of rules. These rules tackle at first the how and when to bid and assume a single-agent perspective. There is no general reasoning as the semantics is an operational one. The language proposed in [19] adopts an assertive perspective: the proposed language allows the representation of a general auction market, but it is too poor for enabling reasoning. In [2], the authors show how a specific auction, namely combinatorial auctions, can be encoded in a logic program. A hybrid approach mixing linear programming and logic programming has been proposed in [12]: the authors focus on sealed-bid auctions and show how qualitative reasoning helps to refine the optimal quantitative solutions. The closest contribution is the *Market Specification Language* [22] based on the *Game Description Language* (GDL) [6]. The proposed language is rich enough for representing an auction through a set of rules and then interpreting an auction-instance with the help of a state-based semantics. However, the main limit is the single-agent perspective.

1.2 Contribution

In this paper, we focus on single-side auctions: that is one seller and multiple buyers or vice versa. The language is general enough for taking care of goods' quantity (single or multi-unit) and whether it is open or not (sealed-bid). We also focus on the auctioneer perspective: how the auction is organized, how the goods are allocated and how to know if the auction is complete.

While different sorts of auction protocols can be expressed in ADL, the bids are represented through the agents' actions. ADL actions are predefined in the auction signature and may have integer parameters representing amount and quantity. However, we believe that the use of bidding languages to generate the action set would allow agents' to bid over goods combinations (bundles), quantities and preferences in an easier and more flexible way. A bidding language is better fitted to combinatorial auctions and its addition to ADL will be investigated for future work.

The paper is organized as follows: we first describe the auction description language and its associated state-transition semantics (Section 2). We then show how ADL can be instantiated to represent two types of single-side auctions: Section 3 focuses on single-side auction and details the specification of an English Auction. Section 4 presents multi-unit auctions with a specification of Vickrey Auction. In both sections, we first present the definition of the auction and then general properties from the auctioneer point of view, namely general properties and properties requiring some rationality assumption about the bidders. To do so, we take advantage of the logical perspective used for specifying auction. Finally, Section 5 concludes the paper and opens some perspective on the future of ADL.

2 AUCTION DESCRIPTION LANGUAGE

In this section, we introduce a logical framework for auction specification. The framework is based in GDLZ [15], a numerical extension of GDL state transition model and language [23]. The resulting framework can consider numerical variables, numerical terms, and payment and allocation structures. We call the framework *Auction Description Language*, denoted ADL.

To describe an auction, we first define an auction signature, that specifies who are the auction participants (the agents), how they bid (the actions for each agent) and what are the aspects that describe each state in the auction (the propositions and numerical variables). We define an auction signature as follows:

Definition 2.1. An *auction signature* $\mathcal{S}$ is a tuple $(N, \mathcal{A}, \Phi, X)$, where:

- $N = \{r_1, r_2, \cdots, r_k\}$ is a nonempty finite set of *agents*;
- $\mathcal{A} = \bigcup_{r \in N} A^r$ where $A^r = \{a^r_1(\bar{z}_1), \cdots, a^r_m(\bar{z}_m)\}$ consists of a nonempty set of *actions* performed by agent $r \in N$, where $\bar{z}_i \in \mathbb{Z}^l$ is a possibly empty tuple of l parameters for the action a^r_i, $i \leq m$ and $l \in \mathbb{N}$. For convenience, we occasionally write a^r_i for denoting an action $a^r_i(\bar{z}_i) \in \mathcal{A}$;
- $\Phi = \{p, q, \cdots\}$ is a finite set of atomic propositions for specifying individual features of an auction state;
- $X = \{x_1, x_2, \cdots, x_n\}$ is a set of numerical variables for specifying numerical features of an auction state.

Given an auction signature, we define the auction protocol through a state transition model, that allows us to represent the key aspects of an auction, at first the allocation and payment rules.

Definition 2.2. A state transition ST-*model* M is a tuple $(W, \bar{w}, T, L, U, g, p, alloc, \pi_\Phi, \pi_\mathbb{Z}, \prec_N)$, where: (i) W is a nonempty set of *states*; (ii) $\bar{w} \in W$ is the *initial* state; (iii) $T \subseteq W$ is a set of *terminal* states; (iv) $L \subseteq W \times \mathcal{A}$ is a *legality* relation, describing the legal actions at each state; (v) $U : W \times \prod_{r \in N} A^r \rightarrow W$ is an *update* function, specifying the transitions for each combination of joint actions; (vi) $g : N \rightarrow 2^W$ is a *goal* function, specifying the winning states for each agent; (vii) $p : W \times N \rightarrow \mathbb{Z}$ is a *payment* function, specifying the payment for each agent in each state; (viii) $alloc : W \times N \rightarrow \mathbb{Z}$ is an *allocation* function, specifying the quantity of goods allocated to each agent in each state; (ix) $\pi_\Phi : W \rightarrow 2^\Phi$ is the valuation function for the state propositions; (x) $\pi_\mathbb{Z} : W \times X \rightarrow \mathbb{Z}$ is the valuation function for the numerical variables; and (xi) $\prec_N$ is a total order on N, defining the tie-breaking priority between two agents.

Given $d \in \prod_{r \in N} A^r$, let $d(r)$ be the individual action for agent r in the joint action d. Let $L(w) = \{a \in \mathcal{A} \mid (w, a) \in L\}$ be the set of all legal actions at state w.

Note that the update function is deterministic, i.e. given a state and a joint action, the update function will lead to a unique state. In an auction, non-deterministic solutions may be used to solve ties during the winner determination. A usual approach in Auction Theory is to assume that if there are ties, each winning bidder has an equal likelihood of being awarded the object, i.e. the winner is chosen randomly [11] (for discussion about the strategic effect of tie-breaking rules in auctions, see [7]). Considering distributional strategies for a wide class of private value auctions, the set of equilibria is invariant to the tie-breaking rule [7].

In this paper, the tie-breaking solution is based on a fixed-order between the agents (the usual approach in computational social choice [5]). Another deterministic solution that could be represented in ADL is to define a special player whose action would determine the winner in a tie (similar to the "random player" in GDL-II [20]). Thereby, the non-deterministic aspects of the winner determination can be easily simulated in ADL.

Definition 2.3. Given an ST-model $M = (W, \bar{w}, T, L, U, g, p, alloc, \pi_\Phi, \pi_\mathbb{Z}, \prec_N)$, a *path* is a sequence of states $\bar{w} \overset{d_1}{\rightarrow} w_1 \overset{d_2}{\rightarrow} \cdots \overset{d_j}{\rightarrow} \cdots$ such that for any $j \geq 1$: (i) $w_0 = \bar{w}$; (ii) $w_j \neq w_0$; (iii) $d_j(r) \in L(w_{j-1})$ for any $r \in N$, (iv) $w_j = U(w_{j-1}, d_j)$; and (v) if $w_{j-1} \in T$, then $w_{j-1} = w_j$.

A path δ is *complete* if $\delta[e] \in T$, for some $e > 0$. After reaching a terminal state $\delta[e]$, for any $e' > e$, $\delta[e'] = \delta[e]$, i.e. no other state is reachable besides $\delta[e]$. Let $\mathcal{P}(M)$ denote the set of all complete paths in M. When M is fixed, we simply write $\mathcal{P}$. Given $\delta \in \mathcal{P}$, the states on δ are called *reachable states*. Let $\delta[j]$ denote the j-th reachable state of δ, $\theta(\delta, j)$ denote the joint action performed at stage j of δ; and $\theta_r(\delta, j)$ denote the action of agent r performed at stage j of δ.

2.1 Syntax

The *language* is denoted by $\mathcal{L}_{ADL}$ and a *formula* φ in $\mathcal{L}_{ADL}$ is defined by the following BNF grammar:

$$\varphi ::= p \mid initial \mid terminal \mid legal(a^r(\bar{z})) \mid does(a^r(\bar{z})) \mid wins(r) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid z > z \mid z < z \mid z = z \mid x(z) \mid r \prec r \mid payment(r, z) \mid allocation(r, z)$$

A number list $\bar{z}$ is defined as: $\bar{z} ::= z \mid z, \bar{z} \mid \varepsilon$. Finally, a numerical term z is defined by $\mathcal{L}_z$, which is generated by the following BNF: $z ::= z' \mid x \mid add(z, z) \mid sub(z, z) \mid min(z, z) \mid max(z, z) \mid times(z, z)$, where $p \in \Phi, r \in N, a^r(\bar{z}) \in \mathcal{A}, z' \in \mathbb{Z}, x \in X$, and ε represents the empty word.

