



# Representing and reasoning about auctions

MunIQUE MittelmANN<sup>1</sup> · Sylvain Bouveret<sup>2</sup> · Laurent Perrussel<sup>1</sup>

Accepted: 1 February 2022 / Published online: 7 March 2022  
© Springer Science+Business Media, LLC, part of Springer Nature 2022

## Abstract

The goal of this paper is to propose a framework for representing and reasoning about the rules of auction-based protocols. Such a framework is of interest for building digital marketplaces based on this type of mechanism. Hence the framework should fulfill two requirements: (i) it should enable bidders to express their preferences over combinations of items and (ii) it should allow the mechanism designer to describe the rules governing the market, namely the legality of bids, the allocative choice, and the payment rule. To do so, we define a logical language in the spirit of the *Game Description Language*, namely *Auction Description Language* with a set of functions  $\mathcal{F}_B$  ( $ADL[\mathcal{F}_B]$ ).  $ADL[\mathcal{F}_B]$  is the first language for describing auctions in a logical framework. With our approach, each stage in a protocol is seen as an independent direct revelation mechanism. Our contribution is three-fold: first, we illustrate the general dimension by representing different kinds of protocols. Second, we show how this machine-processable language enables reasoning about auction properties, including playability, termination, and classical conditions from mechanism design (e.g., budget-balance and individual rationality). Finally, we develop a model-checking algorithm for  $ADL[\mathcal{F}_B]$ , with complexity in PTIME when the functions in  $\mathcal{F}_B$  can be computed in polynomial time.

**Keywords** Logics for multi-agents · Game description language · Auction-based markets

---

✉ MunIQUE MittelmANN  
munIQUE.mittelmANN@irit.fr

Sylvain Bouveret  
sylvain.bouveret@imag.fr

Laurent Perrussel  
laurent.perrussel@irit.fr

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

<sup>2</sup> Université Grenoble Alpes - LIG, Grenoble, France

## 1 Introduction

Auction-based markets are widely used for automated business transactions. There are numerous variants depending on the parameters considered, including the number of distinct items<sup>1</sup> and their copies and the number of sellers and buyers [35, 36]. For a fixed set of parameters, the protocol, i.e., the bidding, payment and allocation rules, may also differ. Building intelligent agents that can switch between different auctions and process their rules is a key issue for building automated auction-based marketplaces. In this setting, the auction designer should at first describe the rules governing the auction and second allow participants to express their preferences. The aim of this paper is to propose a language with clear semantics for enabling the representation of auctions as well as the reasoning about its rules and properties. More precisely, such a language should address the six auction dimensions introduced by Parkes and Kalagnanam [33]:

1. Resources: the auction protocol may involve a single item or multiple items, with single or multiple units of each item. The type of the item may also be considered, i.e., the item may be a multi-attribute commodity;
2. Market structure: the auction may differ on whether it has single or multiple buyers and/or sellers. In single-sided auctions, there is either a single seller selling resources to multiple buyers (i.e., forward auctions), or a single buyer sourcing resources from multiple suppliers (i.e., reverse auctions). In a double auction, there are multiple suppliers selling resources to multiple buyers. Finally, an exchange is a generalization of double auctions, where the participants *trade* items;
3. Preference structure: the participants' preferences define their utilities over different outcomes (e.g., marginal utility, quasi-linear utility);
4. Bid structure: it refers to the flexibility with which participants can express their preferences. The bid structure ranges from simple statements of willingness to accept a given selling price to complex bids that state prices, quantities, bundles, and logical connectives. In single-dimensional auctions, only one attribute is considered in the bid (e.g., the price offered for a good). On the other hand, multi-dimensional auctions handle a number of attributes in the bid (e.g., quality, delivery date) [55];
5. Market clearing: in relation to the method for matching the supply to demand, an auction may be single-sourcing (i.e., matching pairs of buyers and sellers) or multi-sourcing (i.e., matching multiple sellers with a single buyer, or vice-versa);
6. Information feedback: auctions may also differ on whether they are direct (i.e., one-stage protocols without information feedback) or indirect mechanisms (i.e., protocols where agents can adjust their bids in response to information feedback).

In the spirit of the *General Game Playing* (GGP) [21] where games are described with the help of *Game Description Language* (GDL), we previously introduced a logical language for describing auctions, denoted Auction Description Language (ADL) [45, 46].

In this paper, we go further and present a new version of ADL, named ADL [ $\mathcal{F}_B$ ], which is described as *Auction Description Language* with a set of functions  $\mathcal{F}_B$ . ADL [ $\mathcal{F}_B$ ] builds upon bidding languages, and hence provides a natural way to represent a wide range of protocols, ranging from single-units auctions to iterative combinatorial exchanges [53]. As

<sup>1</sup> Throughout the paper we use the terms “item” and “good” interchangeably.

for GDL and ADL, we propose a precise semantics based on state-transition models, that gives a clear meaning to the auction rules. Different bidding languages can be used with ADL  $[\mathcal{F}_B]$ , including the Tree-Based Bidding Language [53], which generalizes known languages such as XOR/OR [50] to combinatorial exchange. To the best of our knowledge, ADL  $[\mathcal{F}_B]$  is the first framework offering a unified perspective on an auction mechanism and it offers two benefits: (i) with this language, one can represent many kinds of auctions in a compact way and (ii) the precise state-transition semantics can be used to derive key properties.

Our motivation for proposing ADL  $[\mathcal{F}_B]$  is twofold. First, to enable automated reasoning about properties and features of an auction. These include determining whether the protocol is well-formed, in terms of termination and playability, as well as properties related to the mechanism itself, such as individual rationality and efficiency. Second, inspired by General Game Playing, our goal is to implement general players and an auction server to evaluate their adaptability among different protocols and their ability to learn from other agents' behavior. Just like for the GGP, ADL  $[\mathcal{F}_B]$  provides the ground for testing auction players and creating market-based competitions.

## 1.1 Contributions

Our main contribution is to provide a general framework for representing auctions. For discussing the generality of ADL  $[\mathcal{F}_B]$ , we refer again to the six dimensions of auctions [33]:

1. **Resources** as from ADL, we can represent variants with single and multiple units. ADL  $[\mathcal{F}_B]$  also considers multiple types of goods;
2. **Market structure** while ADL is restricted to single-sided auctions, ADL  $[\mathcal{F}_B]$  also represents double auctions and exchange protocols. However, we cannot represent dynamic sets of agents and goods;
3. **Preference structure** as we are concerned with the general representation of auctions, we focus on how agents can *express* their preferences rather than on the underlying structures of their utility functions. Following the literature on mechanism design [54], we consider agents with quasi-linear utilities. In such a case, the agents' utilities are based on a preference function over the outcomes and her payment. For a discussion on the hierarchy of preference functions, the reader may refer to Feige et al. [19];
4. **Bid structure** ADL  $[\mathcal{F}_B]$  can be used alongside different bidding languages. We define requirements for the bidding set and we illustrate how to employ two languages with different expressiveness in ADL  $[\mathcal{F}_B]$ . This aspect was not previously explored with ADL;
5. **Market clearing** as for ADL, we can model single-sourcing auctions. ADL  $[\mathcal{F}_B]$  can also encode multi-sourcing auctions;
6. **Information feedback** in ADL  $[\mathcal{F}_B]$ , direct mechanisms are described by a one-stage protocol and indirect mechanisms are represented by iterative protocols.

Hence, ADL  $[\mathcal{F}_B]$  addresses all these dimensions and is general enough to represent most settings. Notice that, similar to GDL (Game Description Language), ADL  $[\mathcal{F}_B]$  focuses on deterministic and perfect information protocols. GDL-II is an extension for handling imperfect information [64] and may be considered for auctions such as iterative sealed-bid auctions. These are however less common.

Beyond the auction setting,  $\text{ADL}[\mathcal{F}_B]$  is also able to represent several kinds of resource allocation problems: as noticed by Chevaleyre et al. [15], auctions can be seen as a sub-division of allocation mechanisms. The main characteristics of an auction are (i) central authority (the auctioneer), (ii) monetary transfer among participants, and (iii) agents' preferences expressed through bids. All these key features are expressible in  $\text{ADL}[\mathcal{F}_B]$ .

We also show how  $\text{ADL}[\mathcal{F}_B]$  can be used for the automated verification of mechanism design properties and for automatically checking whether descriptions written in  $\text{ADL}[\mathcal{F}_B]$  are well-formed. We illustrate the generality of  $\text{ADL}[\mathcal{F}_B]$  by focusing on two auction types: simultaneous ascending auction (SAA) and combinatorial exchange as we believe they are representative to demonstrate the expressive power of  $\text{ADL}[\mathcal{F}_B]$ . Combinatorial exchange is a generalization of combinatorial and double-sided auctions, and SAA generalizes the English auction to multiple items. We evaluate these protocols in terms of the aforementioned properties. Finally, we show that if functions in  $\mathcal{F}_B$  can be computed in polynomial time, then the model-checking problem for a  $\text{ADL}[\mathcal{F}_B]$ -formula belongs to PTIME, that is, that it can be solved in polynomial time using a deterministic Turing machine.

The paper is organized as follows: in Sect. 2 we discuss the related work. In Sect. 3 we detail the semantics and syntax of  $\text{ADL}[\mathcal{F}_B]$ . In Section 4 we introduce general properties for evaluating  $\text{ADL}[\mathcal{F}_B]$ -based protocols. In Sects. 5 and 6, we show how to represent auctions in  $\text{ADL}[\mathcal{F}_B]$  and we demonstrate how to derive properties from their description. In Sect. 7, we explore the complexity of deciding whether an  $\text{ADL}[\mathcal{F}_B]$ -formula is true with respect to a model and an execution of such model. Finally, Sect. 8 concludes the paper and discusses future work. The proofs omitted in the main text are presented in the appendix.

## 2 Related work

Our work is rooted in the key contributions on Auction Theory [36, 50, 51, 72]. All these works adopt a mechanism design perspective: they focus on designing and evaluating protocols and bidding languages. Our work has a different purpose. The  $\text{ADL}[\mathcal{F}_B]$  language is intended to represent the auction protocols and to allow a modular definition of actions sets, which may be characterized by a subset of a bidding language.  $\text{ADL}[\mathcal{F}_B]$  can be used to automatically derive properties for these protocols.  $\text{ADL}[\mathcal{F}_B]$  is also a tool for mechanism design, since it is a well suited framework for testing new auctions.

### 2.1 Auction representation

To the best of our knowledge, almost all contributions on the computational representation of auctions focus on the implementation of the winner determination problem. For instance, Baral and Ulyan [4] 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 by Lee and Lee [38]: they focus on sealed-bid auctions and show how qualitative reasoning helps to refine the optimal quantitative solutions. Giovannucci et al. [22] explore a graphical formalism to compactly represent the winner determination problem for multi-unit combinatorial auctions. Linear Logic has also been used [57] for modeling combinatorial auctions. The authors explore the representation of bids and the winner determination. However, there is no temporal operator to allow reasoning on iterative auctions.

The descriptive auction language [59] allows a formal specification of auctions. Agents can only bid through XOR combinations of items. Although the XOR language is widely used, the specification approach is more flexible when the bidding language is not fixed for all protocols. In fact, as stated by Nisan [51], the choice of a bidding language should aim to find a balance between expressivity (i.e., being able to express every preference function) and simplicity (i.e., being intuitive and computationally efficient). There are numerous auction protocols that do not require the expressive power of the XOR language and can be implemented with simpler bidding languages. For instance, in a Dutch auction, agents may simply say whether they accept the current selling price or not. Our proposal is intended to be more flexible, as one may use different bidding languages with ADL [ $\mathcal{F}_B$ ] for defining distinct protocols.

A rule-based scripting language for representing single-dimensional auctions have been proposed by Lochner et al. [39]. Since it is single-dimensional, bids are composed exclusively by prices. On the other hand, the framework Multiple criteria English Reverse Auctions (MERA) [7, 8] characterizes bids with vectors of attributes and criteria. MERA allows to represent English reverse auctions that differ in relation to the aggregation model of bidders' preferences and the information feedback provided to the participants.

## 2.2 Representation of negotiation protocols

A related problem to auction specification is the one of representing negotiation protocols. As noted by Meyer et al. [43] negotiation is investigated from different perspectives, such as economics, psychology, computer science and others. Consequently, there is no agreement in relation to its definition and the distinction between negotiation and auctions may be vague. Although some authors consider auctions as types of negotiation protocols, in this work we consider them related, but rather distinct, types of allocation procedures.

Following the definition from Chevaleyre et al. [15], by auctions we refer to centralized mechanisms that specify trades of items and payments based on the bidders' reported preferences. On the other hand, we consider that in negotiation protocols, allocations emerge as the result of a sequence of local negotiation steps. That is, they describe the negotiation over resources in a distributed setting. Another important difference is that negotiation protocols for exchanging goods may not be dependant on monetary transfer.

In their paper [3], Bădică et al. discuss how rule-based approaches can be used to automated negotiation. The paper focuses on experimental results for concurrent negotiation and do not address the generality of their approach. Huder et al. [25] propose a framework to enable negotiations according to bilateral and multilateral negotiation protocols using a meta-language to define protocols based on a set of attributes and parameters. In a subsequent work [24], the authors propose a language based on Extensible Markup Language (XML) for meta-negotiating the choice of negotiation protocols and for instantiating and parameterizing the system used for conducting the chosen protocol. Similarly, rule-markup languages have also been used for the declarative representation of negotiations [18].

Due to the lack of a precise semantics, the approaches considered up to here are too poor to enable reasoning. This limitation motivates the use of logic-based languages. In two papers [70, 71], Wooldridge and Parsons briefly compare the use of different languages for negotiation, including propositional logic, a language for electronic commerce and a negotiation meta-language. The problems considered involve simple protocols with multi-attribute negotiation of single goods. Even in this setting, the problems of determining if agreement has been reached and determining if a particular protocol will terminate

are computationally hard. A basic logical framework for negotiation has also been proposed by Meyer et al. [43]. In the two agents setting, the work explores modes of negotiation from which an agreement over an outcome can be reached when the participants are rational, cooperative and truthful.

### 2.3 GDL-based approaches

The *Game Description Language* (GDL) is the official language used for the *General Game Playing* (GGP) competition [21]. Due to its initial limitations, several extensions and variants have been successfully proposed, including GDL with imperfect information (GDL-II) [64] and introspection (GDL-III) [65] for specifying epistemic games. GDL has also been extended for defining, comparing and combining strategies in [77, 78]. This extension includes the dual connectives *prioritised disjunction* and *prioritised conjunction* for expressing preferences when combining strategies.

The use of GDL-based languages for describing market-based protocols have also been studied. De Jonge and Zhang [31] discuss the use of GDL for modeling negotiation. The main advantage is being able to apply the existing domain-independent techniques from GGP. For instance, the Monte Carlo Tree Search algorithm has been adapted for negotiations in non-zero-sum games [32]. In another paper [30], they propose the use of GDL as a unifying language for defining general and complex negotiation domains.

The closest contributions to ours are the Market Specification Language (MSL) [66] and ADL [45, 46], also based on GDL. Both works focus on representing single item auctions through a set of rules and then interpreting an auction-instance with the help of a state-based semantics. MSL is limited to single agent perspective while ADL is not. However, the main limit of both approaches is the lack of a clear link between the language, the mechanism formalization and the agents' preferences, which prevents the evaluation of GDL-based specifications as mechanisms. In this paper, we extend ADL [46] for enabling the integration with bidding language and going further on the representation of quantitative aspects.

### 2.4 Formal verification of auctions

To some extent, our work is also related to the literature of formal verification of auctions. Some works explore computed-aided verification [5, 12, 34], where the process is only assisted by a reasoner. There are also works that focus on a particular variant of an auction. For instance, model-checking approaches have been proposed for verifying suspect behaviors of bidders in concurrent online auctions [73, 74]. Based on Artifact Systems techniques, Belardinelli [6] explores the model checking of properties concerning the evolution of the bidding process in simultaneous ascending auctions. A similar approach investigates the verification of agent-based English auctions using temporal logic [2]. A more general concept has also been introduced [56, 69], where the authors advocate the use of Alternating-time Temporal Logic [1] to reason about economic mechanisms. Strategy Logic (SL) [14, 47] allows for explicit representation of strategies, which enables capturing solution concepts from Game Theory [52]. In [42], the authors investigate a quantitative version of SL with imperfect information for the automated verification of indirect mechanisms. However, as SL and ATL cannot model the internal structures of strategies, these languages are not fitted for reasoning about the actions that were actually executed in

a moment of the game, which is an important feature to be considered for General Game Playing.

In this paper, we show how ADL  $[\mathcal{F}_B]$  is suitable for the automated verification of direct revelation mechanisms. We encode individual rationality, efficiency and budget-balance as ADL  $[\mathcal{F}_B]$ -formulae. Verifying such properties amounts to model-checking ADL  $[\mathcal{F}_B]$ -formulae, which can be done in PTIME, when functions in  $\mathcal{F}_B$  can be computed in PTIME. Other properties can be analyzed via meta-reasoning and we exemplify this case with strategyproofness.