Other connectives $\vee, \rightarrow, \leftrightarrow, \top$ and $\bot$ 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 comparison operators $>, <, =, \leq, \geq, \neq$ and numerical terms $max(z_1, z_2), min(z_1, z_2), add(z_1, z_2)$ to multiple arguments is straightforward.

Intuitively, $initial$ and $terminal$ specify the initial state and the terminal state, respectively; $does(a^r(\bar{z}))$ asserts that agent r takes action a with the parameters $\bar{z}$ at the current state; $legal(a^r(\bar{z}))$ asserts that agent r is allowed to take action a with the parameters $\bar{z}$ at the current state; and $wins(r)$ asserts that agent r wins at the current state. The formula $\bigcirc\varphi$ means "φ holds at the next state". The formulas $z_1 > z_2$, $z_1 < z_2$, $z_1 = z_2$ means that a numerical term z_1 is greater, less and equal to a numerical term z_2, respectively. The formula $x(z)$ asserts the current value for the numerical variable x, i.e. the x variable in X has the value in z. The tie-breaking priority is represented by the formula $r_1 \prec r_2$, i.e. agent r_1 precedes r_2 in the order $\prec_N$. Formulas $payment(r, z_1)$ and $allocation(r, z_2)$ say that agent r must pay the value z_1 and that z_2 goods are allocated to r. The numerical terms $add(z_1, z_2)$, $sub(z_1, z_2)$ and $times(z_1, z_2)$ specify the value obtained by adding, subtracting and multiplying z_2 from z_1, respectively. The terms $min(z_1, z_2)$ and $max(z_1, z_2)$ specify the minimum and maximum value between z_1 and z_2, resp.

2.2 Semantics

The semantics for the ADL language is given in two steps. First, we define function v to define the meaning of numerical terms $z \in \mathcal{L}_z$ in a specified state. Next, a formula $\varphi \in \mathcal{L}_{ADL}$ is interpreted with respect to a stage in a path.

Definition 2.4. Given an ST-model M, a path δ, a stage j and the functions $minimum$ and $maximum$[3] define function $v : \mathcal{L}_z \times W \to \mathbb{Z}$, assigning any $z \in \mathcal{L}_z$ in a state $w \in W$ to a number in $\mathbb{Z}$:

$$v(z, w) = \begin{cases} z & \text{if } z \in \mathbb{Z} \\ \pi_{\mathbb{Z}}(w, z) & \text{if } z \in X \\ v(z', w) + v(z'', w) & \text{if } z = add(z', z'') \\ v(z', w) - v(z'', w) & \text{if } z = sub(z', z'') \\ v(z', w) \times v(z'', w) & \text{if } z = times(z', z'') \\ minimum(v(z', w), v(z'', w)) & \text{if } z = min(z', z'') \\ maximum(v(z', w), v(z'', w)) & \text{if } z = max(z', z'') \end{cases}$$

Definition 2.5. Let M be an ST-Model. Given a complete path δ of M, a stage j on δ, a formula $\varphi \in \mathcal{L}_{ADL}$ and Function v, we say φ is true (or satisfied) at j of δ under M and v, denoted by $M, \delta, j \models \varphi$, according with the following definition:

$$\begin{array}{lll}
M, \delta, j \models p & \text{iff} & p \in \pi_{\Phi}(\delta[j]) \\
M, \delta, j \models \neg\varphi & \text{iff} & M, \delta, j \not\models \varphi \\
M, \delta, j \models \varphi_1 \wedge \varphi_2 & \text{iff} & M, \delta, j \models \varphi_1 \text{ and } M, \delta, j \models \varphi_2 \\
M, \delta, j \models initial & \text{iff} & \delta[j] = \bar{w} \\
M, \delta, j \models terminal & \text{iff} & \delta[j] \in T \\
M, \delta, j \models wins(r) & \text{iff} & \delta[j] \in g(r) \\
M, \delta, j \models payment(r, x) & \text{iff} & x = p(\delta[j], r) \\
M, \delta, j \models allocation(r, x) & \text{iff} & x = alloc(\delta[j], r) \\
M, \delta, j \models r_1 \prec r_2 & \text{iff} & r_1 \prec r_2 \in \prec_N \\
M, \delta, j \models legal(a^r(\bar{z})) & \text{iff} & (a^r(\bar{z})) \in L(\delta[j]) \\
M, \delta, j \models does(a^r(\bar{z})) & \text{iff} & \theta_r(\delta, j) = a^r(\bar{z}) \\
M, \delta, j \models \bigcirc\varphi & \text{iff} & \text{if } j < |\delta|, \text{ then } M, \delta, j+1 \models \varphi \\
M, \delta, j \models z_1 > z_2 & \text{iff} & v(z_1, \delta[j]) > v(z_2, \delta[j]) \\
M, \delta, j \models z_1 < z_2 & \text{iff} & v(z_1, \delta[j]) < v(z_2, \delta[j]) \\
M, \delta, j \models z_1 = z_2 & \text{iff} & v(z_1, \delta[j]) = v(z_2, \delta[j]) \\
M, \delta, j \models x(z) & \text{iff} & v(z, \delta[j]) = \pi_{\mathbb{Z}}(\delta[j], x)
\end{array}$$

A formula φ is globally true through δ, denoted by $M, \delta \models \varphi$, if $M, \delta, j \models \varphi$ for any stage j of δ. A formula φ is *globally true* in an ST-Model M, written $M \models \varphi$, if $M, \delta \models \varphi$ for all complete paths δ in M. Finally, let Σ be a set of formulas in $\mathcal{L}_{ADL}$, then M is a *model* of Σ if $M \models \varphi$ for all $\varphi \in \Sigma$.

[3] Through the rest of this paper, the functions $minimum(a, b, c, \cdots)$ and $maximum(a, b, c, \cdots)$ return the minimum and maximum value between $a, b, c, \cdots \in \mathbb{Z}$, respectively.

Proposition 2.1. *The following problem is in PTIME: Given an ST-model M, a path δ of M, a stage j on δ and a formula $\varphi \in \mathcal{L}_{ADL}$, determine if $M, \delta, j \models \varphi$ or not.*

Proof. To determinate if $M, \delta, j \models \varphi$, we proceed in the following way: For each subformula ϕ of φ, first we evaluate the truth values of all the proper subformulas of ϕ; with these truth values, the truth value of ϕ can be then easily obtained by $\mathcal{L}_{ADL}$ semantics, including for the next operator. As we visit each subformula at most once, and the number of subformulas in the φ is not greater than the size of φ, these procedure can be implemented in a polynomial-time deterministic Turing machine. □

We conclude the section by defining rational players in a ADL model: bidders that play by considering some private value.

Definition 2.6. Given an ST-model M, an agent $r \in N$ is called a *rational player* iff he has a private value $\vartheta_r \in \mathbb{N}$ and for any state $w \in W$, agent r always tries maximize the payoff function $\Pi_r = \vartheta_r - p(w, r)$. Let $N^{\vartheta} = \{r : r \in N \ \& \ r \text{ is a rational player}\}$ be the set of rational players in N.

3 REPRESENTING SINGLE-UNIT AUCTIONS

In this section, we consider the basic type of single-unit auctions. The two most popular variant are the English (ascending) and Dutch (descending) ones. There are both open cry and can be viewed as a sequence of bidding steps. A sealed-bid auction allows only one bid and all bidding values are private while in open cry auctions bidding values are public and multiple sequential bids may be considered [10].

ADL is able to represent several variants of single-unit auctions and hereafter we detail the English Auction (or Ascending-bid Auction). In the second part of the section, we show how we take advantage of $\mathcal{L}_{ADL}$ to express general properties on (English) Auction.

3.1 English Auction

To represent an English Auction with k bidders, we first describe the auction signature, written $\mathcal{S}_{eng}$, as follows:

- $N_{eng} = \{r_1, r_2, \cdots, r_k\}$;
- $\mathcal{A}_{eng} = \bigcup_{r \in N_{eng}} A^r_{eng}$, where $A^r_{eng} = \{accept^r, decline^r\}$, where $accept^r$ and $decline^r$ represents that r accepts and declines the bid offered by the auctioneer, respectively;
- $\Phi_{eng} = \{first, isBidding(r), currWinner(r) : r \in N_{eng}\}$, where *first* says whether it is the first turn, $isBidding(r)$ specifies if player r is still bidding or if r gave up, and $currWinner(r)$ says whether r is the winner candidate if every bidder gives up in the next round;
- $X_{eng} = \{Bid\}$, where Bid represents the current bid value proposed by the auctioneer.

Each instance of an English Auction is specific and is defined with respect to three constant values: $\mathsf{k}, \mathsf{inc} \in \mathbb{N} \setminus \{0\}$ and $\mathsf{startingBid} \in \mathbb{N}$, respectively representing the size of N_{eng}, the increment in each bidding turn and the starting bid value. The rules of an English Auction can be formulated by ADL-formulas as shown Figure 1.

In the initial state, no one is bidding and the starting bid value is defined (Rule 1). In each round, the players can accept to raise the bid or decline and thus give up from the auction (Rules 6 and 7). The propositions and variables are updated to the next turn, where

1. $initial \leftrightarrow first \wedge Bid(\mathsf{startingBid}) \wedge \bigwedge_{r \in N_{eng}} \neg isBidding(r) \wedge \neg currWinner(r)$
2. $terminal \leftrightarrow \neg first \wedge \bigwedge_{r \in N_{eng}} \neg isBidding(r) \vee \bigvee_{r \in N_{eng}} wins(r)$
3. $wins(r) \leftrightarrow (isBidding(r) \vee currWinner(r)) \wedge \bigwedge_{i \neq r \in N_{eng}} \neg isBidding(i)$
4. $payment(r, x) \wedge allocation(r, 1) \leftrightarrow wins(r) \wedge Bid(x)$
5. $payment(r, 0) \wedge allocation(r, 0) \leftrightarrow \neg wins(r)$
6. $legal(accept^r) \leftrightarrow initial \vee isBidding(r)$
7. $legal(decline^r) \leftrightarrow \top$
8. $\bigcirc Bid(add(x, inc)) \leftrightarrow Bid(x) \wedge \bigvee_{r \in N_{eng}} does(accept^r)$
9. $\bigcirc Bid(x) \leftrightarrow Bid(x) \wedge (terminal \vee \bigwedge_{r \in N_{eng}} does(decline^r) \vee \bigvee_{r \in N_{eng}} wins(r))$
10. $\bigcirc isBidding(r) \leftrightarrow \neg does(decline^r) \vee (isBidding(r) \wedge terminal)$
11. $\bigcirc currWinner(r) \leftrightarrow isBidding(r) \wedge \bigwedge_{y \neq r \in N_{eng}} \neg isBidding(y) \vee r \prec y$
12. $\bigcirc \neg first \leftrightarrow \top$

Figure 1. English Auction represented by Σ_{eng}

the bid is raised if at least one bidder accepts it (Rules 8 to 12). If there is only one or none active bidder, the auction ends (Rule 2). The winner is the last bidder to accept or one of the bidders that had accepted before if everyone declines in the current bidding turn (Rule 3). The losers do not pay, while the winner pays the highest value that he accepted (Rule 4 and 5). Let Σ_{eng} be the set of rules 1-12.

Let $\mathcal{M}_{eng}$ be the set of ST-models M_{eng} defined for any k, startingBid and inc. Each $M_{eng} \in \mathcal{M}_{eng}$ is defined as follows:

- $W_{eng} = \{\langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle : currBid \in \mathbb{N}\ \&\ isB_r, init \in \{true, false\}\ \&\ r \in N_{eng}\ \&\ winner \in N_{eng} \cup \{none\}\}$ is the set of states, where $currBid$, $init$ and isB_r represent the value of Bid, $first$, isB_r, resp., for $r \in N_{eng}$, and $winner$ assigns a winner candidate;
- $\bar{w}_{eng} = \langle \mathsf{startingBid}, true, none, false, false, \cdots, false\rangle$;
- $T_{eng} = \{\langle currBid, false, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k} : currBid \in \mathbb{N}\ \&\ isB_r = true\ \&$ for every $i \neq r, isB_i = false\rangle\} \cup \{\langle currBid, false, winner, false, false, \cdots, false\rangle : currBid \in \mathbb{N}\}$, where $r, i \in N_{eng}$;
- $L_{eng} = \{(\langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle, decline^r)\} \cup \{(\langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle, accept^r) : isB_r = isBidding(r)$ or $init = true\}$, for all $\langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle \in W_{eng}$;
- U_{eng} is defined as follows: for all $w = \langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle \in W_{eng}$ and all $(a^{r_1}, \cdots, a^{r_k}) \in \prod_{r \in N_{eng}} A^r_{eng}$:
  - If $w \notin T_{eng}$ and for every $r \in N_{eng}$, $a^r \in L_{eng}(w)$, let $U_{eng}(w, (a^{r_1}, \cdots, a^{r_k})) = \langle currBid', false, winner', isB'_{r_1}, isB'_{r_2}, \cdots, isB'_{r_k}\rangle$, such that $currBid' = currBid +$ inc iff $accept^r$, for some $r \in N_{eng}$ and $currBid' = currBid$, otherwise. For each $r \in N_{eng}$, $isB'_r = false$ iff $decline^r$. Otherwise, $isB'_r = true$. Finally, $winner' = r$ iff $isB_r \&$ $\neg isB_y$ or $r \prec y \in \prec_{Eng}$, for all $y \neq r \in N_{eng}$. Otherwise, $winner' = none$.
  - Otherwise, $U_{eng}(w, (a^{r_1}, \cdots, a^{r_k})) = w$.
- $g_{eng}(r) = \{\langle currBid, false, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k} : currBid \in \mathbb{N}\ \&\ isB_r\ \&$ for every $i \neq r, \neg isB_i\rangle\} \cup \{\langle currBid, false, r, false, false, \cdots, false : currBid \in \mathbb{N}\rangle\}$;
- $alloc_{eng}(w, r) = 1$ and $p_{eng}(w, r) = value$, where $value = currBid$ iff $w \in g_{eng}(r)$ and $currBid \neq 0$. Otherwise, $alloc_{eng}(w, r) = 0$ and $value = 0$;
- $\prec_{N_{eng}}$ is defined as: $r' \prec r'' \in \prec_{N_{eng}}$ iff (r', r'') is in the lexicographic order of N_{eng}.

For each state $w = \langle currBid, init, winner, isB_{r_1}, isB_{r_2}, \cdots, isB_{r_k}\rangle \in W_{eng}$, let $\pi_{\Phi,eng}(w) = \{isBidding(r) : isB_r\ \&\ r \in N_{eng}\} \cup \{first : init\} \cup \{currWinner(r) : winner = r\}$; and $\pi_{\mathbb{Z},eng}(w, Bid) = currBid$.

Hereafter, we assume an instance of $M_{eng} \in \mathcal{M}_{eng}$ and Σ_{eng} for the constant values k, inc $\in \mathbb{N} \setminus \{0\}$ and startingBid $\in \mathbb{N}$.

Proposition 3.1. *M_{eng} is an ST-model and it is a model of Σ_{eng}.*

Proof. *(Sketch)* It is routine to check that M_{eng} is actually an ST-model. Given a path δ, any stage j of δ in M_{eng}, we need to show that $M_{eng}, \delta, j \models \varphi$, for each $\varphi \in \Sigma_{eng}$. Let us verify Rule 1. Assume $M_{eng}, \delta, j \models initial$, then $\delta[j] = \bar{w}_{eng}$. By the definition of $\pi_{\Phi,eng}$ and $\pi_{\mathbb{Z},eng}$, we have $first \in \pi_{\Phi,eng}(\bar{w}_{eng})$, $\pi_{\mathbb{Z},eng}(\bar{w}_{eng}) =$ startingBid and $isBidding(r), currWinner(r) \notin \pi_{\Phi,eng}(\bar{w}_{eng})$, for all $r \in N_{eng}$. Thus, $M_{eng}, \delta, j \models first \wedge Bid(\mathsf{startingBid}) \wedge \bigwedge_{r \in N_{eng}} \neg isBidding(r) \wedge \neg currWinner(r)$. Conversely, assume $M_{eng}, \delta, j \models first \wedge Bid(\mathsf{startingBid}) \wedge \bigwedge_{r \in N_{eng}} \neg isBidding(r) \wedge \neg currWinner(r)$, then by the definition of U_{eng}, we have $\delta[j] = \langle \mathsf{startingBid}, true, none, false, false, \cdots, false\rangle$, so $\delta[j] = \bar{w}_{eng}$ and $M_{eng}, \delta, j \models initial$.