### 3 Auction description language with a set of functions $\mathcal{F}_B$

The Auction Description Language with a set of functions  $\mathcal{F}_B$  (ADL  $[\mathcal{F}_B]$ ) is a framework for specification of auction-based markets and it is composed by a state-transition model and a logical language. To encode an auction, we first define its signature, which specifies the participants (the agents), the goods being traded and the propositions and variables describing each state of the auction:

**Definition 1** An *auction signature*  $\mathcal{S}$  is a tuple  $(N, G, \mathcal{B}, \Phi, Y, I, \mathcal{F}_B)$ , where:

- $N = \{1, \dots, n\}$  is a nonempty finite set of *agents* (or *bidders*);
- $G = \{1, \dots, m\}$  is a finite set of good types;
- $\mathcal{B}$  is a nonempty finite set of *bids* (or *actions*);
- $\Phi$  is a finite set of atomic propositions specifying features of a state;
- $Y$  is a finite set of numerical variables specifying numerical features of a state;
- $I \subseteq \mathbb{Z}$  is an interval of integer numbers, denoting the value range for any countable component of the framework. We assume  $I$  is equipped with a partial order<sup>2</sup>  $\leq_I$ , capturing the standard less-than-or-equal relation over its elements. We let  $z_{\min}, z_{\max} \in \mathbb{Z}$  denote the minimum and maximum values in  $I$ , that is,  $z_{\min} \leq_I z$  and  $z \leq_I z_{\max}$  for each  $z \in I$ . By convenience, we assume  $0 \in I$  and denote  $I_{\geq 0} = [0, z_{\max}]$  and  $I_{> 0} = (0, z_{\max}]$  as the non-negative and positive parts of  $I$ , respectively. Similarly,  $I_{\leq 0} = [z_{\min}, 0]$  and  $I_{< 0} = [z_{\min}, 0)$  denote the non-positive and negative parts of  $I$ .
- $\mathcal{F}_B \subseteq \{f : \mathcal{B}^a \times I^b \rightarrow I \mid a, b \in I_{\geq 0}\}$  is a set of state-independent functions of possibly different arities.

Hereafter, we will fix an auction signature  $\mathcal{S}$  and all concepts will be based on this signature, except if stated otherwise. Note that  $z_{\min}$  and  $z_{\max}$ , in the definition of  $I$ , should be large enough to represent the total supply of goods being traded as well as the cumulative available money among agents. Through the rest of this paper, we assume that is the case. Any function value outside  $I$  is rounded to the nearest value  $z_{\min}$  or  $z_{\max}$ . We assume that  $\mathcal{F}_B$  contains the functions  $\text{sum}(z_1, z_2) = z_1 + z_2$ ,  $\text{sub}(z_1, z_2) = z_1 - z_2$ ,  $\text{times}(z_1, z_2) = z_1 \cdot z_2$ , denoting the addition, subtraction and multiplication of two integers  $z_1, z_2 \in I$ , respectively. Similarly, we also assume  $\mathcal{F}_B$  contains the functions

<sup>2</sup> A partial order is a relation that is reflexive, antisymmetric and transitive.

$$\max(z_1, z_2) = \begin{cases} z_1 & \text{if } z_2 \leq_1 z_1 \\ z_2 & \text{otherwise} \end{cases}$$

and

$$\min(z_1, z_2) = \begin{cases} z_1 & \text{if } z_1 \leq_1 z_2 \\ z_2 & \text{otherwise} \end{cases}$$

capturing the maximum and minimum among two integers  $z_1, z_2 \in I$ .

Given a list integers  $(z_i)_{i \in N} \in I^n$  and a 2-ary function  $f \in \mathcal{F}_B$  such that  $f : I \times I \rightarrow I$ , we will use the following shortcut:

$$f_{i \in N}(z_i) = f(z_1, f(z_2, f(\dots, f(z_{n-1}, z_n))))$$

A *trade* is a tuple  $(\lambda_{i,j})_{j \in G, i \in N} \in I^m$ , where  $\lambda_{i,j}$  denotes the number of units  $j$  being traded by agent  $i$ . A trade specifies which items each agent is selling and/or buying. A positive trade expresses how many units of a good type are purchased and a negative trade represents how many units are sold. Similarly, a positive payment denotes how much a buyer will pay and a negative payment expresses how much a seller will receive.

A tuple of objects indexed by agents in  $N$  is called a *profile*. In a profile, we may omit the index set and we write it in bold, e.g.,  $\lambda$  for  $(\lambda_i)_{i \in N}$  and  $(\lambda_j)_{j \in G}$  for  $(\lambda_{i,j})_{j \in G, i \in N}$ . Given a profile  $\sigma$ , we let  $\sigma_r$  be the component of agent  $r$  and  $\sigma_{-r}$  be  $(\sigma_i)_{i \neq r}$ .

Let us now present the model and the language’s syntax and semantics.

### 3.1 Syntax

Let  $\mathcal{L}_z$  be the set of numerical terms, with each  $z \in \mathcal{L}_z$  defined as follows:

$$z ::= x \mid y \mid f(\beta, \dots, \beta, z, \dots, z)$$

where  $\beta \in \mathcal{B}$  is an action,  $x \in I$  is an integer,  $y \in Y$  is a numerical variable and  $f \in \mathcal{F}_B$  is a function<sup>3</sup>.

The logical language for ADL  $[\mathcal{F}_B]$  is denoted by  $\mathcal{L}_{ADL[\mathcal{F}_B]}$  and a formula  $\varphi$  in  $\mathcal{L}_{ADL[\mathcal{F}_B]}$  is defined by the following Backus-Naur Form grammar:

$$\varphi ::= p \mid \textit{initial} \mid \textit{terminal} \mid \textit{legal}_i(\beta) \mid \textit{does}_i(\beta) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi \mid z \leq z$$

where  $p \in \Phi$  is a proposition,  $i \in N$  is an agent,  $\beta \in \mathcal{B}$  is an action and  $z \in \mathcal{L}_z$  is a numerical term.

Intuitively, *initial* and *terminal* specify the initial terminal states, resp.;  $\textit{legal}_i(\beta)$  asserts that agent  $i$  is allowed to take action  $\beta$  at the current state and  $\textit{does}_i(\beta)$  asserts that agent  $i$  takes action  $\beta$  at the current state. The formula  $\bigcirc\varphi$  means “ $\varphi$  holds at the next state”. The formula  $z_1 \leq z_2$  means that the numerical term  $z_1$  is smaller or equal to the numerical term  $z_2$ .

We use the standard abbreviations from propositional logic, such as  $\varphi \vee \psi$  for  $\neg(\neg\varphi \wedge \neg\psi)$ ,  $\varphi \rightarrow \psi$  for  $\neg\varphi \vee \psi$ , and  $\varphi \leftrightarrow \psi$  for  $(\varphi \rightarrow \psi) \wedge (\psi \rightarrow \varphi)$ . We take  $\top$  to be an abbreviation for some fixed propositional tautology such as  $p \vee \neg p$ , and let  $\perp$  be an

<sup>3</sup> Notice  $f(\cdot)$  may be 0-ary.



abbreviation of  $\neg T$ . We also use abbreviations for comparison operators, such as  $z_1 = z_2$  for  $z_1 \leq z_2 \wedge z_2 \leq z_1$ ,  $z_1 < z_2$  for  $z_1 \leq z_2 \wedge \neg(z_1 = z_2)$ ,  $z_1 \neq z_2$  for  $\neg(z_1 = z_2)$ ,  $z_1 \geq z_2$  for  $z_2 \leq z_1$  and  $z_1 > z_2$  for  $z_1 \geq z_2 \wedge z_1 \neq z_2$ . The extension of the comparison operators to multiple arguments is straightforward.

Assume the bids  $\beta = (\beta_i)_{i \in N}$ , where  $\beta_i \in \mathcal{B}$  is a bid associated to the agent  $i \in N$ . The formula  $does(\beta) =_{\text{def}} \bigwedge_{i \in N} does_i(\beta_i)$  represents that the agents in  $N$  perform the joint action  $\beta$ .

### 3.2 Semantics

The semantics of ADL  $[\mathcal{F}_B]$  are based on state-transition models, which allows us to represent the key aspects of an auction, at first the legal bids and the transitions among states.

**Definition 2** A state-transition-model (ST-model for short)  $M$  is a tuple  $(W, \bar{w}, T, L, U, \pi_\Phi, \pi_Y)$ , where:

- $W$  is a nonempty set of *states*;
- $\bar{w} \in W$  is the *initial state*;
- $T \subseteq W$  is a set of *terminal states*;
- $L \subseteq W \times N \times \mathcal{B}$  is a *legality* relation, describing the legal actions at each state, we let  $L(w, i) = \{a \in \mathcal{B} \mid (w, i, a) \in L\}$  be the set of all legal actions for agent  $i$  at state  $w$ ;
- $U : W \times \mathcal{B}^n \rightarrow W$  is an *update* function, given  $d \in \mathcal{B}^n$ , let  $d_i$  be the individual action for agent  $i$  in the joint action  $d$ ;
- $\pi_\Phi : W \rightarrow 2^\Phi$  is the valuation function for the state propositions;
- $\pi_Y : W \times Y \rightarrow I$ , is the valuation function for the numerical variables.

A *path* represents a run or execution of an auction protocol. Formally,

**Definition 3** Given an ST-model  $M = (W, \bar{w}, T, L, U, \pi_\Phi, \pi_Y)$ , a *path* is a sequence of states and joint actions  $w_0 \xrightarrow{d^1} w_1 \xrightarrow{d^2} \dots \xrightarrow{d^t} w_t \xrightarrow{d^{t+1}} \dots$  such that for any  $t \geq 1$ : (i)  $w_0 = \bar{w}$ ; (ii)  $w_t \neq w_0$ ; (iii)  $d_i^t \in L(w_{t-1}, i)$  for any  $i \in N$ , (iv)  $w_t = U(w_{t-1}, d^t)$ ; and (v) if  $w_{t-1} \in T$ , then  $w_{t-1} = w_t$ .

For any path  $\delta$ , let  $\delta[t]$  denote the  $t$ -th state of  $\delta$ ,  $\theta(\delta, t)$  denote the joint action performed at stage  $t$  of  $\delta$ ,  $\theta_i(\delta, t)$  denote the action of agent  $i$  performed at stage  $t$  of  $\delta$ , and  $\delta[0, t]$  denote the finite prefix  $\bar{w} \xrightarrow{d^1} w_1 \xrightarrow{d^2} \dots \xrightarrow{d^t} w_t$ . A path  $\delta$  is *complete* if  $\delta[e] \in T$ , for some  $e > 0$ . After reaching a terminal state  $\delta[e]$ , for any  $e' > e$ ,  $\delta[e'] = \delta[e]$ . Finally, for a given model  $M$ , any state  $w$  such that there exists a complete path  $\delta$  of  $M$  such that  $w \in \delta$  will be called a *reachable state* of  $M$ .

The semantics for ADL  $[\mathcal{F}_B]$  is given in two steps. First, we define function  $f_z$  to compute the meaning of numerical terms  $z \in \mathcal{L}_z$  in some specific state. Next, a formula  $\varphi \in \mathcal{L}_{ADL[\mathcal{F}_B]}$  is interpreted with respect to a stage in a path.

**Definition 4** Given an ST-model  $M$ , we define function  $f_z : \mathcal{L}_z \times W \rightarrow I$ , assigning any  $z \in \mathcal{L}_z$  and state  $w \in W$  to a number in  $I$ :

$$f_z(z, w) = \begin{cases} \pi_Y(w, z) & \text{if } z \in Y \\ f(\beta_1, \dots, \beta_n, f_z(z_1, w), \dots, f_z(z_m, w)) & \text{if } z = f(\beta_1, \dots, \beta_n, z_1, \dots, z_m) \\ z & \text{if } z \in I \end{cases}$$

where  $n, m \in \mathbb{I}_{\geq 0}$ .

**Definition 5** Let  $M$  be an ST-Model. Given a path  $\delta$  of  $M$ , a stage  $t$  on  $\delta$  and a formula  $\varphi \in \mathcal{L}_{ADL[\mathcal{F}_B]}$ , we say  $\varphi$  is true (or satisfied) at  $t$  of  $\delta$  under  $M$ , denoted by  $M, \delta, t \models \varphi$ , according to the following definition:

- $M, \delta, t \models p$  iff  $p \in \pi_\Phi(\delta[t])$
- $M, \delta, t \models \neg\varphi$  iff  $M, \delta, t \not\models \varphi$
- $M, \delta, t \models \varphi_1 \wedge \varphi_2$  iff  $M, \delta, t \models \varphi_1$  and  $M, \delta, t \models \varphi_2$
- $M, \delta, t \models \text{initial}$  iff  $\delta[t] = \bar{w}$
- $M, \delta, t \models \text{terminal}$  iff  $\delta[t] \in T$
- $M, \delta, t \models \text{legal}_i(\beta)$  iff  $\beta \in L(\delta[t], i)$
- $M, \delta, t \models \text{does}_i(\beta)$  iff  $\theta_i(\delta, t) = \beta$
- $M, \delta, t \models \bigcirc\varphi$  iff  $M, \delta, t + 1 \models \varphi$
- $M, \delta, t \models z_1 \leq z_2$  iff  $f_z(z_1, \delta[t]) \leq_I f_z(z_2, \delta[t])$

A formula  $\varphi$  is globally true through  $\delta$ , denoted by  $M, \delta \models \varphi$ , if  $M, \delta, t \models \varphi$  for any stage  $t$  of  $\delta$ . A formula  $\varphi$  is globally true in an ST-model  $M$ , written  $M \models \varphi$ , if  $M, \delta \models \varphi$  for all paths  $\delta$  in  $M$ . Finally, let  $\Sigma$  be a set of formulae in  $\mathcal{L}_{ADL[\mathcal{F}_B]}$ , then  $M$  is a model of  $\Sigma$  if  $M \models \varphi$  for all  $\varphi \in \Sigma$ .

Similar to Epistemic GDL [29], the following propositions hold:

**Proposition 1** Let  $M$  be an ST-model, for each agent  $i \in N$  and each action  $\beta \in \mathcal{B}$ ,

1.  $M \models \text{does}_i(\beta) \rightarrow \neg \text{does}_i(\beta')$ , for any  $\beta' \in \mathcal{B}$  such that  $\beta' \neq \beta$
2.  $M \models \bigvee_{\beta' \in \mathcal{B}} \text{does}_i(\beta')$
3.  $M \models \text{does}_i(\beta) \rightarrow \text{legal}_i(\beta)$
4.  $M \models \neg \bigcirc \text{initial}$
5.  $M \models \text{terminal} \wedge \varphi \rightarrow \bigcirc\varphi$ , for any  $\varphi \in \mathcal{L}_{ADL[\mathcal{F}_B]}$
6.  $M \models \text{initial} \rightarrow \neg \text{terminal}$

Statements 1 and 2 specify an agent performs exactly one action in each state. Furthermore, if she does an action, then it must be legal (Statement 3). As a consequence from the path construction, we have that no state can be followed by the initial one (Statement 4) and any formula that holds in a terminal state also holds in the subsequent state (Statement 5). Finally, if a state is the initial one then it is not a terminal state (Statement 6).

We also have tautologies related to the partial order  $\leq$ :

**Observation.** Let  $M$  be an ST-model,  $i \in N$  be an agent,  $z = (z_1, z_2, \dots, z_n)$  be a list of numerical terms (i.e.,  $z \in \mathcal{L}_z^n$ ) and  $f \in \mathcal{F}_B$  be any function such that  $f : \mathbb{I}^n \rightarrow \mathbb{I}$ , for some  $n \in \mathbb{I}_{>0}$ ,

1.  $M \models z_1 \mathfrak{R} z_1$ , for  $\mathfrak{R} \in \{\leq, \geq, =\}$
2.  $M \models z_1 \mathfrak{R} z_2 \wedge z_2 \mathfrak{R} z_1 \rightarrow z_1 = z_2$ , for  $\mathfrak{R} \in \{\leq, \geq, =\}$

3.  $M \models z_1 \mathfrak{R} z_2 \wedge z_2 \mathfrak{R} z_3 \rightarrow z_1 \mathfrak{R} z_3$ , for  $\mathfrak{R} \in \{\leq, \geq, =\}$
4.  $M \models z_i = z_j \leftrightarrow f(z_i, z_{-i}) = f(z_j, z_{-i})$ , for any  $1 \leq i \leq n$
5.  $M \models z_1 = z_2 \leftrightarrow z_2 = z_1$
6.  $M \models z_1 \mathfrak{R} z_2 \rightarrow \neg(z_2 \mathfrak{R} z_1)$ , for  $\mathfrak{R} \in \{<, >\}$

Statements 1, 2 and 3 are a consequence from  $\preceq_1$  being reflexive, antisymmetric and transitive, respectively. *Ceteris paribus*, two numerical terms are equal if and only if the value obtained after applying  $f$  is equal for both terms (Statement 4). Statement 5 says that the equality operator is symmetric and Statement 6 states that the operators representing the relations *smaller-than* and *greater-than* are asymmetric.

Requirements for representing verifiable auctions For verifying a protocol expressed in ADL  $[\mathcal{F}_B]$  with respect to properties from mechanism design, its signature  $S = (N, G, \mathcal{B}, \Phi, Y, I, \mathcal{F}_B)$  must comply with the following requirements:

- $\mathcal{F}_B$  should include a function  $v_i : \mathcal{B} \times I^{nm} \rightarrow I$  for each agent  $i \in N$ , where  $v_i(\beta, \lambda)$  denotes the value of  $\beta$  given a joint trade  $\lambda \in I^{nm}$ , i.e.,  $v_i(\beta, \lambda)$  represents the value reported for trade  $\lambda$  under  $i$ 's bid  $\beta$ ;
- There are no duplicate bids in  $\mathcal{B}$ , that is, there are no two bids  $\beta, \beta' \in \mathcal{B}$ , such that  $\beta \neq \beta'$  and  $v_i(\beta, \lambda) = v_i(\beta', \lambda)$ , for any trade  $\lambda \in I^{nm}$  and any agent  $i$ ;
- Each payment and trade should be represented as a numerical variable, that is,  $\{payment_i, trade_{i,j} : i \in N, j \in G\} \subseteq Y$ . The variables  $payment_i$  and  $trade_{i,j}$  denote the value in a state of agent  $i$ 's payment and her trade for the good  $j$ , respectively.

Other functions may as well be included in  $\mathcal{F}_B$ . For instance, for indirectly representing market clearing with ADL  $[\mathcal{F}_B]$ , one may encode a winner determination function, such that it assigns bids and allocations to trades. Such function is not a compulsory requirement for the bidding language, since we can also directly represent the market clearing through  $\mathcal{L}_{ADL[\mathcal{F}_B]}$ -rules.<sup>4</sup>

## 4 Evaluating auction-based protocols

In this section, we explore the general evaluation of auction-based protocols. First, we recall concepts from mechanism design and present their formulation in ADL  $[\mathcal{F}_B]$ . As for GDL, we then define well-formulated protocol descriptions.