The remaining rules are verified in a similar way. □

Example 3.1 describes an English Auction with to two rational agents, i.e. players that bid as long as the current value of the auction has not reached their private values.

Example 3.1. Let $M_{eng} \in \mathcal{M}_{eng}$, with startingBid $= 0$, increase $= 1$ and $N_{eng} = \{r_1, r_2\}$, i.e. k $= 2$. Thereby, we have $\mathcal{A}_{eng} = \{accept^{r_1}, decline^{r_1}, accept^{r_2}, decline^{r_2}\}$, $\Phi_{eng} = \{isBidding(r_1), isBidding(r_2)\}$ and $X' = \{Bid\}$.

Figure 2 illustrates a path in M_{eng} where we assume that r_1 and r_2 are rational players, i.e. $N_{eng} = N^{\vartheta}_{eng}$, and r_1 and r_2 private values are defined as $\vartheta_{r_1} = 3$ and $\vartheta_{r_2} = 2$. On the exhibited path we assume that both agents bid wrt. to their private values.

We now characterize the behavior of the protocol. Prop. 3.2 shows that: (i) if an agent is a winner, then no one else wins; (ii) if an agent pays more than zero, then all the remaining agents will pay zero, and (iii) if an agent pays x, he does not have pay any other amount.

Proposition 3.2. *Given $M_{eng} \in \mathcal{M}_{eng}$, for all $r \in N_{eng}$, $x \in \mathbb{N}$,*

1. $M_{eng} \models wins(r) \rightarrow \bigwedge_{r' \neq r \in N_{eng}} \neg wins(r')$;
2. $M_{eng} \models payment(r, x) \wedge x > 0 \rightarrow \bigwedge_{r' \neq r \in N_{eng}} payment(r', 0)$;
3. $M_{eng} \models payment(r, x) \rightarrow \bigwedge_{x' \neq x \in \mathbb{N}} \neg payment(r, x')$.

Proof. (1) Assume $M_{eng}, \delta, j \models wins(r)$ iff $\delta[j] \in g_{eng}(r)$ iff $(isBidding(r) \vee currWinner(r))$ and for all $r' \neq r \in N_{eng}$, $\neg isBidding(r')$. We have that $M_{eng}, \delta, j \models currWinner(r)$ iff $M_{eng}, \delta, j-1 \models isBidding(r) \wedge (r \prec r' \vee \neg isBidding(r'))$, for every $r' \neq r \in N_{eng}$. Thus, $M_{eng}, \delta, j \models \neg currWinner(r')$. For all $r' \neq r \in N_{eng}$, we have $\neg isBidding(r') \wedge \neg currWinner(r')$

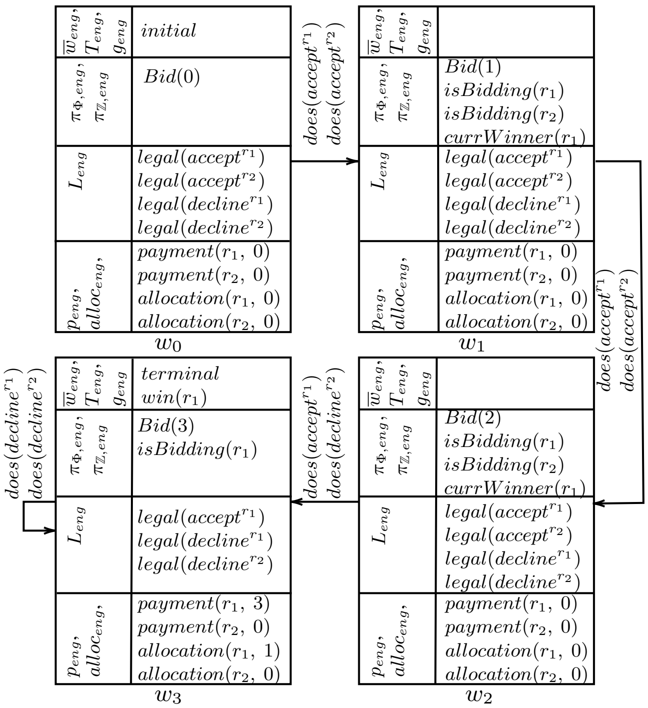

Figure 2. A Path in M_{eng}, where r_1 and r_2 are rational players

iff $\delta[j] \notin g_{eng}(r')$. Thereby, $M_{eng}, \delta, j \models \neg wins(r')$ and $M_{eng}, \delta, j \models wins(r) \to \bigwedge_{r' \neq r \in N_{eng}} \neg wins(r')$.

(2) Assume $M_{eng}, \delta, j \models payment(r, x) \wedge x > 0$, then $x = p_{eng}(\delta[j], r)$. Since $x \neq 0$, $\delta[j] \in g_{eng}(r)$ and $M_{eng}, \delta, j \models wins(r)$. By Statement (1), for all $r' \neq r \in N_{eng}$, we have $M_{eng}, \delta, j \models \neg wins(r')$ iff $\delta[j] \notin g_{eng}(r')$, then $p_{eng}(\delta[j], r') = 0$. Thus, $M_{eng}, \delta, j \models \bigwedge_{r' \neq r \in N_{eng}} payment(r', 0)$.

(3) Assume $M_{eng}, \delta, j \models payment(r, x)$ iff $x = p_{eng}(\delta[j], r)$. Thereby, for any $x' \neq x \in \mathbb{N}$, $x' \neq p_{eng}(\delta[j], r)$. Thus, $M_{eng}, \delta, j \bigwedge_{x' \neq x \in \mathbb{N}} \neg payment(r, x')$. □

Let us now characterize the impact of considering an English Auction with rational players. The following proposition describes their bidding strategy which was given as an intuition in Example 3.1:

Theorem 3.1. *Given the ST-model for the English Auction M_{eng}, for any $r \in N^{\vartheta}_{neg}$ with the private value ϑ_r,*

1. $M_{eng} \models \neg terminal \wedge add(Bid, \mathsf{inc}) > \vartheta_r \to does(decline^r)$;
2. $M_{eng} \models \neg terminal \wedge add(Bid, \mathsf{inc}) < \vartheta_r \to does(accept^r)$.

Proof. (*Sketch*) (1) Given a path δ in M_{eng} and a stage j in δ, assume that $M_{eng}, \delta, j \models \neg terminal \wedge add(Bid(x), \mathsf{inc}) > \vartheta_r$. Let $x = \pi_{\mathbb{Z},eng}(\delta[j], Bid) + \mathsf{inc}$. Note that inc, $\pi_{\mathbb{Z}(\delta[j], Bid), eng} > 0$ (by U_{eng} and $\bar{w}_{eng}$). If we have $\delta[j+1] \notin T_{eng}$, we have that $p_{eng}(\delta[j+1], r') = 0$, for every $r' \in N_{eng}$. Thereby, let us restrict our analysis to the case where $\delta[j+1] \in T_{eng}$, i.e. $M_{eng}, \delta, j \models \bigcirc terminal$. Let us assume that $\theta_r(\delta, j) = does(accept^r)$, then $\pi_{\mathbb{Z},eng}(\delta[j+1], Bid) = x$, (note that $x > \vartheta_r$). Since $\delta[j+1] \in T_{eng}$, it must be the case where $\theta_{r'}(\delta, j) = does(decline^{r'})$, for every $r' \neq r \in N_{eng}$, and $\delta[j+1] \in g_{eng}(r)$. Thereby, $p_{eng}(\delta[j+1], r) = x$ and $\Pi_r = \vartheta_r - x < 0$. Now, assume that $\theta_r(\delta, j) = does(decline^r)$. If $\delta[j+1] \notin g_{eng}(r)$, then $\Pi_r = 0$. Otherwise, by g_{eng} definition, it must be the case where $\theta_{r'}(\delta, j) = does(decline^{r'})$, for every $r' \in N_{eng}$ and $\pi_{\mathbb{Z},eng}(\delta[j+1], Bid) = \pi_{\mathbb{Z},eng}(\delta[j], Bid) = x - \mathsf{inc}$, i.e. bid value does not increase. Then, $\Pi_r = \vartheta_r - (x - \mathsf{inc}) > \Pi_r = \vartheta_r - x$. Therefore, to maximize Π_r, agent r would take action $decline^r$, i.e. $M_{eng}, \delta, j \models does(decline^r)$.