### 4.1 Mechanism design

A mechanism aggregates agents' preferences and decides for an *outcome* (e.g., an allocation of goods, a result of an election, etc) [16]. Auctions are a type of mechanism, in which the outcome is described in terms of trades and monetary transfers among the participants. According to a given objective, the goal of Mechanism Design (MD) is to design a game (i.e., the mechanism) such that an outcome with desirable features is reached, despite the agents' self interests [62]. The objective of a mechanism can include, for instance,

<sup>4</sup> A winner determination function is shown in Sect. 6.1.2 and the protocol presented in Sect. 5 has the market clearing represented through  $\mathcal{L}_{ADL[\mathcal{F}_B]}$ -rules.

truthfulness of agents (i.e., strategyproofness), maximization of social welfare (i.e., efficiency), voluntary participation (i.e., individual rationality), and so on.

Let us recall concepts from MD and show how to represent some classical but important objectives (hereafter called *properties*) in ADL  $[\mathcal{F}_B]$ , namely budget-balance, efficiency, individual rationality and strategyproofness. As we focus on auctions, we denote the mechanism outcome as a pair  $(\lambda, \mathbf{p})$ , where  $\lambda = (\lambda_j)_{j \in G}$  is a trade and  $p_i$  is the payment for agent  $i$

The preference of an agent  $i$  in  $N$  over a trade  $\lambda$  is modeled by a preference function  $\vartheta_i : I^{nm} \rightarrow I$ , where  $\vartheta_i \in V_i$  and  $V_i$  is a finite set of possible preference functions for  $i$ . We call  $V_i$  the preference space of  $i$ . We classically assume that the utility of agent  $i$  over an outcome  $(\lambda, \mathbf{p})$  is quasi-linear (i.e., the utility's dependence on the payment is separable and linear)<sup>5</sup>, defined as  $\vartheta_i(\lambda) - p_i$ .

Since different action sets may have different expressiveness and syntactical structures, we use the value reported in a bid for a given trade to assess whether the bid represents the agent's preference function.

**Definition 6** Let  $\mathcal{B}$  be an action set and  $V_i$  be the preference space of agent  $i \in N$ . A bid  $\beta \in \mathcal{B}$  represents a preference function  $\vartheta_i \in V_i$ , denoted by  $\beta \sim_i \vartheta_i$ , iff  $v_i(\beta, \lambda) = \vartheta_i(\lambda)$ , for all trade  $\lambda \in I^{nm}$ .

Similarly, an action set may represent a preference space. In this case, exactly one bid in the action set should represent a preference function.

**Definition 7** Given a set of actions  $\mathcal{B}$  and a preference space  $V_i$  of agent  $i \in N$ , we say  $\mathcal{B}$  represents  $V_i$ , denoted by  $\mathcal{B} \approx_i V_i$  iff for each  $\vartheta_i \in V_i$  there exists a unique bid  $\beta \in \mathcal{B}$  such that  $\beta \sim_i \vartheta_i$  and for each bid  $\beta \in \mathcal{B}$  there exists a preference function  $\vartheta_i \in V_i$  such that  $\beta \sim_i \vartheta_i$ .

If  $\mathcal{B} \approx_i V_i$ , for each  $\vartheta_i \in V_i$  we let  $\beta_{\vartheta_i}$  denote the bid  $\beta \in \mathcal{B}$  such that  $\beta \sim_i \vartheta_i$ , that is,  $\beta_{\vartheta_i}$  is the bid that represents  $\vartheta_i$ . Given a preference profile  $\vartheta$ , let  $\beta_\vartheta = (\beta_{\vartheta_i})_{i \in N}$  denote the profile of bids representing  $\vartheta$ .

Notice not all elements of  $I$  are feasible values for trades, for instance in a traditional English auction, the trade for each agent should be either 0 or 1 while the interval  $I$  could include greater values for encoding the payments. In the following, the possible choices considered for trades in the mechanism are denoted  $\Lambda \subseteq I^{nm}$ .

An indirect mechanism describes the available actions for each agent and an outcome function that maps vectors of actions (also know as *strategies*) into outcomes. In a direct mechanism, each agent's available action consists on reporting preferences from her preference space [26]. Formally, a direct mechanism is defined as follows [52]:

**Definition 8** A direct mechanism  $(s, \mathbf{p})$  specifies a social choice function  $s : \prod_{i \in N} V_i \rightarrow \Lambda$  and a profile of payment functions  $\mathbf{p}$ , where  $p_i : \prod_{i \in N} V_i \rightarrow I$  denotes the amount agent  $i$  pays (or receives).

<sup>5</sup> Quasilinearity of utilities refers to the fact that the utility function is a linear combination of the preference valuation function and the price paid by the agent. However, the preference function  $\vartheta_i$  itself can be general.

Regardless whether a protocol is multi- or single-stage, we view each step  $w_e \xrightarrow{d^{e+1}} w_{e+1}$  of a path  $w_0 \xrightarrow{d^1} w_1 \xrightarrow{d^2} \dots \xrightarrow{d^t} w_t \xrightarrow{d^{t+1}} \dots$  in an ST-model  $M$  as a direct mechanism, where  $e \geq 0$  and the social choice and payments are encoded by the update and valuation functions in  $M$ . Formally,

**Definition 9** Given a preference space profile  $V$ , an ST-model  $M$  and a state  $w \in W$  such that  $L(w, i) \approx_i V_i$  for each  $i \in N$ , and let the set of possible trades be defined as  $\Lambda = \{(\pi_Y(w', trade_{i,j}))_{i \in N, j \in G} : w' = U(w, \beta_\vartheta) \ \& \ \vartheta \in \prod_{i \in N} V_i\}$ . Then state  $w$  is a direct mechanism  $(s, p)$ , where for each  $\vartheta \in \prod_{i \in N} V_i$ , the outcome is denoted by the valuation of the numerical variables regarding trades and payments in the state  $w' = U(w, \beta_\vartheta)$ . The social choice function is

$$s(\vartheta) = (\lambda_j)_{j \in G}$$

where  $\lambda_{i,j} = \pi_Y(w', trade_{i,j})$  for each agent  $i$  and good  $j$ . The payment function for agent  $i$  is

$$p_i(\vartheta) = \pi_Y(w', payment_i)$$

The state  $w' = U(w, \beta_\vartheta)$  in the above definition is called an *outcome state*. Any reachable state in an ST-model is an outcome state, except the initial state. For the next subsections, we fix an ST-Model  $M$  and a preference space profile  $V$  such that  $L(w, i) \approx_i V_i$  for each agent  $i$  and state  $w \in W$ .

### 4.1.1 Budget-balanced mechanisms

A mechanism is *strongly budget-balanced* (SBB) if the cumulative payment among the bidders is zero, for every preference they may have [44]. A mechanism where there is no monetary deficit, that is, where only the designer can earn revenue, is called *weakly budget-balanced* (WBB) [44]. This condition is a relaxation from SBB, where the cumulative payment among the bidders cannot be negative.

**Definition 10** A direct mechanism  $(s, p)$  is strongly budget-balanced (resp. weakly budget-balanced) if for each  $\vartheta \in \prod_{i \in N} V_i$ ,

$$\sum_{i \in N} p_i(\vartheta) = 0 \text{ (resp. } \sum_{i \in N} p_i(\vartheta) \geq 0)$$

We denote the condition of a state being SBB by the following ADL  $[F_B]$ -formula:

$$sbb =_{\text{def}} \text{sum}_{i \in N} (payment_i) = 0$$

The formula *wbb* is defined similarly, with  $\geq$  instead of  $=$ . Remind we consider that each stage in  $M$  represents a direct mechanism. The ST-model  $M$  is SBB (resp. WBB) if that is the case for all outcome states of all paths in  $M$ , that is, if  $M \models Osbb$  (resp.  $M \models Owbb$ ).

Similarly, we could represent conditions for the balance of trades. *Balance of supply-demand* requires the cumulative of trades for each good to be exactly zero [37]. Mechanisms with *free disposal* allow trades to sell more items than are purchased [54], that is, the cumulative of trades for each good must be at most zero.

### 4.1.2 Strategyproof mechanisms

A mechanism is *strategyproof* (SP), or incentive compatible, if each agent  $i$  prefers reporting her real preference  $\vartheta_i$  than reporting any other preference  $\vartheta'_i$ , since  $\vartheta_i$  gives her at least the same utility [52].

**Definition 11** A direct mechanism  $(s, p)$  is strategyproof if for every agent  $i \in N$ , every preference profile  $\vartheta$  and every  $\vartheta'_i \in V_i$

$$\vartheta_i(s(\vartheta)) - p_i(\vartheta) \geq \vartheta_i(s(\vartheta'_i, \vartheta_{-i})) - p_i(\vartheta'_i, \vartheta_{-i})$$

Now we reformulate this condition in terms of states from an ST-model. Let  $V_i$  denote the preference space for each agent  $i$ .

Given the ST-model  $M$ , a path  $\delta$  in  $M$  and a stage  $t \geq 0$  in  $\delta$ , such that  $L(\delta[t], i) \approx_i V_i$  for each  $i$ . For any preference profile  $\vartheta$ , let  $\delta_\vartheta$  denote a path such that  $\delta[0, t] = \delta_\vartheta[0, t]$  and  $\theta(\delta_\vartheta, t) = \beta_\vartheta$  (i.e.,  $M, \delta_\vartheta, t \models \text{does}(\beta_\vartheta)$ ). In other words,  $\delta_\vartheta$  is a path with the same prefix as  $\delta$ , but one in which agents report the preferences  $\vartheta$  in  $\delta_\vartheta[t]$  instead of the actions they perform in  $\delta[t]$ .

We say that  $\delta[t]$  is strategyproof if for every  $i \in N$ , every preference profile  $\vartheta$  and every  $\vartheta'_i \in V_i$ , we have that, for some  $x \in I$ ,

$$M, \delta_\vartheta, t \models \text{Osub}(v_i(\beta_{\vartheta_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) = x$$

and

$$M, \delta_{(\vartheta'_i, \vartheta_{-i})}, t \models \text{Osub}(v_i(\beta_{\vartheta_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) \leq x$$

$M$  is strategyproof if each stage  $t \geq 0$  of each path  $\delta$  in  $M$  is strategyproof.

### 4.1.3 Efficient mechanisms

A mechanism is *efficient* (EF) if the social choice function maximizes the (utilitarian) social welfare [54], i.e., the cumulative preference among the agents.

**Definition 12** A direct mechanism  $(s, p)$  is efficient if for every preference profile  $\vartheta$ ,

$$\sum_{i \in N} \vartheta_i(s(\vartheta)) = \max_{\lambda \in \Lambda} \sum_{i \in N} \vartheta_i(\lambda)$$

Let us express this condition in terms of an ST-model  $M$ . The following formula determines whether the current trade maximizes the social welfare:

$$\text{eff}(\beta) =_{\text{def}} \left( \text{sum}_{i \in N}(v_i(\beta_i, (\text{trade}_j)_{j \in G})) = \max_{\lambda \in \Lambda} (\text{sum}_{i \in N}(v_i(\beta_i, \lambda))) \right)$$

We say  $M$  is EF if, after performing a joint action, the trade in the outcome state maximizes agents' preferences, that is  $M \models \text{does}(\beta) \rightarrow \text{Oeff}(\beta)$ , for every  $\beta \in \mathcal{B}^n$ .

#### 4.1.4 Individually rational mechanisms

A mechanism is (ex-post) *individually rational* (IR), if agents always get non-negative utility [52]. Given a reported preference, in an IR mechanism, the agent's utility when participating is at least as good as if she did not participate (assuming the utility of non-participation is zero). Individual rationality is also known as voluntary participation since it expresses the idea that agents are not forced to participate in the mechanism [54].

**Definition 13** A direct mechanism  $(s, p)$  is individually rational if for every agent  $i \in N$ , every  $\vartheta \in \prod_{r \in N} V_r$

$$\vartheta_i(s(\vartheta)) - p(\vartheta) \geq 0$$

We use the following ADL  $[\mathcal{F}_B]$ -formula to denote whether an state is IR:

$$ir(\beta) =_{\text{def}} \bigwedge_{i \in N} \text{sub}(v_i(\beta_i, (\text{trade}_j)_{j \in G}), \text{payment}_i) \geq 0$$

The ST-model  $M$  is IR if performing a joint action leads to an individually rational state, that is,  $M \models \text{does}(\beta) \rightarrow \bigcirc ir(\beta)$ , for every  $\beta \in \mathcal{B}^n$ .

The properties described in this section are classical in mechanism design, since they describe desirable features of the outcome. The objective of a mechanism may include a combination of different properties. However, well-known impossibility results restrict the feasible combination of such properties: no mechanism can be efficient, strongly budget-balanced and individual-rational [49] and no mechanism can be efficient, incentive compatible and strongly budget-balanced [23].

In this paper, we verify ST-models by considering that each stage is a direct mechanism, that is, an iterative protocol is treated as a sequence of (independent) direct mechanisms. The revelation principle [52] states that any indirect mechanism that implements a function in dominant strategies can be converted into a strategyproof direct mechanism. For this reason, considering direct mechanisms is of first interest.

If we want to verify an ST-model  $M$  as an (unique) indirect mechanism, we need to evaluate properties considering the final outcome, that is, on the terminal states of each path. The classical approach in mechanism design requires properties to hold in strategic equilibrium rather than for all possible outcomes. As ADL  $[\mathcal{F}_B]$  does not involve quantification over strategies, we need meta-reasoning to capture the strategic equilibrium for a given solution concept (such as Nash or dominant strategy equilibrium). The reader may refer to [52] for a discussion on the problem of finding strategic equilibria.

In the next subsection, we will focus on properties that ensure well-formed descriptions in ADL  $[\mathcal{F}_B]$ . This properties are not related to the mechanism outcome but require protocols to be playable and eventually end.

## 4.2 Characterizing well-formed protocols

Love et al. [40] introduced constraints for games used in General Game Playing. This constraints constitute desirable features of games described in GDL by ensuring their descriptions to be meaningful (or *well-formed*), in the sense that games are playable, eventually terminate and are weakly winnable by any player.

We rephrase the constraints for termination and playability in terms of ST-models. First, termination refers to whether each path from an ST-model reaches a terminal state.

**Definition 14** An ST-model  $M$  *terminates* if each path  $\delta$  in  $M$  is complete, that is,  $\delta[t] \in T$ , for some  $t > 0$ .

Playability means there is a legal action for each agent to take in each moment of the auction.

**Definition 15** An ST-model  $M$  is *playable* if  $L(w, i) \neq \emptyset$  for each *reachable* state  $w \in W$  and agent  $i \in N$ .

Weak winnability means that, for each agent, there is a sequence of joint actions that leads to a terminal state where the goal value is maximal. In the logical formulations of GDL, weak winnability means that every player has a chance to win [60, 75], that is, there exists a path to a winning state. In  $ADL[\mathcal{F}_B]$ , weak winnability follows from our assumption that each stage represents a direct mechanism. Since we understand the agent's preferences as represented by legal actions, there exists a joint action in each stage leading to a state that maximizes her utility among the possible outcomes. Hence, we do not focus on weak winnability for determining whether a protocol is well-formed.

A well-formed protocol is a set of rules in  $ADL[\mathcal{F}_B]$  whose model is an ST-model that satisfies both termination and playability.

**Definition 16** Given an ST-Model  $M$  and a finite set of  $ADL[\mathcal{F}_B]$ -formulae  $\Sigma \subset \mathcal{L}_{ADL[\mathcal{F}_B]}$ ,  $\Sigma$  is a *well-formed* protocol over  $M$  if  $M$  is a model of  $\Sigma$ ,  $M$  terminates and it is playable.

It is possible to have different descriptions that are well-formed with respect to the same model (e.g., due to redundancy). In fact, a given auction description may also be sound in respect to several models. Investigating minimal descriptions is an interesting non trivial open question. A potential path is to characterize the minimum equivalent of an original ST-model, that is, the canonical model. This problem was explored for GDL [27], where they use the notion of bisimulation equivalence between ST-models. In a recent paper, Zhang [76] investigates the equivalence of two GDL-descriptions when they describe games behaviourally the same.

Similar to  $ADL[\mathcal{F}_B]$ , the semantics of GDL is based on fixed paths. Thus, the constraints for well-formed descriptions cannot be encoded through formulae using the standard formulation of GDL. Zhang [75] proposes a GDL-based modal logic to enable reasoning over game descriptions, with which one can express conditions such as playability, termination and winnability. However, the work does not investigate the complexity of verifying formulae in this modal logic. Ruan et al. [60] use ATL [1] to reason about GDL-specified games. They prove that the problem of interpreting ATL formulae over propositional GDL descriptions is EXPTIME-complete and show how to use ATL for the verification of well-formedness conditions, which might or might not hold on various games. Deciding whether a GDL description [40] is well-formed is undecidable in general, since deciding whether a description leads to games that always terminates would solve the halting problem for a Turing machine [61].

$ADL[\mathcal{F}_B]$  is useful for representing the rules of an auction as well as for verifying a number of properties. As we discussed above, we can encode different properties of direct mechanisms



as  $\text{ADL}[\mathcal{F}_B]$ -formulae (e.g., individual rationality and budget balance). Although we cannot represent strategyproofness and the constraints for well-formed descriptions entirely as  $\text{ADL}[\mathcal{F}_B]$ -formulae, we are still able to infer them by meta-reasoning over the model specification.