(2) The proof is performed in a similar way. □

Note that whenever $\vartheta_r = \pi_{\mathbb{Z},eng}(\delta[j], Bid) + \mathsf{inc}$, both actions $decline^r$ and $accept^r$ lead to the same payoff $\Pi_r = 0$. Thus, the rational agent r could take any of the actions. Thanks to Theorem 3.1, when all players are rational, it is possible to determinate the winner and the payment by observing their private values.

Proposition 3.3. *If $N_{eng} = N^{\vartheta}_{eng}$ and $wins(r)$, for $r \in N_{eng}$ with the private value ϑ_r, then:*

1. *$\vartheta_r = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng})$;*
2. *$\bigvee_{x \in I} payment(r, x)$, where $I = \{\vartheta, \cdots, add(\vartheta, \mathsf{inc})\}$ and $\vartheta = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng}$ & $\vartheta_{r'} \neq \vartheta_r)$.*

Proof. (*Sketch*) Given a path δ in M_{eng} and a stage j in δ, such that for all $0 \leq i < j$, $\delta[i] \notin T_{eng}$. Let us first verify part (1). Assume that $N_{eng} = N^{\vartheta}_{eng}$ and $wins(r)$, for some $r \in N_{eng}$ with the private value ϑ_r. Then, $\delta[j] \in g_{eng}(r)$ and $isBidding(r') \notin \pi_{\Phi,eng}(\delta[j])$, for all $r' \neq r \in N_{eng}$ and either case (i) $isBidding(r) \in \pi_{\Phi,eng}(\delta[j])$ or case (ii) $winner = r$. In case (i), we have that $\theta_r(\delta, j-1) \neq decline^r$ and $\pi_{\mathbb{Z},eng}(\delta[j-1], Bid) + \mathsf{inc} > \vartheta_r$ does not hold, i.e. $\pi_{\mathbb{Z},eng}(\delta[j-1], Bid) + \mathsf{inc} \leq \vartheta_r$ (by Theorem 3.1). Since $isBidding(r') \notin \pi_{\Phi,eng}(\delta[j])$, for all $r' \neq r \in N_{eng}$, we have $\theta_{r'}(\delta, j-1) \neq accept^{r'}$. Then, we have that $\pi_{\mathbb{Z},eng}(\delta[j-1]) + \mathsf{inc} < \vartheta_{r'}$ does not hold, i.e. $\pi_{\mathbb{Z},eng}(\delta[j-1]) + \mathsf{inc} \geq \vartheta_{r'}$. Thereby, we have $\vartheta_{r'} \leq \pi_{\mathbb{Z},eng}(\delta[j-1]) + \mathsf{inc} \leq \vartheta_r$, for all $r' \neq r \in N_{eng}$. Thus, $\vartheta_r = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng})$ and $payment(r, x)$, for some $\vartheta \leq x \leq \mathsf{inc} + \vartheta$, where $\vartheta = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng}$ & $\vartheta_{r'} \neq \vartheta_r)$. In case (ii), we have that $\theta_y(\delta, j-1) \neq accept^y$, for all $y \in N_{eng}$ (what show us that $\pi_{\mathbb{Z},eng}(\delta[j-1], Bid) + \mathsf{inc} \geq \vartheta_y$, by Theorem 3.1) and $currWinner(r) \in \pi_{\Phi,eng}(\delta[j-1])$. Thus, $\theta_r(\delta, j-2) \neq decline^r$ and $\pi_{\mathbb{Z},eng}(\delta[j-2], Bid) + \mathsf{inc} \leq \vartheta_r$. We also have $\theta_{r'}(\delta, j-1) \neq accept^{r'}$ and $\pi_{\mathbb{Z},eng}(\delta[j-1]) + \mathsf{inc} \geq \vartheta_{r'}$. Thereby, we have $\vartheta_{r'} \leq \pi_{\mathbb{Z},eng}(\delta[j-1]) + \mathsf{inc} \leq \vartheta_r$, for all $r' \neq r \in N_{eng}$. Thus, $\vartheta_r = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng})$ and $payment(r, x)$, for some $\vartheta \leq x \leq \mathsf{inc} + \vartheta$, where $\vartheta = maximum(\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng}$ & $\vartheta_{r'} \neq \vartheta_r)$.

Statement (2) follows from the previous proof. □

Let us focus on a specific class of players: agents that always bid, and evaluate the impact of having such players in an English Auction.

Definition 3.1. Given an ST-model $M_{eng} \in \mathcal{M}_{eng}$, an agent $r \in N_{eng}$ is called *always bidder* iff $\theta_r(\delta, j) = accept^r$, for any stage j and path δ. Let $N^{ab}_{eng} = \{r : r \in N_{eng}$ & r is an always bidder$\}$ be the subset of agents that always bid in N_{eng}.

Hereafter, we show that any path in a M_{eng} model is complete whenever there is at most one *always bidder* in the agent set. In other words, if all the bidders, except possibly one, stop bidding at some threshold, then any path is complete. If there are at least two distinguish always bidders, any path in M_{eng} is not complete.

Theorem 3.2. *Given any ST-model $M_{eng} \in \mathcal{M}_{eng}$,*

1. *If $|N^{ab}_{eng}| \leq 1$, then any path δ in M_{eng} is a complete path;*
2. *If $r, r' \in N^{ab}_{eng}$, where $r \neq r'$, then any δ in M_{eng} is not complete.*

Proof. (1) Assume $|N^{ab}_{eng}| = 0$, i.e. all the agents in N_{eng} are not always bidders, then for each $r \in N_{eng}$ there is some stage j_r in δ, where $\theta_r(\delta, j_r) \neq accept^r$, i.e. $\theta_r(\delta, j_r) = decline^r$. By U_{eng} and L_{eng}, for any $j > j_r$, $accept^r \notin L_{eng}(\delta[j])$ and $decline^r \in L_{eng}(\delta[j])$, i.e. the only legal action for r after doing decline is $decline^r$. Thereby, at stage $k = maximum(j_{r_1}, \cdots, j_{r_k})$, for every $r \in N_{eng}$, we have $\theta_r(\delta, k) = decline^r$. Thus, $\delta[k] \in T_{eng}$ and δ is a complete path. Now assume $|N^{ab}_{eng}| = 1$. Then, there is one agent $r' \in N_{eng}$ that is an always bidder. As we saw, there is a stage k where $\theta_r(\delta, k) = decline^r$ for all $r \neq r' \in N_{eng}$. Since $\theta_{r'}(\delta, k) = accept^r$, we have $M_{eng}, \delta, k+1 \models isBidding(r') \wedge \bigwedge_{r \neq r' \in N_{eng}} \neg isBidding(r)$ and $M_{eng}, \delta, k+1 \models wins(r')$. Thus, $\delta[k+1] \in T_{eng}$ and δ is complete.

(2) Assume that $r, r' \in N^{ab}_{eng}$ and $r \neq r'$. Let δ be a path in M_{eng}. Then, for any stage $j \geq 0$ in δ, we have $\theta_r(\delta, j) = accept^r$ and $\theta_{r'}(\delta, j) = accept^{r'}$. By U_{eng} definition we have that $isBidding(r), isBidding(r') \in \pi_{\Phi,eng}(\delta[j+1])$. Thus, $\delta[j+1] \notin T_{eng}$ and δ is not a complete path. □

For any $M_{eng} \in \mathcal{M}_{eng}$, we show that an always bidder is not a rational player and vice versa.

Proposition 3.4. *For any $r \in N_{eng}$, $r \in N^{ab}_{eng}$ iff $r \notin N^{\vartheta}_{eng}$.*

Proof. (*Sketch*) Assume that $r \in N^{ab}_{eng}$ for some $r \in N_{eng}$. Let δ be a path in M_{eng} and j a stage in δ. By Definition 3.1, we have $\theta_r(\delta, j) = accept^r$. For the sake of contradiction, let us assume that $r \in N^{\vartheta}_{eng}$. Thereby, r has a private value $\vartheta_r \in \mathbb{N}$. Assume that $\pi_{\mathbb{Z},eng}(\delta[j], Bid) + \mathsf{inc} > \vartheta_r$, then, by Theorem 3.1, we have that $\theta_r(\delta, j) = decline^r$. Since we have a contradiction, it must be that $r \notin N^{\vartheta}_{eng}$. The conversely is done in a similar way. □

The following proposition shows that if there is one always bidder and all other players are rational, then this always bidder agent wins the auction and its payment is determinate by the highest private value of the rational players.

Proposition 3.5. *If $r \in N^{ab}_{eng}$ and for all $r' \neq r \in N_{eng}$, $r' \in N^{\vartheta}_{eng}$, then for any $e > 0$, if $\delta[e] \in T_{eng}$ then $\delta[e] \in g_{eng}(r)$ and $\vartheta \leq p_{eng}(\delta[e], r) \leq \mathsf{inc} + \vartheta$, where $\vartheta = maximum(\{\vartheta_{r'} : \vartheta_{r'}$ is the private value of $r' \in N^{\vartheta}_{eng}\})$.*

Proof. (*Sketch*) The proof is done in a similar way then Proposition 3.3. The proof idea is that for any $r' \neq r \in N_{eng}$, since r' is a rational player, he will stop bidding if Bid is larger or equal to his private value (see Theorem 3.1). Thereby, r will win the auction when the agent with the highest private value declines. □

We conclude the section by briefly discussing variants of single-good and single-side Auction. A Descending-bid Auction (or Dutch Auction), is similar to Σ_{eng}. The key difference is that in the Dutch Auction the bidding value should be decreased at each round until at least one agent accepts the bid. As for English Auction, the tie-breaking is solved with the help of the total order $\prec_N$. In a similar way, a sealed-bid auction can also be represented in ADL: once bids have been submitted, the auctioneer stops the auction and defines the winner and the payment (rules are close to those in Σ_{eng}).

4 REPRESENTING MULTI-UNIT AUCTIONS

In this section, we consider multi-unit auctions. The most popular are the multi-unit sealed-bid auctions. There exist several variants depending on the payment rules: first price, second price or based on the Vickrey-Clarke-Groves mechanism [11]. Hereafter, we represent a Multi-Unit Vickrey Auction, where the payment for the winners is the highest losing bid. The payment is uniform: all the winners pay the same price.

4.1 Multi-Unit Vickrey Auction

To describe a Multi-Unit Vickrey Auction with k bidders, we first define the auction signature, written $\mathcal{S}_{vic}$, as follows:

- $N_{vic} = \{r_1, r_2, \cdots, r_\mathsf{k}\}$;
- $\mathcal{A}_{vic} = \bigcup_{r \in N_{vic}} A^r_{vic}$ where $A^r_{vic} = \{bid^r(x, y), noop^r : x, y \in \mathbb{N}\}$; where $bid^r(x, y)$, means that r bids x for each unit and intend to get at most y units and $noop^r$ says that r does nothing;
- $\Phi_{vic} = \{bidding, outbid(r, r') : r, r' \in N_{vic}\}$, where *outbid*$(r, r')$ represents whether r outbids r', i.e. the bid of r is higher than the bid of r'. When r and r' bid the same value, $outbid(r, r')$ is true if r precedes r' in the lexicographic order;
- $X_{vic} = \{offer^r, amt^r, firstLoser : r \in N_{vic}\}$, where $offer^r$ represents the bid value for r, amt^r represents how many goods r wants to get (at most), *firstLoser* represents the highest bid value of between the agents that do not win.

Let $\mathsf{total}, \mathsf{k} \in \mathbb{N} \setminus \{0\}$ and $\mathsf{reservePrice} \in \mathbb{N}$ be constants describing the quantity of units, the quantity of bidders in N_{vic} and the reserve price for each unit, respec. The rules of a Multi-Unit Vickrey Auction are formulated by ADL-formulas as shown in Figure 3.

1. $initial \leftrightarrow firstLoser(\mathsf{reservePrice}) \wedge bidding \wedge$
 $\bigwedge_{r,r' \in N_{vic}} offer^r(0) \wedge amt^r(0)$
 $\wedge \neg outbid(r, r')$
2. $allocation(r, y_r) \leftrightarrow (y_r > 0 \wedge offer^r \geq \mathsf{reservePrice} \wedge$
 $(amt^r(y_r) \wedge add(\{y_{r'} : r' \in N_{vic}\ \&\ outbid(r', r)$
 $\&\ allocation(r', y_{r'})\}, y_r) \leq \mathsf{total} \vee amt^r(k) \wedge$
 $y_r < k \wedge add(\{y_{r'} : r' \in N_{vic}\ \&\ outbid(r', r)\ \&$
 $allocation(r', y_{r'})\}, y_r) = \mathsf{total})) \vee y_r = 0$
3. $firstLoser(max(x, \mathsf{reservePrice})) \leftrightarrow offer^r(x))$
 $\wedge \neg wins(r) \wedge \bigwedge_{r' \in N_{vic}} \neg wins(r') \wedge$
 $outbid(r, r') \vee \neg outbid(r, r')$
4. $wins(r) \leftrightarrow allocation(r, x) \wedge x > 0$
5. $payment(r, times(firstLoser, y)) \wedge allocation(r, y)$
6. $terminal \leftrightarrow \neg bidding$
7. $legal(bid^r(x, y)) \leftrightarrow initial$
8. $legal(noop^r) \leftrightarrow terminal$
9. $\bigcirc(offer^r(x) \wedge amt^r(y)) \leftrightarrow terminal \wedge$
 $offer^r(x) \wedge amt^r(y) \vee does(bid^r(x, y))$
10. $\bigcirc \neg bidding(r) \leftrightarrow \top$
11. $\bigcirc outbid(r, r') \leftrightarrow terminal \wedge outbid(r, r') \vee$
 $does(bid^r(x, y)) \wedge does(bid^{r'}(x', y'))$
 $\wedge (x > x' \vee x = x' \wedge r \prec r')$

Figure 3. Multi-Unit Vickrey Auction represented by Σ_{vic}

In the initial state, no one is bidding and the reserve price value is set (Rule 1). Each player can bid and say (at most) how many goods he wants (Rule 7). The propositions and variables are updated to the next state according to the bids (Rules 9 to 11). A player is assigned to at most the desired quantity of goods if he bids at least the reserve price and if there are goods left after the allocation to the players who outbid him. Otherwise, the player is assigned to zero goods (Rule 2). The payment for each player is determined according to the allocation rule and the highest losing price (Rule 3 and 5). A player wins if he gets at least one good and he can only do action $noop$ at terminal states (Rules 4 and 8). We are in a terminal state if it is not the bidding turn (Rule 6). Let Σ_{vic} be the set of rules 1-11.