Other logical languages are suitable for encoding properties. Termination and playability, for instance, can be written as ATL-formulae [1]. As SL [14] includes quantification over strategies, it allows the evaluation of games in strategic equilibria. Therefore, it can be used to encode properties for indirect mechanisms as logical formulae, as proposed in [42]. However, the expressivity of such languages brings a computational cost. The model-checking problem for ATL\* with perfect information is in PSPACE in the memoryless case and deterministic double exponential time with perfect recall [63]. As for SL, the model-checking is Non Elementary with respect to the size of the specification. More specifically, it is  $k$ -EXPSpace-hard in the alternation number  $k$  of quantifications in the specification [48]. As we shall see in Sect. 7, the model-checking for  $\text{ADL}[\mathcal{F}_B]$  is in PTIME in the size of the formula. For this reason, we believe  $\text{ADL}[\mathcal{F}_B]$  provides a reasonable cost-benefit for expressing and evaluating general auctions. As a drawback for ATL, SL and  $\text{ADL}[\mathcal{F}_B]$ , representing auctions as concurrent game structures or state-transition models may require exponential size.

In the next sections, we illustrate the use of  $\text{ADL}[\mathcal{F}_B]$  for specification of different although representative types of auctions: a simultaneous ascending auction, a Vickrey–Clarke–Groves mechanism and an iterative combinatorial exchange.

## 5 Representing a simultaneous ascending auction

Let us now consider the simultaneous ascending auction (SAA), which is a single-side and single-unit auction similar to the traditional English auction, except that several goods are sold at the same time, and that the participants simultaneously bid for any number of goods they want. According to Cramton [17]:

“The simultaneous ascending auction (and its variants) will remain the best method for auctioning many related items in a wide range of circumstances, even settings where some of the goods are complements for some bidders.”

To represent a SAA with  $n$  types of goods and  $m$  agents, we first describe the auction signature, written  $\mathcal{S}_{sa} = (\mathbf{N}, \mathbf{G}, \mathcal{B}, \Phi, \mathbf{Y}, \mathbf{I}, \mathcal{F}_B)$ , where  $\mathbf{N} = \{1, \dots, m\}$ ,  $\mathbf{G} = \{1, \dots, n\}$ ,  $\Phi = \{\text{sold}_j, \text{bid}_{i,j} : j \in \mathbf{G} \ \& \ i \in \mathbf{N}\}$ ,  $\mathbf{Y} = \{\text{price}, \text{price}_j, \text{trade}_{i,j}, \text{payment}_i : j \in \mathbf{G} \ \& \ i \in \mathbf{N}\}$  and  $\mathbf{I} \subset \mathbb{N}$ . The propositions  $\text{sold}_j$  and  $\text{bid}_{i,j}$  represent whether the good  $j$  was sold and whether  $i$  is bidding for  $j$ , resp. The variables  $\text{price}$  and  $\text{price}_j$  specify the current price for any unsold good and the selling price for  $j$ , resp. Agents may specify the value they are willing to pay for each good in a given state. The action set is defined as follows:  $\mathcal{B} = \{(p_1, \dots, p_m) : p_j \in \mathbb{I}_{\geq 0}, 1 \leq j \leq m\}$ , where  $p_j$  denotes the price for good  $j$ .

$\mathcal{F}_B$  includes the functions previously introduced (e.g.,  $\text{sum}(z_1, z_2)$ ,  $\text{max}(z_1, z_2)$ ). It also contains the function  $v_i : \mathcal{B} \times \mathbf{I}^{nm} \rightarrow \mathbf{I}$ , for each agent  $i \in \mathbf{N}$ . This function is defined as follows:

$$v_i((p_1, \dots, p_m), \lambda) = \sum_{j \in \mathbf{G}} \lambda_{i,j} \cdot p_j$$

for a trade  $\lambda \in \mathbf{I}^{nm}$  and a bid  $(p_1, \dots, p_m) \in \mathcal{B}$ .

Each instance of a SAA is specific and defined with respect to  $\mathcal{B}$ ,  $\mathbf{I}$  and the constant values  $\text{inc}, m, n \in \mathbb{I}_{>0}$  and  $\text{start} \in \mathbb{I}_{\geq 0}$ , representing the quantity of agents and types of

1.  $initial \leftrightarrow price = start \wedge \bigwedge_{j \in G} (price_j = start \wedge \bigwedge_{i \in N} (\neg bid_{i,j} \wedge trade_{i,j} = 0))$
2.  $sold_j \leftrightarrow \bigvee_{i \in N} trade_{i,j} = 1$ , for each  $j \in G$
3.  $terminal \leftrightarrow \neg initial \wedge \bigwedge_{j \in G} (sold_j \vee \bigwedge_{i \in N} \neg bid_{i,j})$
4.  $\bigcirc(trade_{i,j} = 1 \leftrightarrow bid_{i,j} \wedge \bigwedge_{r \in N \setminus \{i\}} \neg bid_{r,j})$ , for each  $i \in N, j \in G$
5.  $\bigcirc(trade_{i,j} = 0 \leftrightarrow \neg(bid_{i,j} \wedge \bigwedge_{r \in N \setminus \{i\}} \neg bid_{r,j}))$ , for each  $i \in N, j \in G$
6.  $legal_i(p_1, \dots, p_m) \leftrightarrow \bigwedge_{j \in G} ((p_j = 0 \wedge trade_{i,j} = 0) \vee (p_j = \text{sum}(price, inc) \wedge \neg sold_j) \vee (p_j = price_j \wedge trade_{i,j} = 1))$ , for each  $i \in N, p_1, \dots, p_m \in \{x : 0 \leq x < z_{max} - inc\}$
7.  $\neg terminal \wedge price = x \rightarrow \bigcirc price = \text{sum}(x, inc)$ , for each  $x \in I_{\geq 0}$
8.  $\neg terminal \wedge price_j = x \rightarrow \bigcirc((price_j = x \wedge sold_j) \vee (price_j = \text{sum}(x, inc) \wedge \neg sold_j))$ , for each  $j \in G, x \in I_{\geq 0}$
9.  $\bigcirc bid_{i,j} \leftrightarrow (does_i(p_1, \dots, p_m) \wedge p_j \neq 0) \vee (bid_{i,j} \wedge terminal)$ , for each  $i \in N, j \in G$  and some  $p_1, \dots, p_m \in I_{\geq 0}$
10.  $payment_i = \text{sum}_{j \in G} (\text{times}(price_j, trade_{i,j}))$ , for each  $i \in N$

Fig. 1 Simultaneous Ascending Auction represented by  $\Sigma_{sa}$

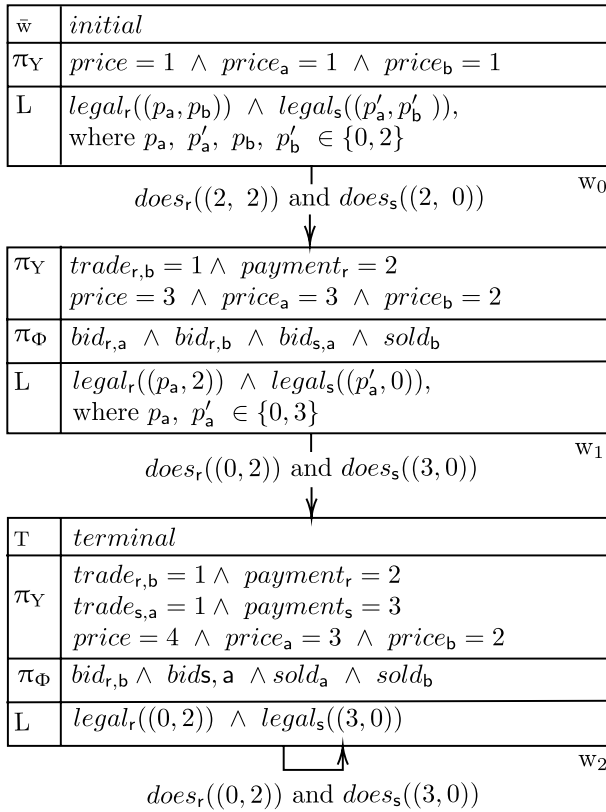
goods, the bid increment, and the starting price, respectively. Then, the rules of an SAA are formulated by ADL [ $\mathcal{F}_B$ ]-formulae as shown in Fig. 1.

In the initial state, no agent is bidding, no trade is performed and the prices have the value *start* (Rule 1). A good is sold if it is traded to some agent (Rule 2). In a terminal state, all the goods are either sold or no one is bidding for them (Rule 3). A good will be traded to an agent in the next state if she is currently the only active bidder for this item, otherwise there is no trade (Rules 4–5). For each good, an agent can either bid the value 0, an increment on the current price (for unsold goods) or repeat her winning bid for this good (Rule 6). In a non-terminal state, the propositions and numerical variables are updated as follows: (i) the current price increases, (ii) the selling price increases for unsold goods, and (iii) the active bidders for each good are updated with respect to their bids (Rules 7–9). The payment for an agent is the cumulative value of the selling price for her traded goods (Rule 10). Let  $\Sigma_{sa}$  be the set of Rules 1–10.

### 5.1 Representing as a model

Next, we address the model representation of SAA. Let  $\mathcal{M}_{sa}$  be the set of ST-models  $M_{sa}$  defined for any constant values  $start \in I_{\geq 0}$  and  $inc, n, m \in I_{> 0}$ . Each  $M_{sa}$  is defined as follows:

- $W = \{\langle (b_j)_{j \in G}, (\lambda_j)_{j \in G}, p, (p_j)_{j \in G} \rangle : b_{i,j}, \lambda_{i,j} \in \{0, 1\} \ \& \ p, p_j \in I_{\geq 0} \ \& \ i \in N \ \& \ j \in G\}$ , where  $b_{i,j}$  denotes whether agent  $i$  is bidding for good  $j$ ,  $\lambda_{i,j}$  specifies the number of goods with type  $j$  traded for agent  $i$ ,  $p$  denotes the current price and  $p_j$  represents the selling price for  $j$ ;
- $\bar{w} = \langle 0, \dots, 0, 0, \dots, 0, start, start, \dots, start \rangle$ , in the initial state, there is no trade or active agent and the prices are *start*;



**Fig. 2** A Path in  $M_{sa}$ , with 2 bidders and 2 goods

- $T = \{w : w = \langle (b_j)_{j \in G}, (\lambda_j)_{j \in G}, p, (p_j)_{j \in G} \rangle \in W \setminus \{\bar{w}\} \ \& \ \text{for all } j \in G, \text{ either (i) } \lambda_{i,j} = 1 \text{ for some } i \in N \text{ or (ii) } bid_{i,j} = 0, \text{ for all } i \in N\}$ , the terminal states are the ones where every good was sold or there is no bidder interested on purchasing it;
- $L = \{(w, i, (pr_j)_{j \in G}) : i \in N \ \& \ w = \langle (b_j)_{j \in G}, (\lambda_j)_{j \in G}, p, (p_j)_{j \in G} \rangle \in W \ \& \ \text{for all } j \in G, \text{ and all } 0 \leq pr_j < z_{max} - inc \text{ such that either (i) } pr_j = 0 \ \& \ \lambda_{i,j} = 0 \text{ or (ii) } pr_j = p + inc \ \& \ \lambda_{r,j} \neq 1, \text{ for all } r \in N \text{ or (iii) } pr_j = p_j \ \& \ \lambda_{i,j} = 1\}$ , that is, agents can choose to raise their bid or to give up of unsold goods, if an agent bought a good, she must keep her bid for it;
- For every  $w = \langle (b_j)_{j \in G}, (\lambda_j)_{j \in G}, p, (p_j)_{j \in G} \rangle$  in  $W$  and all  $d \in \mathcal{B}^m$ ,  $U$  is defined as follows: if  $w \in T$  then  $U(w, d) = w$ . Otherwise,  $U(w, d) = \langle (b'_j)_{j \in G}, (\lambda'_j)_{j \in G}, p', (\lambda_j)_{j \in G} \rangle$ , where for every  $i \in N$  and  $j \in G$  each component is updated as follows,
  - (i)  $b'_{i,j} = 1$  iff  $d_i \neq 0$ ; and  $b'_{i,j} = 0$  otherwise;
  - (ii)  $\lambda'_{i,j} = 1$  iff  $b'_{i,j} = 1$  and for all  $r \in N \setminus \{i\}, b'_{i,j} \neq 1$ ; and  $\lambda'_{i,j} = 0$  otherwise;
  - (iii)  $p' = p + inc$ ;
  - (iv)  $p'_j = p_j + inc$  iff  $\lambda'_{r,j} = 0$  for all  $r \in N$ ; and  $p'_j = p_j$  otherwise.
- For each  $w = \langle (b_j)_{j \in G}, (\lambda_j)_{j \in G}, p, (p_j)_{j \in G} \rangle$  in  $W, i \in N$  and  $j \in G$ ,

- (i)  $\pi_Y(w, trade_{i,j}) = \lambda_{i,j}$ ;
  - (ii)  $\pi_Y(w, price) = p$ ;
  - (iii)  $\pi_Y(w, price_j) = p_j$ ;
  - (iv)  $\pi_Y(w, payment_i) = \sum_{j \in G} p_j \cdot \lambda_{i,j}$
- For each  $w \in W$ ,  $\pi_\Phi(w) = \{sold_j : \lambda_{i,j} = 1 \ \& \ j \in G \ \& \ i \in N\} \cup \{bid_{i,j} : b_{i,j} = 1 \ \& \ j \in G \ \& \ i \in N\}$ .

Hereafter, we assume an instance of  $M_{sa} \in \mathcal{M}_{sa}$  and  $\Sigma_{sa}$  for some  $start \in I_{\geq 0}$  and  $inc, n, m \in I_{> 0}$ .

**Example 1** Let  $M_{sa} \in \mathcal{M}_{sa}$ , such that  $start = 1$  and  $inc = 1$ . We assume there are two agents in  $N$ , denoted by  $i$  and  $s$ , and two types of goods in  $G$ , denoted by  $a$  and  $b$ . Figure 2 illustrates a path in  $M_{sa}$ , showing the value of the numerical variables and the propositions that hold in each state. For convenience, we omit the numerical variables when their value is 0. In state  $w_0$ , agents  $r$  and  $s$  bid for good  $a$ , but only agent  $r$  bids for good  $b$ . In state  $w_1$ , since  $r$  is the only bidder for  $b$ , it is sold to her. Agent  $r$  cannot change her bid for  $b$  and  $s$  can no longer bid for it. In  $w_1$ , only agent  $s$  accepts to increase her bid for  $a$ . In state  $w_2$ ,  $a$  is sold to  $s$ . Since all the goods were sold, this state is terminal.

### 5.2 Evaluating the protocol

Let us now evaluate the protocol. First, we show that  $\Sigma_{sa}$  is a sound representation of  $M_{sa}$ .

**Lemma 1**  $M_{sa}$  is an ST-model and it is a model of  $\Sigma_{sa}$ .

Next, we show that no good can be bought by two different agents, i.e., given any two agents and a good, one of them will have her trade equal to zero. When a good is sold, it will still be sold in the next state.

**Proposition 2** For each  $j \in G$  and each  $i, r \in N$  such that  $i \neq r$ ,

1.  $M_{sa} \models trade_{i,j} = 0 \vee trade_{r,j} = 0$
2.  $M_{sa} \models sold_j \rightarrow \bigcirc sold_j$
3.  $M_{sa} \models \neg sold_j \rightarrow price = price_j$

Each path in  $M_{sa}$  reaches a terminal state, and thus the protocol satisfies the termination condition. Furthermore,  $M_{sa}$  satisfies playability, that is, there is always a legal action for each agent to take. Thus,  $\Sigma_{sa}$  is well-formed over  $M_{sa}$ .

**Theorem 1**  $\Sigma_{sa}$  is a well-formed protocol over the ST-model  $M_{sa}$ .

**Proof** Since  $M_{sa}$  is a model of  $\Sigma_{sa}$  (see Lemma 1), we have to show that for each path  $\delta$  in  $M_{sa}$  and each agent  $i \in N$ ,

1.  $\delta$  is a complete path;
2.  $M_{sa} \models \bigvee_{a \in B} legal_i(a)$ .

Let us start by verifying Statement 1. Remind  $start \in I_{\geq 0}$  and  $inc \in I_{> 0}$ . Let  $\delta$  be a path in  $M_{sa}$ . In  $\delta[0]$ ,  $\pi_Y(\delta[0], price) = start$ . By the update function, for any stage  $t$ , if  $\delta[t] \notin T$ , then  $\pi_Y(\delta[t + 1], price) = \pi_Y(\delta[t], price) + inc$ .

For the sake of contradiction, let us assume  $\delta$  is not complete. Let  $i \in N$  be an agent. By the definition of  $L$ ,  $(p_1, \dots, p_m) \in L(\delta[t], i)$ , for all  $0 \leq p_j < z_{max} - inc$  and  $j \in G$ , such that either (i)  $p_j = 0$  &  $\pi_Y(\delta[t], trade_{i,j}) = 0$ , or (ii)  $p_j = price + inc$  &  $\pi_Y(\delta[t], trade_{r,j}) = 0$  for all  $r \in N$ , or (iii)  $p_j = price_j$  &  $\pi_Y(\delta[t], trade_{r,j}) = 0$ . Since  $\pi_Y(\delta[t + 1], price) > \pi_Y(\delta[t], price)$ , there will be a stage  $e \geq 0$  in  $\delta$ , where the condition (ii) will not be true for any  $0 \leq p_j < z_{max} - inc$ . Thus, for each good  $j$ , it will be the case that  $i$  bids 0 for it or the good was assigned to her (i.e.,  $\pi_Y(\delta[e], trade_{i,j}) = 1$ ). From Rules 3 and 9 in  $\Sigma_{sa}$ , it follows that  $\delta[e + 1] \in T$ . Thus,  $\delta$  is a complete path, which is a contradiction.

We now consider Statement 2. Given a path  $\delta$  in  $M_{sa}$  and a stage  $t$  in  $\delta$ , we show that there is a legal action for agent  $i$  in  $\delta[t]$ . For each  $j \in G$ , let  $p_j = 0$  if  $\pi_Y(\delta[t], trade_{i,j}) = 1$ . Otherwise, let  $p_j = \pi_Y(\delta[t], price_j)$ . By  $L$  definition, we have  $(p_1, \dots, p_m) \in L(\delta[t], i)$ . Thus,  $M_{sa}, \delta, t \models \bigvee_{a \in B} legal_i(a)$ .

From being a single-side auction where all agents are buyers, it follows that there is no monetary deficit in  $M_{sa}$ , but it is not strongly budget-balanced.

**Proposition 3**  $M_{sa} \models \text{owbb}$  and  $M_{sa} \not\models \text{osbb}$

The simultaneous ascending auction is only efficient on states preceding the terminal one.

**Proposition 4** Given a joint action  $\beta \in B^n$ ,

1.  $M_{sa} \not\models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$
2.  $M_{sa} \models \text{does}(\beta) \rightarrow \text{O}(\text{terminal} \rightarrow \text{ef}(\beta))$

The auction described by  $M_{sa}$  is individually rational, since agents pay at most their bids.

**Theorem 2** Given a joint action  $\beta \in B^n$ ,  $M_{sa} \models \text{does}(\beta) \rightarrow \text{Oir}(\beta)$

**Proof** Given a path  $\delta$  in  $M_{sa}$  and a stage  $t$ , assume  $M_{sa}, \delta, t \models \text{does}(\beta)$  for some  $\beta \in B^n$ . We consider first the case where  $\delta[t] \notin T$ . Let  $i \in N$  be an agent and  $(p_1, \dots, p_m)$  denote the bid of  $i$  in the joint action  $\beta$ . Let us consider the good  $j \in G$ . We denote by  $pr_j = \pi_Y(\delta[t + 1], price_j)$  the price of good  $j$  in  $\delta[t + 1]$  and by  $\lambda = (\pi_Y(\delta[t + 1], trade_{s,j}))_{j \in G, s \in N}$  the trade in  $\delta[t + 1]$ . Recall function  $v_i((p_1, \dots, p_m), \lambda) = \sum_{j \in G} \lambda_{i,j} \cdot p_j$ . Similarly, by Rule 10, we have that the payment for agent  $i$  in  $\delta[t + 1]$  is  $\pi_Y(\delta[t + 1], payment_i) = \sum_{j \in G} pr_j \cdot \lambda_{i,j}$ .

We have  $\theta_i(\delta, t) = (p_1, \dots, p_m)$ . Notice  $inc, p_j \in I_{\geq 0}$  by the definition of  $S_{sa}$ . According with the action legality (Rule 6), the value of  $p_j$  can be either zero, the price of  $j$  in  $\delta[t]$  or

the current price incremented by  $inc$ . For each of the three cases, we show that the part of  $i$ 's payment corresponding to  $j$  is equal to the part of  $v_i(p_1, \dots, p_m, \lambda)$  corresponding to  $j$ :

- If  $p_j = 0$ , agent  $i$  chosen to not bid for good  $j$ . By Rules 5 and 9,  $M_{sa}, \delta, t + 1 \models \neg bid_{i,j} \wedge trade_{i,j} = 0$ . Thus, we have  $\lambda_{i,j} \cdot p_j = pr_j \cdot \lambda_{i,j} = 0$ .
- If  $p_j = \pi_V(\delta[t], price_j)$ , by the definition of  $L$ , it must be the case that the good was already sold to agent  $i$ , that is  $M_{sa}, \delta, t \models trade_{i,j} = 1 \wedge sold_j$ . Since good  $j$  is sold, Rule 6 ensure other agents can only bid the value 0 for  $j$ , and thus,  $M_{sa}, \delta, t + 1 \models \bigwedge_{r \in N \setminus \{i\}} \neg bid_{r,j}$ . By Rules 4 and 9, we have  $M_{sa}, \delta, t + 1 \models bid_{i,j} \wedge trade_{i,j} = 1$ . Since good  $j$  is sold, its price in  $\delta[t]$  is the same as in  $\delta[t]$  (Rule 8), that is  $M_{sa}, \delta, t \models price_j = p_j$ . Thus, we have  $\lambda_{i,j} \cdot p_j = pr_j \cdot \lambda_{i,j} = pr_j$ .
- Let  $prevprice = \pi_V(\delta[t], price)$ . If  $p_j = prevprice + inc$ , then  $M_{sa}, \delta, t + 1 \models bid_{i,j}$ . From the legality definition, it should be the case that  $j$  is not sold, that is  $M_{sa}, \delta, t \models \neg sold_j \wedge price_j = sum(prevprice_j, inc)$ . From Statement 3 of Proposition 2, we have  $M_{sa}, \delta, t \models price = price_j$ . The value of  $trade_{i,j}$  depends on the actions of other agents in  $\delta[t]$ . From Rules 5 and 4, we have  $M_{sa}, \delta, t + 1 \models trade_{i,j} = 0 \vee trade_{i,j} = 1$ . In the first case,  $\lambda_{i,j} \cdot p_j = pr_j \cdot \lambda_{i,j} = 0$ . Otherwise, the trade has the value 1 and  $\lambda_{i,j} \cdot p_j = pr_j \cdot \lambda_{i,j} = pr_j$ , since  $p_j = pr_j = prevprice + inc$ .

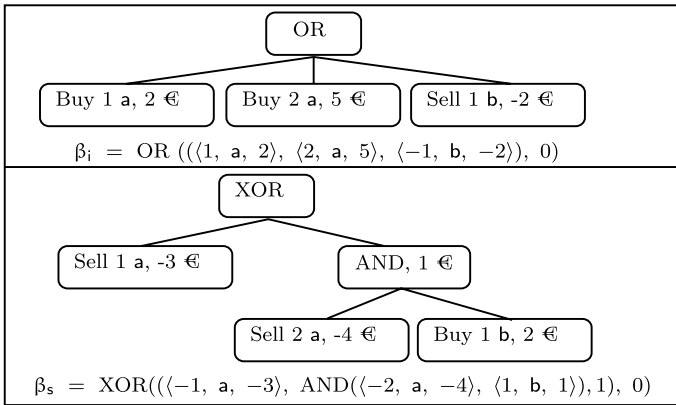
It follows that  $M_{sa}, \delta, t + 1 \models v_r(\beta, (trade_j)_{j \in G}) = payment_r$  and  $M_{sa}, \delta, t + 1 \models \text{sub}(v_r(\beta, (trade_j)_{j \in G}), payment_r) \geq 0$ . Thus,  $M_{sa}, \delta, t \models \text{Oir}(\beta)$ .

If  $\delta[t] \in T$ , the loop in the path definition ensures  $M_{sa}, \delta, t \models \text{Oir}(\beta)$  if and only if  $M_{sa}, \delta, t \models ir(\beta)$ .

Under the assumption that each stage in  $M_{sa}$  is a (direct) mechanism for which the legality set represents the agents' preference spaces, the  $M_{sa}$  is strategyproof. As it is only legal to accept or decline to raise the current price for unsold goods (represented by bidding the value 0 or  $price + inc$ ), there is no utility improvement if the agent accepts when she would prefer to decline (and vice-versa). When a good is sold for an agent, there is only one value that is legal to bid for the good bought, and thus the agent cannot strategize.

**Proposition 5**  $M_{sa}$  is strategyproof.

We conclude the section by discussing variants of non-combinatorial auctions. When the number of good types is one (that is,  $|G| = 1$ ),  $\Sigma_{sa}$  corresponds to the Japanese-English auction. For representing the standard variant of the English auction, one should define the legality rule such that agents are also allowed to bid any price above the current price. Furthermore, the price in the next state should be updated according to the highest bid in the current one. The Dutch auction is also similar to  $\Sigma_{sa}$ . The key difference is that in the Dutch auction the bidding value should be decreased at each round until at least one agent accepts to pay the current price. As we saw in this section, with the  $ADL[\mathcal{F}_B]$  description of a given auction, we are able to formally analyze it, both in relation to domain-specific properties, well-formedness of its protocol and from a mechanism design perspective.



**Fig. 3** Examples of tree-bids  $\beta_i$  and  $\beta_s$  reported by agents  $i$  and  $s$ , resp

## 6 Representing combinatorial exchange

We now consider two protocols for combinatorial exchange: a one-shot protocol and a multi-stage variant. We consider the setting with multiple goods and multiple copies of each good. Agents hold an initial allocation of goods and can trade items with each other. Remind a trade denotes the number of goods being exchanged among the agents in a state. With *allocation* we refer to the number of goods the agents initially have.

Both protocols use the Tree-Based Bidding Language (TBBL) for defining its action set. TBBL [41, 53] is a language designed for Combinatorial Exchange. It allows to represent buyers and sellers demands in the same structure. We adopt TBBL as it is a highly expressive and compact language. TBBL is general enough to represent any kind of utility function (full expressivity), like OR-like bidding languages [9]. It is even more expressive in the sense that it is able to mix preferences for buying and selling bundles in the same framework. In relation to which kind of utility functions this framework is able to represent concisely, [13] compares TBBL with XOR and OR\* bidding languages with this respect, and shows that TBBL is more compact, in the sense that there are valuation functions that admit an exponentially larger representation in these latter languages than in TBBL.

### 6.1 Tree-based bidding language

The bidding language we present in this section, denoted  $\mathcal{L}_{TBBL}$ , only differs from the original definition of TBBL in the fact that we assume all language components and related optimization problems are bounded by  $I$ .

**Definition 17** A formula in  $\mathcal{L}_{TBBL}$  is called a *bid-tree* (or simply a *bid*) and is generated by the following BNF:

$$\beta ::= \langle z, j, z \rangle \mid IC_y^x(\bar{\beta}, z)$$

where  $\bar{\beta} ::= \bar{\beta}, \beta \mid \beta$  is a nonempty *bid list*,  $j \in G$ ,  $z \in I$ ,  $y \leq x$  and  $y, x \in I_{\geq 0}$ .

A bid in the form  $\langle q, j, v \rangle$  is called a leaf and represents that the agent is willing to buy (or sell)  $q$  units of the good  $j$  and pay (or receive)  $v$ . The *interval-choose* (IC) operator defines a range on the number of child nodes that must be satisfied. Thus, a bid  $IC_y^x(\bar{\beta}, v)$  indicates the agent is willing to pay (or receive)  $v$  for the satisfaction of at least  $y$  and at most  $x$  of the children nodes  $\bar{\beta}$ . The IC operator can express logical connectors. For instance,  $IC_1^1(\bar{\beta}, v)$  is equivalent to the XOR operator applied to the bids in the list  $\bar{\beta}$ . Let  $z = |\bar{\beta}|$  (i.e., the list size),  $IC_z^z(\bar{\beta}, v)$  is equivalent to an AND operator and  $IC_z^1(\bar{\beta}, v)$  is equivalent to an OR operator. For simplicity, we hereafter use the corresponding shortcuts  $XOR(\bar{\beta}, v)$ ,  $AND(\bar{\beta}, v)$  and  $OR(\bar{\beta}, v)$ .

For instance, in Fig. 3, agent  $i$  reports her willingness to buy 1 unit of a paying 2€ or 2 units of a for 5€ or to sell 1 unit of b receiving 2€. Agent  $s$  bids an exclusive disjunction for either (i) to sell one unit of a and receive 3€; or (ii) to sell 2 units of a receiving 4€ and to buy one unit of b paying 2€. The node representing the condition (ii) has an additional value of 1.

Hereafter, we introduce some extra notations to characterize solutions and winners. Let  $\beta_i \in \mathcal{L}_{TBBL}$  be a bid-tree from bidder  $i$ , the set  $Node(\beta_i)$  denotes all nodes in the tree  $\beta_i$ , that is, all its inner bids, including  $\beta_i$  itself. Formally, if  $\beta_i$  is in the form  $\langle q, j, v \rangle$ , then  $Node(\beta_i) = \{\beta_i\}$ . Otherwise,  $\beta_i$  is in the form  $IC_y^x(\bar{\beta}, v')$  and  $Node(\beta_i) = \{\beta_i\} \cup Node(\bar{\beta}_1) \cup \dots \cup Node(\bar{\beta}_z)$ , where  $z = |\bar{\beta}|$  and  $\bar{\beta}_k$  is the  $k$ -th element of  $\bar{\beta}$ .

Let  $\alpha \in Node(\beta_i)$ , the set  $Child(\alpha) \subset Node(\beta_i)$  denotes the children of node  $\alpha$ . If  $\alpha$  is in the form  $IC_y^x(\bar{\beta}, v)$ , then  $Child(\alpha) = \{\bar{\beta}_1, \dots, \bar{\beta}_z\}$ , where  $z = |\bar{\beta}|$ . Otherwise,  $Child(\alpha) = \{\}$ . The leaves of a bid-tree  $\beta_i$  are denoted by  $Leaf(\beta_i) = \{\langle q, j, v \rangle \in Node(\beta_i) : q \in I, v \in I \ \& \ j \in G\}$ . The value specified at node  $\alpha$  is denoted by  $b_i(\alpha) \in I$ . If  $\alpha$  is in the form  $\langle q, j, v \rangle$ , then  $b_i(\alpha) = v$ . Otherwise,  $\alpha$  is in the form  $IC_y^x(\bar{\beta}, v')$  and  $b_i(\alpha) = v'$ . Finally, the quantity of units of the good  $j$  specified at a leaf  $\alpha = \langle q, j, v \rangle$  is denoted by  $q(\alpha, j) = q$ . For any other  $j' \neq j \in G$ ,  $q(\alpha, j') = 0$ . For any node  $\alpha \notin Leaf(\beta_i)$  and  $j \in G$ ,  $q(\alpha, j) = 0$ . If  $\alpha$  is not a leaf (i.e.,  $\alpha \in Node(\beta_i) \setminus Leaf(\beta_i)$ ), then it is in the form  $IC_y^x(\bar{\beta})$  and we denote by  $x_\beta$  and  $y_\beta$  the interval-choose constraints  $x$  and  $y$ , respec.

### 6.1.1 Trade value and valid solutions

Given a bid-tree  $\beta_i$  from agent  $i$ , the value of a trade  $\lambda \in I^{nm}$  is defined as the sum of the values in all satisfied nodes, where the set of satisfied nodes is chosen to provide the *maximal* total value. Let  $sat_i(\alpha) \in \{0, 1\}$  denote whether a node  $\alpha \in Node(\beta_i)$  is satisfied and  $sat_i = \{\alpha : sat_i(\alpha) = 1, \text{ for all } \alpha \in Node(\beta_i)\}$  denote the nodes satisfied in a solution.

A solution  $sat_i$  is valid for a tree  $\beta_i$  and trade  $\lambda_i$ , written  $sat_i \in valid(\beta_i, \lambda_i)$  if the following rules R1 and R2 hold [41]:

$$x_\beta sat_i(\alpha) \leq \sum_{\gamma \in Child(\alpha)} sat_i(\gamma) \leq y_\beta sat_i(\alpha) \tag{R1}$$

$$\forall \alpha \in Node(\beta_i) \setminus Leaf(\beta_i)$$

$$\sum_{\alpha \in Leaf(\beta_i)} q_i(\alpha, j) sat_i(\alpha) \leq \lambda_{i,j}, \forall j \in G \tag{R2}$$

Rule R1 ensures that no more and no less than the appropriate number of children are satisfied for any node that is satisfied. Rule R2 requires that the total increase in quantity of each item across all satisfied leaves is no greater than the total number of units traded.



The total value of trade  $\lambda \in I^m$  for agent  $i$ , given her bid  $\beta_i$ , is defined as the solution to the following problem:

$$v_i(\beta_i, \lambda) = \max_{\text{sat}_i} \sum_{\beta \in \text{Node}(\beta_i)} b_i(\beta) \cdot \text{sat}_i(\beta)$$

s.t. rules  $R1, R2$  hold

### 6.1.2 Winner determination

Given an auction signature, the bid-trees  $\beta = (\beta_i)_{i \in N}$  and an allocation  $X = (x_{i,j})_{j \in G, i \in N}$ , where  $\beta_i \in \mathcal{L}_{\text{TBBL}}$  denotes the bid from agent  $i \in N$  and  $x_{i,j} \in I_{\geq 0}$  represents how many copies of  $j$  agent  $i$  initially holds.

**Definition 18** The winner determination (WD) defines a pair  $(\lambda, \text{sat})$  obtained by the solution to the following mixed-integer program [41]:

$$\text{WD}(\beta, X) : \max_{\lambda, \text{sat}} \sum_{i \in N} \sum_{\beta \in \text{Node}(\beta_i)} b_i(\beta) \cdot \text{sat}_i(\beta) \tag{C1}$$

$$\begin{aligned} \text{s.t. } \lambda_{i,j} + x_{i,j} &\geq 0, \forall i \in N, j \in G \\ \sum_{i \in N} \lambda_{i,j} &\leq 0, \forall j \in G \end{aligned} \tag{C2}$$

$$\text{sat}_i \in \text{valid}(\beta_i, \lambda_i), \forall i \in N \tag{C3}$$

$$\text{sat}_i(\beta) \in \{0, 1\}, \lambda_{i,j} \in I \tag{C4}$$

where  $\text{sat} = (\text{sat}_i)_{i \in N}$ . Constraint **C1** ensures that the trade  $\lambda$  is *feasible* given  $X$ , that is, no agent sells more items than she initially hold. Constraint **C2** provides free disposal and allows trades to sell more items than are purchased (but not vice-versa). Constraint **C3** ensures that each trade for an agent  $i$  is valid given her bid-tree. Constraint **C4** defines the range for trades and node satisfaction. We denote by  $\text{WD}_\lambda(\beta, X)$  a function that obtains the trade  $\lambda$  in the solution  $\text{WD}(\beta, X) = (\lambda, \text{sat})$ . Similarly,  $\text{WD}_{\lambda_{i,j}}(\beta, X)$  captures the number of units of  $j$  traded by agent  $i$  in  $\text{WD}_\lambda(\beta, X)$ .

If there are two or more solutions for  $\text{WD}(\beta, X)$ , the trade  $\text{WD}_\lambda(\beta, X)$  will be chosen w.r.t. some total order among the elements of  $I^m$ . This tie-breaking order is omitted to avoid overloading the notation. In the examples, we assume this order is compatible with the Pareto dominance relation [67].