Let $\mathcal{M}_{vic}$ be the set of ST-models M_{vic} defined with respect to constant values total, k and $\mathsf{minimumBid}$. Each $M_{vic} \in \mathcal{M}_{vic}$ is defined as follows:

- $W_{vic} = \{\langle isBid, b_{r_1}, \cdots, b_{r_\mathsf{k}}, qtd_{r_1}, \cdots, qtd_{r_\mathsf{k}}, out_{r_1,r_1}, \cdots, out_{r_\mathsf{k},r_\mathsf{k}}\rangle : b_r, qtd_r \in \mathbb{N} \,\&\, isBid, out_{r,r'} \in \{true, false\}\}$ is the set of states, where $isBid$, b_r, qtd_r and $out_{r,r'}$ represents $bidding$, $offer^r$, amt^r and $outbid(r, r')$, resp., for $r, r' \in N_{vic}$;
- $\bar{w}_{vic} = \langle true, 0, \cdots, 0, 0, \cdots, 0, false, \cdots, false\rangle$;
- $T_{vic} = \{\langle false, b_{r_1}, \cdots, b_{r_\mathsf{k}}, qtd_{r_1}, \cdots, qtd_{r_\mathsf{k}}, out_{r_1,r_1}, \cdots, out_{r_\mathsf{k},r_\mathsf{k}}\rangle : b_r, qtd_r \in \mathbb{N} \,\&\, out_{r,r'} \in \{true, false\}\}$, where $r, r' \in N_{vic}$;
- $L_{vic} = \{(w, noop^r) : r \in N_{vic} \,\&\, w \in T_{vic}\} \cup \{(\bar{w}_{vic}, bid^r(x, y)) : x, y \in \mathbb{N} \,\&\, r \in N_{vic}\}$;
- U_{vic} is defined as: for all $w = \langle isBid, b_{r_1}, \cdots, b_{r_\mathsf{k}}, qtd_{r_1}, \cdots, qtd_{r_\mathsf{k}}, out_{r_1,r_1}, \cdots, out_{r_\mathsf{k},r_\mathsf{k}}\rangle \in W_{vic}$ and all $d \in \prod_{r\in N_{vic}} A^r_{vic}$:
  - If $w \notin T_{vic}$ and $d = (bid^{r_1}(x_{r_1}, y_{r_1}), \cdots, bid^{r_\mathsf{k}}(x_{r_\mathsf{k}}, y_{r_\mathsf{k}}))$, where $x_r, y_r \in \mathbb{N}$ and $r \in N_{vic}$, then $U_{vic}(w, d) = \langle false, b'_{r_1}, \cdots, b'_{r_\mathsf{k}}, qtd'_{r_1}, \cdots, qtd'_{r_\mathsf{k}}, out'_{r_1,r_1}, \cdots, out'_{r_\mathsf{k},r_\mathsf{k}}\rangle$, such that for each $r, i \in N_{vic}$, each component is updated as follows: $b'_r = x_r$, $qtd' = y_r$ and $out'_{r,i} = true$ iff either (i) $x_r > x_i$ or (ii) $x_r = x_i$ and $r \prec i \in \prec_{N_{vic}}$.
  - Otherwise, let $U_{vic}(w, d) = w$.
- $g_{vic}(r) = \{w : w \in W_{vic} \,\&\, alloc_{vic}(w, r) > 0\}$;
- $p_{vic}(w, r) = \pi_{\mathbb{Z},vic}(w, firstLoser) \times \pi_{\mathbb{Z},vic}(w, amt^r)$;
- $alloc_{vic}(w, r) = k$, let $qtd_{alloc} = \sum_{r' \neq r \in N_{vic} \,\&\, out_{r',r}} qtd^{r'}$, then k is defined as: (i) $k = qtd^r$ if $\mathsf{total} \geq qtd^r + qtd_{alloc}$ and $b_r \geq \mathsf{reservePrice}$; (ii) $k = q$, where $q = \mathsf{total} - qtd_{alloc}$ if $q \geq 0$ and $b_r \geq \mathsf{reservePrice}$; (iii) otherwise, $k = 0$.
- $\prec_{N_{vic}}$ is defined as: $r' \prec r'' \in \prec_{N_{vic}}$ iff (r', r'') is in the lexicographic order of N_{vic}.

Finally, for each state $w = \langle currBid, snd, isBid, isH_{r_1}, isH_{r_2}, \cdots, isH_{r_\mathsf{k}}\rangle \in W_{vic}$ and $i \in N_{vic}$, let $\pi_{\Phi,vic}(w) = \{outbid(r, r') : out_{r,r'} \,\&\, r, r' \in N_{vic}\} \cup \{bidding : isBid\}$; $\pi_{\mathbb{Z},vic}(w, offer^i) = b_i$ and $\pi_{\mathbb{Z},vic}(w, amt^i) = qtd_i$; $\pi_{\mathbb{Z},vic}(w, firstLoser) = k$, where $k = maximum(x, \mathsf{reservePrice})$, if $x = \pi_{\mathbb{Z},vic}(w, offer^r)$, for some $r \in N_{vic}$ such that $w \notin g_{vic}(r)$ and for all $r' \neq r \in N_{vic}$, either (i) $w \notin g_{vic}(r')$ and $out_{r,r'}$ or (ii)$\neg out_{r,r'}$. Otherwise, $k = \mathsf{reservePrice}$.

Hereafter, we fix an instance of $M_{vic} \in \mathcal{M}_{vic}$ and Σ_{vic} for the constants $\mathsf{total}, \mathsf{k} \in \mathbb{N} \setminus \{0\}$ and $\mathsf{reservePrice} \in \mathbb{N}$.

Proposition 4.1. *M_{vic} is an ST-model and it is a model of Σ_{vic}.*

Proof. (*Sketch*) The proof is performed as for Prop. 3.1. □

Example 4.1. Let $M_{vic} \in \mathcal{M}_{vic}$, with $\mathsf{reservePrice} = 1$, $\mathsf{total} = 10$ and $N_{vic} = \{r_1, r_2, r_3\}$. Figure 4 illustrates a path δ in M_{vic}, where $\theta(\delta, 0) = (bid^{r_1}(2, 2), bid^{r_2}(6, 7), bid^{r_3}(5, 4))$. In the terminal state, since $offer^{r_2} > offer^{r_3} > offer^{r_1}$, the allocation rule assigns 7 units to r_2 (i.e. exactly amt^{r_2}), 3 units to r_3 (i.e. $\mathsf{total} - amt^{r_2}$, which is smaller then amt^{r_3}) and 0 units to r_1.

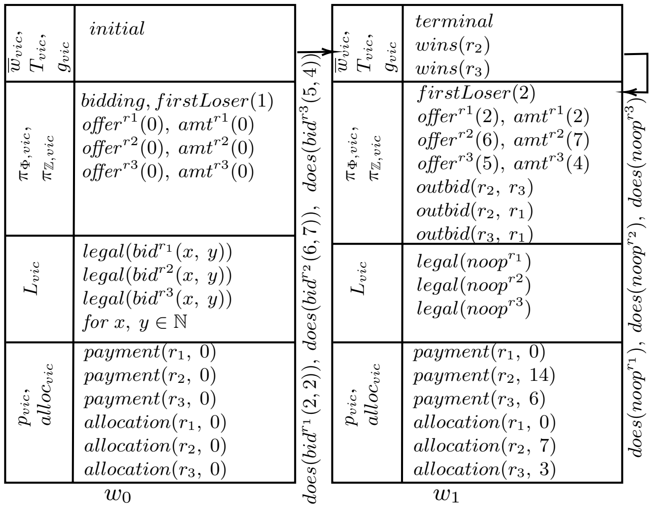

Figure 4. A Path in M_{vic}

The following properties show that (i) the initial state is always succeed by a terminal state, (ii) any path in a M_{vic} model is complete, and (iii) if a player bids, he can no longer bid.

Proposition 4.2. *Given any model $M_{vic} \in \mathcal{M}_{vic}$, $x, y, x', y' \in \mathbb{N}$,*

1. *$M_{vic} \models initial \to \bigcirc terminal$;*
2. *Any path δ in M_{vic} is a complete path;*
3. *$M_{vic} \models does(bid^r(x, y)) \to \bigcirc\neg legal(bid^r(x', y'))$.*

Proof. (1) By Rule 11 of Σ_{vic}, we have that $\bigcirc initial \to \neg bidding$, then, by Rule 7, $\bigcirc initial \to terminal$.

(2) From Statement (1), for any path δ in M_{vic}, since $\delta[0] = \bar{w}_{vic}$ is the initial state, we have $\delta[1] \in T_{vic}$. Thus, δ is a complete path.

(3) Assume $M_{vic}, \delta, j \models does(bid^r(x, y))$, then by Rule 11 of Σ_{vic}, $M_{vic}, \delta, j + 1 \models terminal$. Thus, $bid^r(x', y') \notin L_{vic}(\delta[j + 1])$ and $M_{vic}, \delta, j \models \bigcirc\neg legal(bid^r(x', y'))$. □

The following property shows the behavior of a rational player in a bidding round of a M_{vic} with $total = 1$, i.e. M_{vic} is similar to a Single-Unit Vickrey Auction.