We denote  $\text{noop} =_{\text{def}} \langle 0, j, 0 \rangle$  as the action of not bidding, for some arbitrary  $j \in G$ .

**Lemma 2** For each agent  $i \in N$ , each bid-tree  $\beta \in \mathcal{L}_{\text{TBBL}}$  and each  $\lambda \in I^m$ ,  $v_i(\text{noop}, \lambda) = 0$ .

Next, we illustrate how to represent protocols with TBBL in ADL  $[\mathcal{F}_B]$ .

1.  $initial \rightarrow bidRound \wedge \bigwedge_{i \in N} (payment_i = 0 \wedge \bigwedge_{j \in G} trade_{i,j} = 0)$
2.  $terminal \leftrightarrow \neg initial$
3.  $terminal \rightarrow legal_i(noop)$ , for each  $i \in N$
4.  $initial \rightarrow legal_i(\beta)$ , for each  $i \in N, \beta \in \mathcal{B}$
5.  $does(\beta) \wedge initial \rightarrow \bigcirc(\bigwedge_{i \in N, j \in G} trade_{i,j} = WD_{\lambda_{i,j}}(\beta, X))$ , for each  $\beta \in \mathcal{B}^n$
6.  $does(\beta) \wedge \neg terminal \wedge p_i = sub(\sum_{r \in N \setminus \{i\}} (v_r(\beta_r, WD_{\lambda}^{-i}(\beta, X))), \sum_{r \in N \setminus \{i\}} (v_r(\beta_r, WD_{\lambda}(\beta, X)))) \rightarrow \bigcirc payment_i = p_i$ , for each  $p_i \in I, \beta \in \mathcal{B}^n$  and  $i \in N$
7.  $\bigcirc \neg bidRound$

Fig. 4 Vickrey–Clarke–Groves mechanism represented by  $\Sigma_{vcg}$

### 6.2 Vickrey–Clarke–Groves mechanism

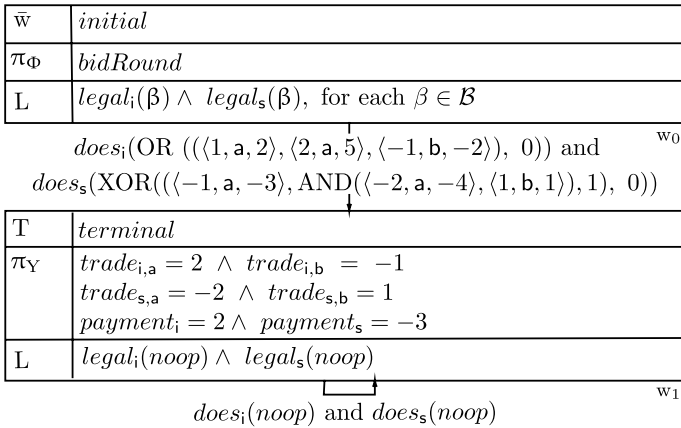
Using TBBL to determinate the action set, let us now represent the Vickrey–Clarke–Groves (VCG) mechanism in ADL  $[\mathcal{F}_B]$ . This mechanism chooses the outcome maximizing the reported preferences [36]. Each agent’s payment corresponds to the damage she causes the other players, that is, the difference in the social welfare of others with and without her participation [52]. We detail the rules specification and the semantic representation. Then, we revisit the conditions from MD and evaluate whether the protocol is well-formed.

To represent a VCG mechanism in the combinatorial exchange setting, we first describe its signature, written  $\mathcal{S}_{vcg} = (N, G, \mathcal{B}, \Phi, Y, I, \mathcal{F}_B)$ , where  $N = \{1, \dots, n\}$ ,  $G = \{1, \dots, m\}$ ,  $\mathcal{B} \subseteq \mathcal{L}_{TBBL}$ ,  $\Phi = \{bidRound\}$ ,  $Y = \{trade_{i,j}, payment_i : i \in N, j \in G\}$ , and  $I \subseteq \mathbb{Z}$ . Finally,  $\mathcal{F}_B$  contains the functions  $v_i(\beta, \lambda)$ ,  $WD_{\lambda}(\beta, X)$  and  $WD_{\lambda_{i,j}}(\beta, X)$  described in the previous section as well as the functions denoting basic mathematical operations (e.g.,  $sum(z_1, z_2)$ ). We also assume  $\mathcal{F}_B$  contains the function  $WD_{\lambda}^{-r} : \mathcal{B}^n \times I^{nm} \rightarrow I^{nm}$  for any two agents  $i$  and  $r$ .  $WD_{\lambda}^{-r}$  is defined exactly like  $WD_{\lambda}$  except that the set  $N$  in the winner determination (see Def. 18) is replaced by  $N \setminus \{r\}$  and that the resulting trade for agent  $r$  and each good  $j$  is equal to zero.

Each instance of a VCG is specific and is defined with respect to  $\mathcal{B}, I$  and the constant values  $n, m \in I_{>0}$  (the size of  $N$  and  $G$ , resp.), and  $X = (x_j)_{j \in G}$ , where  $x_{i,j} \in I_{\geq 0}$ , for each  $i \in N$  and  $j \in G$ . Each constant  $x_{i,j}$  represents the number of units of  $j$  initially held by agent  $i$ . The rules of VCG are represented by ADL  $[\mathcal{F}_B]$ -formulae as shown in Fig. 4.

In the initial state, the trade and payment are zero for every agent and good (Rule 1). Any state that is not initial is terminal (Rule 2). The proposition  $bidRound$  helps to distinguish the initial state from the terminal state where no trade or payment were assigned to any agent (e.g. when all agents bid noop). Once in a terminal state, players can only do noop. Otherwise, they can bid any  $\beta \in \mathcal{B}$  (Rules 3 and 4). After performing a joint bid, in the next state each agent receives a trade for each good, which is assigned by the winner determination over the initial allocation and their bids (Rule 5). After a joint bid in the initial state, the payment for agent  $i$  will be the difference in the others’ welfare with and without her participation (Rule 6). Finally, the proposition  $bidRound$  is always false in the next state (Rule 7).

Notice we could as well represent winner determination explicitly in ADL  $[\mathcal{F}_B]$  by capturing the trade maximizing the social welfare among all trades that satisfy constraints C1–C4. For instance, constraint C1 can be written in  $\mathcal{L}_{ADL[\mathcal{F}_B]}$  as



**Fig. 5** A Path in  $M_{vcg}$ , with 2 bidders and 2 goods

$feasible((\lambda_j)_{j \in G}) =_{def} \bigwedge_{i \in N, j \in G} \text{sum}(\lambda_{i,j}, x_{i,j}) \geq 0$ . We illustrate VCG with an indirect representation of the winner determination for succinctness and clarity of  $\Sigma_{vcg}$ .

### 6.2.1 Representing as a model

Next, we address the model representation. Let  $\mathcal{M}_{vcg}$  be the set of ST-models  $M_{vcg}$  defined for any  $\mathcal{B} \subseteq \mathcal{L}_{TBBL}$ ,  $I \subset \mathbb{Z}$ , and the constants  $n, m \in I_{>0}$  and  $X = (x_j)_{j \in G}$ , where  $x_{i,j} \in I_{\geq 0}$ , for each  $i \in N$  and  $j \in G$ . Each  $M_{vcg}$  is defined as follows:

- $W = \{ \langle b, (\lambda_j)_{j \in G}, p \rangle : b \in \{0, 1\} \ \& \ p_i, \lambda_{i,j} \in I \ \& \ i \in N \ \& \ j \in G \}$ ;
- $\bar{w} = \langle 1, 0, \dots, 0, 0, \dots, 0 \rangle$ ;
- $T = W \setminus \{ \bar{w} \}$ ;
- $L = \{ (w, i, noop) : i \in N \ \& \ w \in T \} \cup \{ (\bar{w}, i, \beta) : \beta \in \mathcal{B} \ \& \ i \in N \}$ ;
- $U$  is defined as follows: for all  $w = \langle b, (\lambda_j)_{j \in G}, p \rangle \in W$  and for all  $d \in \mathcal{B}^m$ :
- If  $w = \bar{w}$ , then  $U(w, d) = \langle 0, (\lambda'_j)_{j \in G}, p' \rangle$ , where each component is updated as follows, for each  $i \in N$  and  $j \in G$ . The number of units  $j$  traded for agent  $i$  is given by the winner determination:  $\lambda'_{i,j} = WD_{\lambda_{i,j}}(d, X)$ . The payment for  $i$  is the difference between the social welfare of others with and without  $i$ 's participation:

$$p'_i = \sum_{r \in N \setminus \{i\}} v_r(d_r, WD_{\lambda}^{-i}(d, X)) - \sum_{r \in N \setminus \{i\}} v_r(d_r, WD_{\lambda}(d, X))$$

- Otherwise,  $U(w, d) = w$ .
- For each  $w \in W$ ,  $i \in N$  and  $j \in G$ ,  $\pi_Y(w, trade_{i,j}) = \lambda_{i,j}$ ;  $\pi_Y(w, payment_i) = p_i$ ; and  $\pi_\Phi(w) = \{ bidRound : b = 1 \}$ .

Hereafter, we assume an instance of  $M_{vcg} \in \mathcal{M}_{vcg}$  and  $\Sigma_{vcg}$  for some  $\mathcal{B} \subseteq \mathcal{L}_{TBBL}$ ,  $I \subset \mathbb{Z}$ ,  $n, m \in I_{>0}$  and  $x_{i,j} \in I_{\geq 0}$ , where  $i \in N$ ,  $j \in G$ .

**Example 2** Let  $M_{vcg} \in \mathcal{M}_{vcg}$  such that the sets of agents and goods are the same from Example 1 and the initial allocation is as follows:  $x_{i,a} = 0$ ,  $x_{i,b} = 1$ ,  $x_{s,a} = 2$  and  $x_{s,b} = 0$  (i.e., at the beginning of the auction, agent  $i$  has 1 unit of  $b$  and agent  $s$  has 2 units of  $a$ ). Figure 5 illustrates a path in  $M_{vcg}$ , where the agents perform the bids previously introduced in Fig. 3. In state  $w_0$ , all the payments and trades are zero. Their joint bid leads to state  $w_1$ , where the trade obtained by the winner determination is  $(2, -1, -2, 1)$ . The tie-breaking ensures that the trade is unique. Thus, in  $w_1$  agent  $i$  has 2 units of  $a$  and agent  $s$  has 1 unit of  $b$ . Since  $w_1$  is terminal, the agents can only bid noop.

### 6.2.2 Evaluating the protocol

Let us now evaluate the protocol representing a VCG mechanism. First, Lemma 3 shows that  $M_{vcg}$  is a sound representation of  $\Sigma_{vcg}$ .

**Lemma 3**  $M_{vcg}$  is an ST-model and it is a model of  $\Sigma_{vcg}$ .

Next, we show  $\Sigma_{vcg}$  is a well-formed protocol, that is, each path in  $M_{vcg}$  reaches a terminal state and there is a legal action for each agent in all reachable states.

**Theorem 3**  $\Sigma_{vcg}$  is a well-formed protocol over the ST-model  $M_{vcg}$ .

**Proof** Since  $M_{vcg}$  is a model of  $\Sigma_{vcg}$  (see Lemma 3), we have to show that for each path  $\delta$  in  $M_{vcg}$  and each agent  $i \in N$ ,

1.  $\delta$  is a complete path;
2.  $M_{vcg} \models \bigvee_{a \in B} legal_i(a)$ .

Given a path  $\delta$  in  $M_{vcg}$  and a stage  $t$  of  $\delta$ . Let us verify Statement 1. We show that  $M_{vcg} \models initial \rightarrow \bigcirc terminal$ . Assume  $M_{vcg}, \delta, t \models initial$ . Then,  $\delta[t] = \bar{w}$ . By the path definition, for any  $j \geq 1$ ,  $\delta[j] \neq \bar{w}$ . By the construction of  $T$ , we have  $T = W \setminus \{\bar{w}\}$ . Thus,  $M_{vcg}, \delta, t + 1 \models terminal$  and  $M_{vcg}, \delta, t \models \bigcirc terminal$ .

Statement 2 is straightforward from Rules 3 and 4 from  $\Sigma_{vcg}$ .

The next lemma shows that if an agent bids noop in an initial state, her payment will be zero. Furthermore, if the payment is zero in a terminal state, it will be zero in the succeeding state.

**Lemma 4** For each agent  $i \in N$ ,  $M_{vcg} \models initial \wedge does_i(noop) \rightarrow \bigcirc payment_i = 0 \wedge \bigwedge_{j \in G} trade_{i,j} \geq 0$

We then focus on properties from Mechanism Design, that is budget balance, individual rationality, efficiency and strategyproofness. These results for VCG have already been proved [36, 52] and here we show how they are rephrased and verified with ADL  $[\mathcal{F}_B]$ . First, due to the VCG payments,  $M_{vcg}$  is not budget balanced.

**Proposition 6**  $M_{vcg} \not\models \bigcirc sbb$  and  $M_{vcg} \not\models \bigcirc wbb$ .

After the agents report their preferences,  $M_{vcg}$  chooses the trade maximizing the social welfare (i.e., the cumulative of values for the trade given the agents' bids).

**Proposition 7** *Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{vcg} \models does(\beta) \rightarrow Oef(\beta)$*

The VCG mechanism is individually rational when the agents' preferences over trades are non-negative [52].

**Proposition 8** *Given a joint action  $\beta \in \mathcal{B}^n$ , if  $\vartheta_i(\lambda) \geq 0$  for all  $\lambda \in I^{nm}$ ,  $\vartheta_i \in V_i$  and  $i \in N$ , then  $M_{vcg} \models does(\beta) \rightarrow Oir(\beta)$ .*

The VCG mechanism is also strategyproof [52], because the bid of an agent does not influence her payment.

**Theorem 4**  *$M_{vcg}$  is strategyproof.*

**Proof** Given a path  $\delta$  in  $M_{vcg}$  and a stage  $t \geq 0$  in  $\delta$ , such that  $L(\delta[t], i) \approx_i V_i$ . Since  $M_{vcg} \models initial \rightarrow Oterminal$  and  $\delta[t + 1] = \delta[t]$  whenever  $\delta[t] \in T$ , it suffices to show consider the case where  $t = 0$ .

Let  $\vartheta \in \prod_{i \in N} V_i$  be a preference profile and  $\delta_\vartheta$  denote a path such that  $\delta[0] = \delta_\vartheta[0]$  and  $\theta(\delta_\vartheta, 0) = \beta_\vartheta$ . We let  $\vartheta'_i \in V_i$  denote a preference of agent  $i$ ,  $\vartheta' = (\vartheta'_i, \vartheta_{-i})$  and  $\delta_{\vartheta'}$  be a path such that  $\delta[0] = \delta_{\vartheta'}[0]$ ,  $\theta_i(\delta_{\vartheta'}, 0) = \beta_{\vartheta'_i}$  and  $\theta_r(\delta_{\vartheta'}, 0) = \beta_{\vartheta_r}$  for each agent  $r \neq i$ . That is,  $M_{vcg}, \delta_\vartheta, 0 \models does(\beta_\vartheta)$  and  $M_{vcg}, \delta_{\vartheta'}, 0 \not\models does_i(\beta_{\vartheta'_i}) \wedge \bigwedge_{r \in N \setminus \{i\}} does_r(\beta_{\vartheta_r})$ .

Let  $u_{\vartheta_i} \in I$  such that  $M_{vcg}, \delta_\vartheta, 0 \models Osub(v_i(\beta_{\vartheta_i}, (trade_j)_{j \in G}), payment_i) = u_{\vartheta_i}$  holds. As we saw in the proof of Proposition 8, agent  $i$ 's utility in  $\delta_\vartheta[1]$  is simply

$$u_{\vartheta_i} = \sum_{r \in N} v_r(\beta_{\vartheta_r}, \lambda) - \sum_{r \in N \setminus \{i\}} v_r(\beta_{\vartheta_r}, \lambda^{-i})$$

where  $\lambda = (\lambda_j)_{j \in G}$  denote the trade performed in  $\delta_\vartheta[1]$  with  $\lambda_{i,j} = \pi_Y(\delta_\vartheta[1], trade) = WD_{\lambda_{i,j}}(\beta, X)$  for each good  $j$  and agent  $i$ , and  $\lambda^{-i} = WD_{\lambda}^{-i}(\beta, X)$  is the trade that would happen if  $i$  did not participate in the auction.

Notice that the bid of  $i$  has no impact in  $\sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i})$  by the definition of  $WD_{\lambda}^{-i}$ . Thus, it means that the bid maximizing  $i$ 's utility is the one that maximize

$$\sum_{r \in N} v_r(\beta_r, \lambda)$$

which, by definition, is the case when she bids truthfully. That is,  $u_{\vartheta_i}$  is the maximum utility  $i$  obtains in the succeeding stages of all paths starting in  $\delta[0]$ . Thus,  $M_{vcg}, \delta_\vartheta, 0 \models Osub(v_i(\beta_{\vartheta_i}, (trade_j)_{j \in G}), payment_i) = u_{\vartheta_i}$  and for any  $\vartheta'$ ,  $M_{vcg}, \delta_{\vartheta'}, 0 \models Osub(v_i(\beta_{\vartheta'_i}, (trade_j)_{j \in G}), payment_i) \leq u_{\vartheta_i}$ .

We conclude the section by discussing variants of combinatorial exchange. In combinatorial auctions, there are two types of participants: buyers and sellers. The main difference to a combinatorial exchange is that buyers can only demand for non-negative quantities and prices while sellers can only ask for non-positive quantities and prices. These restrictions can be easily encoded in ADL  $[F_B]$  by including a proposition for denoting the participants' types and defining legality rules based on their types. In Sect. 5, we exemplify the

representation of a single-sided auction. Restricting the number of participants with the type buyer (or similarly seller) to one is an alternative way of encoding single-sided auctions. Multi-unit auctions with single items can be represented for different market structures (e.g., the single-sided setting) by restricting the number of good types to one (that is,  $|G| = 1$ ). In the next section, we show how to represent an iterative protocol in ADL  $[\mathcal{F}_B]$  using a slightly different version of TBBL.

### 6.3 Iterative combinatorial exchange

We conclude the section on combinatorial exchange by considering an iterative protocol, denoted ICE. The protocol is a simplified and first-price version of the mechanism presented in [53]. In this auction, bidders report an interval of prices they are willing to pay (or receive) for a trade. For ensuring termination, agents need to refine their bids, that is, to shrink the value interval reported in their previous bid. For expressing such conditions, we consider the TBBL extension with value bounds, denoted  $\mathcal{L}_{TBBL+}$ . In this variant proposed by Lubin et al. [41], agents report a pair of valuation bounds in each node of their bid. The syntax of  $\mathcal{L}_{TBBL+}$  is obtained by replacing the value  $v$  of a leaf node  $\langle q, j, v \rangle$  by the lower bound  $\underline{v} \in I$  and upper bound  $\bar{v} \in I$  with  $\underline{v} \leq \bar{v}$ . The bounds  $\underline{v}$  and  $\bar{v}$  denote the minimum and maximum price the bidder considers acceptable to pay (or receive) for  $q$  units of  $j$ , respectively. Bid nodes with IC operators are similarly updated to include value bounds.

Given an agent  $i \in N$ , a bid  $\beta \in \mathcal{L}_{TBBL+}$  and a trade  $\lambda \in I^{nm}$ , the functions  $\underline{v}_i(\beta, \lambda)$ ,  $\bar{v}_i(\beta, \lambda)$ ,  $\underline{b}_i(\beta)$ ,  $\bar{b}_i(\beta)$  are defined with the same semantics from the functions  $v_i$  and  $b_i$  introduced in Sect. 6. Since the lower bound is no greater than the upper bound, we have that  $\underline{b}_i(\beta) \leq \bar{b}_i(\beta)$  and  $\underline{v}_i(\beta, \lambda) \leq \bar{v}_i(\beta, \lambda)$ . For representing this ICE protocol with ADL $[\mathcal{F}_B]$ , we first fix a constant  $\varepsilon \in [0, 1]$  for estimating the weight of the bounds in the value of a bid given a trade (i.e., function  $v_i$ ). If  $\varepsilon = 1$ , the bid value is based only on its lower bounds. Likewise, the upper bounds determinate the bid value when  $\varepsilon = 0$ . Next, we describe the auction signature, written  $S_{ice} = (N, G, \mathcal{B}, \{\}, Y, I, \mathcal{F}_B)$ , where  $N = \{1, \dots, n\}$ ,  $G = \{1, \dots, m\}$ ,  $\mathcal{B} \subseteq \mathcal{L}_{TBBL+}$ ,  $Y = \{trade_{i,j}, payment_i : i \in N, j \in G\}$ , and  $I \subset \mathbb{Z}$ . If  $\langle q, j, \underline{v}, \bar{v} \rangle$  is a bid in  $\mathcal{B}$ , we assume  $\langle q, j, \underline{v}', \bar{v}' \rangle \in \mathcal{B}$  for any  $\underline{v}' \geq \underline{v}$  and  $\bar{v}' \leq \bar{v}$  such that  $\underline{v}' \leq \bar{v}'$ . Similarly, if  $IC_y^x(\underline{\beta}, \underline{v}, \bar{v}) \in \mathcal{B}$ , we assume  $IC_y^x(\underline{\beta}, \underline{v}', \bar{v}') \in \mathcal{B}$  for any  $\underline{v}' \geq \underline{v}$  and  $\bar{v}' \leq \bar{v}$  such that  $\underline{v}' \leq \bar{v}'$ .  $\mathcal{F}_B$  contains the basic mathematical operations as well as the following functions:  $v_i : \mathcal{B} \times I^{nm} \rightarrow I$ ,  $WD_{\lambda_{i,j}}^\varepsilon : \mathcal{B}^n \times I^{nm} \rightarrow I$ ,  $eq : \mathcal{B} \times \mathcal{B} \rightarrow [0, 1]$  and  $uncert : \mathcal{B} \rightarrow I$ . We next describe each of those functions.

Given an agent  $i$ , the value of bid  $\beta \in \mathcal{L}_{TBBL+}$  given a trade  $\lambda \in I^{nm}$  is defined as follows:

$$v_i(\beta, \lambda) = \lceil \varepsilon \cdot \underline{v}_i(\beta, \lambda) \rceil + \lfloor (1 - \varepsilon) \cdot \bar{v}_i(\beta, \lambda) \rfloor$$

Notice the rounding of the terms<sup>6</sup> ensure the result will not be smaller than the lower bound neither greater than the upper bound.

Assuming a joint bid  $\beta \in \mathcal{L}_{TBBL+}^n$  and an initial allocation  $X \in I_{\geq 0}^{nm}$ , function  $WD_{\lambda_{i,j}}^\varepsilon(\beta, X)$  is defined exactly as function  $WD_{\lambda_{i,j}}$  defined in Sect. 6.1.2, except that the winner determination is replaced by the following:

<sup>6</sup> We denote by  $\lfloor x \rfloor$  the greatest integer less than or equal to  $x \in \mathbb{R}$  and  $\lceil x \rceil$  the least integer greater than or equal to  $x \in \mathbb{R}$ .

1.  $initial \rightarrow \bigwedge_{i \in N} (payment_i = 0 \wedge \bigwedge_{j \in G} trade_{i,j} = 0)$
2.  $initial \rightarrow legal_i(\beta)$ , for each  $i \in N, \beta \in \mathcal{B}$
3.  $does_i(\beta) \wedge eq(\beta, \gamma) \wedge uncert(\gamma) < uncert(\beta) \rightarrow \bigcirc legal_i(\gamma)$ , for each  $i \in N, \beta, \gamma \in \mathcal{B}$
4.  $does_i(\beta) \wedge uncert(\beta) = 0 \rightarrow \bigcirc legal_i(\beta)$ , for each  $i \in N$  and  $\beta \in \mathcal{B}$
5.  $(\bigwedge_{i \in N} does_i(\beta_i) \wedge uncert(\beta_i) = 0) \rightarrow \bigcirc terminal$ , for each  $\beta \in \mathcal{B}^n$
6.  $does_i(\beta) \rightarrow \bigcirc payment_i = v_i(\beta, (\mathbf{trade}_j)_{j \in G})$ , for each  $i \in N$  and  $\beta \in \mathcal{B}$
7.  $does(\beta) \rightarrow \bigcirc (\bigwedge_{i \in N, j \in G} trade_{i,j} = WD_{\lambda_{i,j}}^\epsilon(\beta, X))$ , for each  $\beta \in \mathcal{B}^n$

**Fig. 6** An Iterative Combinatorial Exchange represented by  $\Sigma$

$$WD^\epsilon(\beta, X) : \underset{\lambda, sat}{\operatorname{argmax}} \sum_{i \in N} \sum_{\beta \in \text{Node}(\beta_i)} ([\epsilon \underline{b}_i(\beta) \cdot sat_i(\beta)] + [(1 - \epsilon) \bar{b}_i(\beta) \cdot sat_i(\beta)])$$

s.t. Constraints C1 – C4 hold (see Def. 18)

Given two bids  $\beta, \gamma \in \mathcal{L}_{\text{TBBL}^+}$ ,  $eq(\beta, \gamma)$  denotes whether they are equivalent on structure, in the sense of differing only on their valuation bounds, and it is defined as follows:

- If  $\beta$  is in the form  $\langle q, j, \underline{v}, \bar{v} \rangle$  and  $\gamma = \langle q, j, \underline{v}', \bar{v}' \rangle$  for some  $\underline{v}', \bar{v}' \in I$ , then  $eq(\beta, \gamma) = 1$ ;
- If  $\beta$  is in the form  $IC_y^x(\underline{\beta}, \underline{v}, \bar{v})$ ,  $\gamma = IC_y^x(\underline{\beta}', \underline{v}', \bar{v}')$  for some  $\underline{v}', \bar{v}' \in I$  and each  $\beta_k \sim \beta'_k$ , for each  $0 < k_i \leq |\beta|$ , then  $eq(\beta, \gamma) = 1$ ;
- Otherwise,  $eq(\beta, \gamma) = 0$ .

The difference among the bounds of a node represents its uncertainty. That is, the higher the bound difference, the less precise the node is about the agents' preference. The actual willingness-to-pay (or receive), is unknown except when the lower and upper bounds are the same [53]. We define function  $uncert(\langle q, j, \underline{v}, \bar{v} \rangle) = \bar{v} - \underline{v}$  and  $uncert(IC_y^x(\underline{\beta}, \underline{v}, \bar{v})) = \bar{v} - \underline{v} + \sum_{0 < k \leq |\beta|} uncert(\underline{\beta}_k)$  for capturing the uncertainty of bid in  $\mathcal{L}_{\text{TBBL}^+}$ .

Each instance of ICE is specific and is defined with respect to  $\mathcal{B}$ ,  $I$  and the constant values  $\epsilon \in [0, 1]$ ,  $n, m \in I_{>0}$  (the size of  $N$  and  $G$ , resp.), and  $X = (\mathbf{x}_j)_{j \in G}$ , where  $x_{i,j} \in I_{\geq 0}$ , for each  $i \in N$  and  $j \in G$ . The rules of a ICE are represented by ADL  $[\mathcal{F}_{\mathcal{B}}]$ -formulae as shown in Fig. 6.

In the initial state, there is no payment or trade for any agent (Rule 1) and agents can report any bid (Rule 2). After performing a bid, an agent is allowed to report any bid that has the same structure and less uncertainty than her last bid. When the bid has no uncertainty (i.e., for each node, its lower and upper bounds are the same), the agent must repeat her bid in the next turn (Rules 3 and 4). When there is no uncertainty in all bids performed in a state, the next state is terminal (Rule 5). The agents pay their reported values according to their trade in a given state (Rule 6). Finally, the agents' trades are computed in each round using the winner determination given their bids and their initial allocation (Rule 7).

### 6.3.1 Representing as a model

Next, we address the model representation. Let  $\mathcal{M}_{ice}$  be the set of ST-models  $M_{ice}$  defined for any  $\mathcal{B} \subseteq \mathcal{L}_{\text{TBBL}^+}$ ,  $I \subset \mathbb{Z}$ , and the constants  $n, m \in I_{>0}$  and  $X = (\mathbf{x}_j)_{j \in G}$ , where  $x_{i,j} \in I_{\geq 0}$ , for each  $i \in N$  and  $j \in G$ . Each  $M_{ice}$  is defined as follows:

- $W = \{ \langle (\lambda_j)_{j \in G}, p, \mathit{lastbid} \rangle : \mathit{lastbid}_i \in \mathcal{B} \ \& \ p_i, \lambda_{i,j} \in I \ \& \ i \in N \ \& \ j \in G \}$ ;
- $\bar{w} = \langle 0, \dots, 0, 0, \dots, 0, \mathit{noop}, \dots, \mathit{noop} \rangle$ ;
- $T = \{ \langle (\lambda_j)_{j \in G}, p, \mathit{lastbid} \rangle : \mathit{uncert}(\mathit{lastbid}_i) = 0 \ \& \ \mathit{lastbid}_i \in \mathcal{B} \ \& \ p_i, \lambda_{i,j} \in I \ \& \ i \in N \ \& \ j \in G \}$ ;
- $L = \{ (\bar{w}, i, \beta) : i \in N \ \& \ \beta \in \mathcal{B} \} \cup \{ \langle (\lambda_j)_{j \in G}, p, \mathit{lastbid} \rangle, i, \beta \} : \mathit{eq}(\beta, \mathit{lastbid}_i) \ \& \ \mathit{uncert}(\beta) < \mathit{uncert}(\mathit{lastbid}_i) \ \& \ \beta, \mathit{lastbid}_r \in \mathcal{B} \ \& \ p_r, \lambda_{r,j} \in I \ \& \ r, i \in N \ \& \ j \in G \}$ ;
- U is defined as follows: for all  $w = \langle (\lambda_j)_{j \in G}, p, \mathit{lastbid} \rangle \in W$  and for all  $d \in \mathcal{B}^m$ :
  - If  $w \notin T$ , then  $U(w, d) = \langle (\lambda'_j)_{j \in G}, p', d \rangle$ , where each component is updated as follows, for each  $i \in N$  and  $j \in G$ :  $\lambda'_{i,j} = \text{WD}_{\lambda_{i,j}}^\epsilon(d, X)$  and  $p'_i = v_i(d_i, (\lambda'_j)_{j \in G})$ .
  - Otherwise,  $U(w, d) = w$ .
- For each  $w \in W$ ,  $i \in N$  and  $j \in G$ , the valuation of numerical variables is as follows:  $\pi_X(w, \mathit{trade}_{i,j}) = \lambda_{i,j}$  and  $\pi_X(w, \mathit{payment}_i) = p_i$ .

Hereafter, we assume an instance of  $M_{ice} \in \mathcal{M}_{ice}$  and  $\Sigma_{ice}$  for some  $\mathcal{B} \subseteq \mathcal{L}_{\text{TBBL}^+}$ ,  $I \subset \mathbb{Z}$ ,  $n, m \in I_{>0}$  and  $x_{i,j} \in I_{\geq 0}$ , where  $i \in N$ ,  $j \in G$ .

**Example 3** Let  $M_{ice} \in \mathcal{M}_{ice}$ , where (i) there are only two agents, denoted by  $r$  and  $s$ , (ii) there is only one good type, denoted by  $a$ , and (iii)  $x_{r,a} = 1$  and  $x_{s,a} = 0$ , i.e, agent  $r$  holds 1 unit of  $a$  and agent  $s$  has none. Figure 5 illustrates a path in  $M_{ice}$ . In the initial state  $w_0$ , agent  $r$  says she wants to sell  $a$  for a price between 10€ and 20€ and agent  $s$  reports her willingness to buy it for a price between 15€ and 25€. In state  $w_1$ , the agents are informed of the provisional trade and payments. Agent  $s$  changes her bid to specify she is willing to receive exactly 15€ for selling  $a$ . By her turn, agent  $r$  specifies a value range 18€ to 20€ for buying  $a$ . In  $w_2$ , only  $s$  can change her bid, since there is no uncertainty in  $r$ 's bid. Then,  $s$  reports her willingness to pay exactly 18€. State  $w_3$  is terminal because there was no uncertainty in the bids reported on  $w_2$ . The good is traded and the agents pay (and receive) their asking prices.

### 6.3.2 Evaluating the protocol

Now we focus on the evaluation of  $\Sigma_{ice}$  and  $M_{ice}$ . First, Lemma 5 shows the soundness of  $\Sigma_{ice}$  over  $M_{ice}$ .

**Lemma 5**  $M_{ice}$  is an ST-model and it is a model of  $\Sigma$ .

Since  $M_{ice}$  is playable and terminates,  $\Sigma_{ice}$  is well-formed.

**Theorem 5**  $\Sigma_{ice}$  is a well-formed protocol over  $M_{ice}$ .

**Proof** Since  $M_{ice}$  is a model of  $\Sigma_{ice}$ , we show that for each path  $\delta$  in  $M_{ice}$ , each stage  $t \geq 0$  in  $\delta$  and each agent  $i \in N$ ,

1.  $\delta$  is a complete path;
2.  $M_{ice}, \delta, t \models \bigvee_{\beta \in \mathcal{B}} \mathit{legal}_i(\beta)$ .



First, we consider Statement 2. By the definition of  $\mathcal{L}_{\text{TBBL}^+}$ ,  $\text{uncert}(\beta) \geq 0$  for each bid  $\beta \in \mathcal{B}$ . According to the legality definition,  $\text{uncert}(\theta_i(\delta, t)) < \text{uncert}(\theta_i(\delta, t-1))$  or  $\text{uncert}(\theta_i(\delta, t)) = 0$ . That is, either the bid reported by  $i$  in  $\delta[t]$  has less uncertainty than the one she reported in  $\delta[t-1]$  or it has no uncertainty. In the case of no uncertainty,  $\theta_i(\delta, t) = \theta_i(\delta, e)$  for each  $e \geq t$ . Since the uncertainty decreases in each turn until being equal to zero, there exists a stage  $e \geq 0$  such that  $M_{ice}, \delta, e \models \text{does}_i(\beta) \wedge \text{uncert}(\beta) = 0$  for each agent  $i$ . By Rule 5, it follows that the next stage in  $\delta$  is terminal, that is,  $M_{ice}, \delta, e \models \text{Oterminal}$ .

We now verify Statement 2. If  $\delta[t] = \bar{w}$ , then  $M_{ice}, \delta, t \models \text{legal}_i(\beta)$  for each  $\beta \in \mathcal{B}$  (Rule 2). Otherwise,  $t > 0$  and  $M_{ice}, \delta, t-1 \models \text{does}_i(\beta)$  for some  $\beta \in \mathcal{B}$ . By the definition of  $\mathcal{L}_{\text{TBBL}^+}$ ,  $\text{uncert}(\beta) \geq 0$ . If  $\text{uncert}(\beta) = 0$ , then  $M_{ice}, \delta, t \models \text{legal}_i(\beta)$  (Rule 4). Finally, if  $\text{uncert}(\beta) < 0$ , by the definition of  $\mathcal{B}$ , there must exist a bid  $\gamma$  such that  $\text{eq}(\beta, \gamma)$  and  $\text{uncert}(\gamma) < \text{uncert}(\beta)$ . According to Rule 3 of  $\Sigma_{ice}$ ,  $M_{ice}, \delta, t \models \text{legal}_i(\gamma)$ .

The cumulative of payments cannot be smaller than zero. However the auction may have positive transfers.

**Proposition 9**  $M_{ice} \not\models \text{Osb}$  and  $M_{ice} \models \text{Owbb}$ .

ICE is individually rational, since agents pay their reported preferences.

**Proposition 10** Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{ice} \models \text{does}(\beta) \rightarrow \text{Oir}(\beta)$ .