Theorem 4.1. *Given the ST-model for M_{vic}, if $\mathsf{total} = 1$, then for any $r \in N^{\vartheta}_{vic}$ with the private value ϑ_r and $y > 0$, $M_{vic} \models \neg terminal \to does(bid^r(\vartheta_r, y)))$.*

Proof. Given a path δ in M_{vic}, we have that $M_{vic}, \delta, 0 \models initial$ and $M_{vic}, \delta, j \models terminal$, for any $j > 0$ (by path definition and Proposition 4.2). Assuming $M_{vic}, \delta, j \models \neg terminal$, it must be that $j = 0$. Thereby, $M_{vic}, \delta, j \models legal(bid^r(x, y))$, for any $x, y \in \mathbb{N}$ and $M_{vic}, \delta, j \models \neg legal(noop^r)$. Thus, $M_{vic}, \delta, j \models does(bid^r(x, y))$. Since $\mathsf{total} = 1$, the allocation rule will assign at most 1 unit. Thereby, if $outbids(r, r')$, for all $r' \neq\in N_{vic}$, then

$payment(r, firstLoser)$ and $payment(r, 0)$ otherwise. If $x = \vartheta_r$, r would have $\Pi_r = \vartheta - firstLoser$ whenever $\vartheta > firstLoser$ and $\Pi_r = 0$ if $\vartheta \leq firstLoser$. Suppose that $x < \vartheta$. If $\vartheta > x \geq firstLoser$, then $\Pi_r = \vartheta - firstLoser$. If $firstLoser > \vartheta > x$, then $\Pi_r = 0$. However, if $\vartheta > firstLoser > x$, then $\Pi_r = 0$, whereas if $x = \vartheta$, he would have $\Pi_r = \vartheta - firstLoser > 0$. It is easy to see that having $x > \vartheta$, would not increase Π_r. Thereby, $M_{vic}, \delta, j \models does(bid^r(\vartheta, y))$, for $y < 0$.

□

The representation of the First-price Sealed-bid Multi-Unit Auction (or Blind Auction) with ADL-formulas can be defined in a similar way to Σ_{vic}. In this case, the payment rule should be defined with the first winning bid. Actually, many variants of payment rules, such as nonuniform payment, may be considered with ADL.

ADL is also suited to represent multi-unit double-side auctions. For instance, the Multi-Unit Double-Side Vickrey Auction can be defined in a similar way to Σ_{vic}. In this case, the sellers would have distinguished actions from the bidders. A seller's action would specify his available quantity of units and his minimum selling price. Additionally, the allocation rule would also consider a seller ranking, similar to the outbid order among the bidders in Σ_{vic}.

5 CONCLUSION

We aim to design a *General Auction Player* (GAP) that can interpret and reason about the rules governing an auction-based market. To allow an agent to switch between different kinds of markets, the first step is to develop a general Auction Description Language (ADL), a logic-based language for representing the rules of an auction market, which will then allow a GAP to reason strategically in different environments. In this paper, we have seen that ADL is general enough for representing different kinds of auction. We focused on single-side auctions and we have seen that auctions may be represented compactly. We have also seen that the clear semantics expressed in terms of state transition models enables us to express properties about the protocol (terminal states, payment).

For future work, we have two main tracks to explore. First, from the auctioneer point of view, our goal is to explore two main variants of auctions: double-side auction [14] and combinatorial auction [18]. Clearly, ADL is well suited for both of them but requires some extension. Multiple sorts of goods are not yet possible for instance. We aim to embed into ADL a bidding language focusing on bids (goods, quantity, bundles, and preferences) [17].

Second, we want to investigate how ADL-based players may be implemented so that they can reason about the properties of an auction such as the strategy-proof aspect. The key difference, when the player perspective is considered, is the epistemic and strategic aspects: players have to reason about other players' behavior. The epistemic component will allow an agent to bid according to its beliefs about other agents' private values. In order to reduce the model-checking complexity in relation to the epistemic extensions of GDL (GDL-III [21] and Epistemic GDL [8]), we first aim to explore a conservative extension in GDL, where the belief and knowledge operators are restricted to numerical variables representing private values (e.g. "Ann *believes* that Bob's private value for each good is 10"). In this approach, logical connectives such as disjunction will be avoided.

ACKNOWLEDGEMENTS

This research is supported by the ANR project AGAPE ANR-18-CE23-0013. We are grateful to Umberto Grandi, Jérôme Lang, Emiliano Lorini and Francesco Belardinelli for their insightful comments.

REFERENCES

[1] Mark Armstrong, 'Optimal Multi-Object Auctions', *Review of Economic Studies*, (2000).
[2] Chitta Baral and Cenk Uyan, 'Declarative specification and solution of combinatorial auctions using logic programming', in *Lecture Notes in Computer Science*, (2001).
[3] Yann Chevaleyre, Paul E Dunne, Ulle Endriss, Jerome Lang, Michel Lemaitre, Nicolas Maudet, Julian Padget, Steve Phelps, Juan a Rodrguez-Aguilar, and Paulo Sousa, 'Issues in Multiagent Resource Allocation', *Informatica*, **30**(1), 3–31, (2006).
[4] Sven de Vries, James Schummer, and Rakesh V. Vohra, 'On ascending Vickrey auctions for heterogeneous objects', *Journal of Economic Theory*, (2007).
[5] Rupert Freeman, Markus Brill, and Vincent Conitzer, 'General tiebreaking schemes for computational social choice', in *Proc. of the International Joint Conference on Autonomous Agents and Multiagent Systems, AAMAS*, (2015).
[6] Michael Genesereth and Michael Thielscher, *General game playing*, Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan & Claypool Publishers, 2014.
[7] Matthew O. Jackson and Jeroen M. Swinkels, 'Existence of equilibrium in single and double private value auctions', *Econometrica*, **73**(1), 93–139, (2005).
[8] Guifei Jiang, Dongmo Zhang, Laurent Perrussel, and Heng Zhang, 'Epistemic GDL: A logic for representing and reasoning about imperfect information games', in *IJCAI International Joint Conference on Artificial Intelligence*, (2016).
[9] Mireia Jofre-Bonet and Martin Pesendorfer, 'Optimal sequential auctions', *International Journal of Industrial Organization*, (2014).
[10] Paul Klemperer, 'Auction Theory: A Guide to the Literature', *Journal of Economic Surveys*, (1999).
[11] Vijay Krishna, *Auction Theory*, Academic Press, 2009.
[12] H. Geun Lee and R. Lee, 'A hybrid approach of linear programming and logic modeling for the market core of sealed bid auctions', *Annals of Operations Research*, **75**, (1997).
[13] Kevin M. Lochner and Michael P. Wellman, 'Rule-based specification of auction mechanisms', in *Proc. of the Third International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'2004)*, (2004).
[14] R. Preston McAfee, 'A dominant strategy double auction', *Journal of Economic Theory*, **56**(2), 434–450, (April 1992).
[15] Munyque Mittelmann and Laurent Perrussel, 'Game description logic with integers: A GDL numerical extension', in *FoIKS 2020*, eds., Andreas Herzig and Juha Kontinen, Cham, (2020). Springer International Publishing.
[16] D. Nautz, 'Optimal bidding in multi-unit auctions with many bidders', *Economics Letters*, **48**(3-4), 301–306, (1995).
[17] Noam Nisan, 'Bidding languages', *Combinatorial Auctions*, 1–19, (2004).
[18] Aleksandar Pekeč and Michael H. Rothkopf, 'Combinatorial auction design', *Management Science*, (2003).
[19] Daniel Rolli, Stefan Luckner, Henner Gimpel, and Christof Weinhardt, 'A Descriptive Auction Language', *Electronic Markets*, **16**(1), 51–62, (2006).
[20] Michael Thielscher, 'GDL-II', *KI - Künstliche Intelligenz*, **25**(1), 63–66, (2011).
[21] Michael Thielscher, 'GDL-III: A proposal to extend the game description language to general epistemic games', in *Proc. the European Conference on Artificial Intelligence (ECAI)*, volume 285, pp. 1630–1631, Hague, (2016).
[22] Michael Thielscher and Dongmo Zhang, *From General Game Descriptions to a Market Specification Language for General Trading Agents*, 259–274, Springer Berlin Heidelberg, 2010.
[23] Dongmo Zhang and Michael Thielscher, 'Representing and Reasoning about Game Strategies', *Journal of Philosophical Logic*, **44**(2), 203–236, (2014).