The protocol is efficient since the winner determination maximizes the social welfare given the reported bids.

**Proposition 11** Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{ice} \models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$

Since the players' bids influence their payments, they can manipulate the price and the auction is not strategyproof.

**Proposition 12**  $M_{ice}$  is not strategyproof.

The automated verification of strategyproofness and termination requires meta-reasoning over the possible paths of a ST-Model. However, for a number properties (such as efficiency and individual rationality), the problem of analysing a stage as a direct mechanism boils down to model-checking ADL  $[\mathcal{F}_B]$ -formulae.

## 7 Model checking

Now we examine the complexity of the problem of deciding whether an ADL  $[\mathcal{F}_B]$  formula is true with respect to a model, a path and a stage in the path.

**Definition 19** The *model checking problem* for ADL  $[\mathcal{F}_B]$  is the following: Given an ST-Model  $M$ , a path  $\delta$  in  $M$ , a stage  $t \geq 0$  in  $\delta$  and a formula  $\varphi \in \text{ADL}[\mathcal{F}_B]$ , determine whether  $M, \delta, t \models \varphi$  or not.

In the next sections, we show that the model checking problem for  $ADL[\mathcal{F}_B]$  is decidable in polynomial-time deterministic Turing machine (PTIME) when functions in  $\mathcal{F}_B$  can be computed in polynomial time.

---

**Algorithm 1** *modelCheck*( $M, \delta, t, \varphi$ )

---

**Input:** an ST-model  $M = (W, \bar{w}, T, L, U, \pi_\phi, \pi_\psi)$ , a path  $\delta$  in  $M$ , a stage  $t \geq 1$  in  $\delta$  and a formula  $\varphi \in \mathcal{L}_{ADL[\mathcal{F}_B]}$ .

**Output:** **true** if  $M, \delta, t \models \varphi$ , and **false** otherwise

```

1:  $S \leftarrow Subformula(\varphi)$  ordered by ascending length
2: Let  $isTrue[1, \dots, |\varphi|]$  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:        $isTrue[i] \leftarrow isTrue[getIndex(S, \phi')] \wedge isTrue[getIndex(S, \phi'')]$ 
8:     case  $\phi$  is of the form  $\neg\phi'$ 
9:        $isTrue[i] \leftarrow \neg isTrue[getIndex(S, \phi')]$ 
10:    case  $\phi$  is of the form  $\bigcirc\phi'$ 
11:       $isTrue[i] \leftarrow isTrue[i] \wedge modelCheck(M, \delta, j + 1, \phi')$ 
12:    case  $\phi$  is atomic
13:       $isTrue[i] \leftarrow M, \delta, t \models \phi$ 
14: return  $isTrue[|\varphi|]$ 

```

---

### 7.1 Upper bound

Let  $\varphi \in \mathcal{L}$  be a formula and  $M$  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  $\neg\phi$ , or  $\bigcirc\psi$  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  $\phi'$ . We denote  $Subformula(\varphi)$  as the set of all subformulae of  $\varphi$ .

**Theorem 6** *Assuming that functions in  $\mathcal{F}_B$  can be computed in polynomial time, the model checking problem for  $ADL[\mathcal{F}_B]$  is in PTIME.*

**Proof** Assume the functions in  $\mathcal{F}_B$  can be computed in polynomial time. Algorithm *modelCheck* works in the following way: first it gets all subformulae 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  $\delta[j]$  under  $M$ . Since functions in  $\mathcal{F}_B$  can be computed in polynomial time, if  $\phi_i$  does not have any subformula, its truth value is obtained directly from the model. Since  $S$  is ordered by formulae 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  $\bigcirc\phi'$  then its label is recursively defined according to  $\phi'$  truth value in the stage  $t + 1$ . 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 a polynomial-time deterministic Turing machine with PTIME.

## 7.2 Lower bound

Now let us characterize a lower bound of the complexity of model-checking  $ADL[\mathcal{F}_B]$ . To do so, we reduce the model checking problem for GDL, which is known to be PTIME (consequence of Lemma 2 by Jiang et al. [29]).

**Theorem 7** *The model checking problem for  $ADL[\mathcal{F}_B]$  is PTIME.*

**Proof** First, we recall GDL definitions. By convenience, we refer to the GDL formalization presented in [27]. Let  $\mathcal{S}_{GDL} = (N, \mathcal{B}, \Phi)$  be a game signature, where  $\Phi$  is a propositional set,  $N$  is a set of agents, and  $\mathcal{B}$  is the action set. Let  $M_{GDL} = (W, \bar{w}, T, L, U, g, \pi_\Phi)$  be a GDL model over  $\mathcal{S}_{GDL}$ , where  $W, \bar{w}, T, L$  and  $\pi_\Phi$  are defined in the same way as in a  $ADL[\mathcal{F}_B]$  model (see Definition 2),  $U : W \times \mathcal{B}^{N^I} \rightarrow W \setminus \{\bar{w}\}$  is the update function and  $g : N \rightarrow 2^W$  is the goal function.

A path  $\delta$  in  $M_{GDL}$  is an infinite sequence of states and joint actions, defined in the same way as Definition 3. Let  $\delta$  be a path in  $M_{GDL}$ , a formula  $\varphi$  in  $\mathcal{L}_{GDL}$  is defined by the following BNF:

$$\varphi ::= p \mid initial \mid terminal \mid legal_r(a) \mid wins(r) \mid does_r(a) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \bigcirc\varphi$$

where  $p \in \Phi, r \in N$  and  $a \in \mathcal{B}$ .

For any stage  $t \geq 0$  and formula  $\varphi \in \mathcal{L}_{GDL}$ , the semantics of  $\mathcal{L}_{GDL}$  are similar to Definition 5, except by the following case:  $M_{GDL}, \delta, t \models wins(r)$  iff  $\delta[t] \in g(r)$ .

Given a stage  $t > 0$  and a formula  $\varphi \in \mathcal{L}_{GDL}$ , we show how to construct an  $ADL[\mathcal{F}_B]$  model  $M_{ADL[\mathcal{F}_B]}$ , such that  $M, \delta, t \models \varphi$  iff  $M_{GDL}, \delta, t \models \varphi$ .

Let  $\mathcal{S}_{ADL[\mathcal{F}_B]} = (N, \emptyset, \mathcal{A}, \Phi', \emptyset, [0, 0], \emptyset)$  be the auction signature, where  $\Phi' = \Phi \cup \{wins(r) : r \in N\}$ . We define the  $ADL[\mathcal{F}_B]$  model  $M_{ADL[\mathcal{F}_B]} = (W, \bar{w}, T, L, U, \pi'_\Phi, \emptyset)$ , where  $W, \bar{w}, T$  and  $U$  are the same as in  $M_{GDL}$ . The valuation function for state propositions is defined as follows:  $\pi'_\Phi(w) = \pi_\Phi(w) \cup \{wins(r) : w \in g(r) \ \& \ r \in N\}$ , for each  $w \in W$ .

Since  $U(w, \mathbf{d}) \in W$ , for each  $w \in W$  and  $\mathbf{d} \in \mathcal{B}^{N^I}$ , we have that  $U$  is an update function in accordance with Definition 2. Furthermore, if  $\delta$  is a path in  $M_{GDL}$  then it is a path in  $M_{ADL[\mathcal{F}_B]}$ . Since  $\{wins(r) : r \in N\} \subseteq \Phi'$ , we have that  $\mathcal{L}_{GDL} \subseteq \mathcal{L}_{ADL[\mathcal{F}_B]}$ .

Given  $\varphi \in \mathcal{L}_{GDL}$ , if  $\varphi$  is not in the form  $wins(r)$ , then it is straightforward that  $M_{GDL}, \delta, t \models \varphi$  iff  $M_{ADL[\mathcal{F}_B]}, \delta, t \models \varphi$ . Now we consider the case were  $\varphi \in \mathcal{L}_{GDL}$  is in the form  $wins(r)$ , for some  $r \in N$ . Assume  $M_{GDL}, \delta, t \models wins(r)$  iff  $\delta[t] \in g(r)$  iff  $wins(r) \in \Phi'(\delta[t])$  iff  $M_{ADL[\mathcal{F}_B]}, \delta, t \models wins(r)$ .

Theorem 7 shows that Algorithm *modelCheck* is optimal when the functions in  $\mathcal{F}_B$  can be computed in polynomial time. As for GDL, the ST-model may have exponential size.

Since Algorithm *modelCheck* calls the functions in  $\mathcal{F}_B$  in a polynomial number of times (according to on the formula length), the complexity of computing functions in  $\mathcal{F}_B$  will affect its complexity. For instance, the winner determination problem for combinatorial auctions is NP-hard, which is then the complexity of computing the winner determination functions for TBBL [41]. In that case, the model checking problem for  $ADL[\mathcal{F}_B]$  is in  $\Delta_2^P$ , since *modelCheck* consults a NP-oracle a polynomial number of times.

## 8 Conclusion

In this paper, we have presented ADL  $[\mathcal{F}_B]$ , a unified framework for representing auction protocols. Our work is at the frontier of auction theory and knowledge representation. We illustrated the usefulness of our approach by showing how to represent a number of representative auctions, which include features from single and multi-stage protocols, multiple items, multiple copies of those items and exchange protocols (which generalize double-sided auctions).

ADL  $[\mathcal{F}_B]$  provides tools for automated verification of properties from mechanism design. We showed how stages in an ADL  $[\mathcal{F}_B]$  ST-model may represent direct mechanisms and be evaluated as such. The majority of properties that we considered (noting that at the evaluating a stage in an ST-model essentially comes down to model-checking ADL  $[\mathcal{F}_B]$ -formulae) can be checked in PTIME when the functions in  $\mathcal{F}_B$  can be computed in polynomial time. Thus, ADL  $[\mathcal{F}_B]$  enables reasoning about important aspects for designing and playing auctions, while having a reasonable complexity cost.

The main limitation of ADL  $[\mathcal{F}_B]$  is due to having a semantics based on fixed paths, that is, non-alternating executions of an auction. This means it is not possible to encode through ADL  $[\mathcal{F}_B]$ -formulae conditions that compare the effect of different strategic behaviour. An example of such condition is strategyproofness, where one should contrast the agents' utility after truthfully reporting her preference and after lying. Here, we demonstrated the use of meta-reasoning over the state-transition model for comparing alternative executions of an auction. Furthermore, evaluating indirect mechanisms requires capturing the terminal outcomes (that is, the final trades and payments) in strategic equilibrium. Complex solution concepts, such as Nash and dominant strategy equilibrium, cannot be encoded through ADL  $[\mathcal{F}_B]$ -formulae. Logics focused on strategic reasoning are more suitable for considering this problem (e.g., ATL with Strategy Contexts [10] and Strategy Logic (SL) [14]). Finding a balance between expressivity and complexity is an open question, as such expressive languages face decidability issues and high complexity for model-checking (e.g., the satisfiability problem of SL is undecidable and its complexity for model-checking is Non Elementary).

ADL  $[\mathcal{F}_B]$  definitely puts the emphasis on the auctioneer and mechanism designer. Our next direction is to design a ADL  $[\mathcal{F}_B]$ -based *General Auction Player* (GAP) that can interpret and reason about the rules of an auction-based market. The key difference, when the player perspective is considered, is the epistemic and strategic aspects: players have to reason about other players' behavior and may have imperfect information. Additionally, search optimization techniques used for GGP (see, for instance [20, 68]) may be adapted for considering utility optimization in auctions. Another future direction is to explore the synthesis of state-transition models based on a ADL  $[\mathcal{F}_B]$ -description of an auction. This problem entails considering the satisfiability problem for the language.

Last but not least, we intend to develop the axiomatic system for ADL  $[\mathcal{F}_B]$  and prove its soundness and completeness with respect to the semantics based on the state transition model. It would require a combination of techniques used for Epistemic GDL [28], first-order logic with dependent types [58] and finite algebras [11].

## Appendix A: Technical Proof

**Proposition 1** *Let  $M$  be an ST-model, for each agent  $i \in N$  and each action  $\beta \in \mathcal{B}$ ,*

1.  $M \models \text{does}_i(\beta) \rightarrow \neg \text{does}_i(\beta')$ , for any  $\beta' \in \mathcal{B}$  such that  $\beta' \neq \beta$
2.  $M \models \bigvee_{\beta' \in \mathcal{B}} \text{does}_i(\beta')$
3.  $M \models \text{does}_i(\beta) \rightarrow \text{legal}_i(\beta)$
4.  $M \models \neg \bigcirc \text{initial}$
5.  $M \models \text{terminal} \wedge \varphi \rightarrow \bigcirc \varphi$ , for any  $\varphi \in \mathcal{L}_{\text{ADL}[\mathcal{F}_B]}$
6.  $M \models \text{initial} \rightarrow \neg \text{terminal}$

**Proof** Let  $M$  be an ST-model,  $\delta$  be a path,  $t \geq 0$  be an stage in  $\delta$ ,  $i \in N$  be an agent and  $\beta \in \mathcal{B}$  be an action. For Statement 1, assume  $M, \delta, t \models \text{does}_i(\beta)$  iff  $\theta_i(\delta, t) = \beta$ . Then for any  $\beta' \in \mathcal{B}$  such that  $\beta' \neq \beta$ ,  $\theta_i(\delta, t) \neq \beta'$  and  $M, \delta, t \models \neg \text{does}_i(\beta')$ .

Statement 2 follows from the definition of  $\delta$ , since  $\theta_i(\delta, t) \in L(\delta[t], i)$  and  $L(\delta[t], i) \subseteq \mathcal{B}$ , we have  $M, \delta, t \models \bigvee_{\beta' \in \mathcal{B}} \text{does}_i(\beta')$ .

Let us verify Statement 3. Assume  $M, \delta, t \models \text{does}_i(\beta)$ , then  $\theta_i(\delta, t) = \beta$  and by the definition of  $\delta$ ,  $\beta \in L(\delta[t], i)$  and  $M, \delta, t \models \text{legal}_i(\beta)$ .

We consider Statement 4. By the path definition, for all  $t > 0$ ,  $\delta[t] \neq \bar{w}$ . Thus,  $\delta[t + 1] \neq \bar{w}$  and  $M, \delta, t + 1 \models \neg \text{initial}$ . It follows that  $M, \delta, t \models \neg \bigcirc \text{initial}$ .

We now verify Statement 5. Assume  $M \models \text{terminal} \wedge \varphi$ , for some  $\varphi \in \mathcal{L}_{\text{ADL}[\mathcal{F}_B]}$ . Then  $\delta[t] \in T$  and  $\delta[t + 1] = \delta[t]$ . Thus,  $M, \delta, t + 1 \models \varphi$  and  $M, \delta, t \models \bigcirc \varphi$ .

Finally, we consider Statement 6. Assume for the sake of contradiction that  $M, \delta, t \models \text{initial} \wedge \text{terminal}$ . Then,  $\delta[t] = \bar{w}$  and  $\delta[t] \in T$ . By the path definition, it should be the case that  $t = 0$ . Due to the loop on terminal states, it follows that  $w_0 = w_1$ , which is a contradiction with the path requirement  $w_{t'} \neq w_0$ , for any  $t' \geq 1$ .

**Lemma 1**  $M_{sa}$  is an ST-model and it is a model of  $\Sigma_{sa}$ .

**Proof (Sketch)** It is routine to check that  $M_{sa}$  is actually an ST-model. Given a path  $\delta$ , any stage  $t$  of  $\delta$  in  $M_{sa}$ , we need to show that  $M_{sa}, \delta, t \models \varphi$ , for each  $\varphi \in \Sigma_{sa}$ . Let us verify Rule 1. Assume  $M_{sa}, \delta, t \models \text{initial}$  iff  $\delta[t] = \bar{w}$ . By the definition of  $\bar{w}$ ,  $\pi_\Phi$  and  $\pi_Y$ , we have  $\pi_Y(\bar{w}, \text{price}) = \text{start}$ ,  $\pi_Y(\bar{w}, \text{price}_j) = \text{start}$ ,  $\text{bid}_{i,j} \notin \pi_\Phi(\bar{w})$  and  $\text{trade}_{i,j} = 0$ , for all  $i \in N$  and  $j \in G$ . Thus,  $M_{sa}, \delta, t \models \text{initial}$  iff  $M_{sa}, \delta, t \models \text{price} = \text{start} \wedge \bigwedge_{j \in G} (\text{price}_j = \text{start} \wedge \bigwedge_{i \in N} (\neg \text{bid}_{i,j} \wedge \text{trade}_{i,j} = 0))$ .

Now we verify Rule 2. Let  $j \in G$  be a good type. Assume  $M_{sa}, \delta, t \models \text{sold}_j$  iff  $\text{sold}_j \in \pi_\Phi(\delta[t])$  iff  $\pi_Y(\delta[t], \text{trade}_{i,j}) = 1$  for some  $j \in G$  iff  $M_{sa}, \delta, t \models \bigvee_{i \in N} \text{trade}_{i,j} = 1$ .

Next, we consider Rule 3. Assume  $M_{sa}, \delta, t \models \text{terminal}$  iff  $\delta[t] \neq \bar{w}$  and for all  $j \in G$ , either  $M_{sa}, \delta, t \models \text{trade}_{r,j} = 1$  for some  $r \in N$  or  $M_{sa}, \delta, t \models \neg \text{bid}_{i,j}$  for all  $i \in N$ . By Rule 2,  $M_{sa}, \delta, t \models \text{terminal}$  iff  $M_{sa}, \delta, t \models \neg \text{initial} \wedge \bigwedge_{j \in G} (\text{sold}_j \vee \bigwedge_{i \in N} \neg \text{bid}_{i,j})$ .

Now we verify Rule 9. Let  $i \in N$  and  $j \in G$ . Assume  $M_{sa}, \delta, t \models (\text{does}_i(p_1, \dots, p_m) \wedge p_j \neq 0) \vee (\text{bid}_{i,j} \wedge \text{terminal})$ , for some  $p_1, \dots, p_m \in I_{\geq 0}$ . We next prove for the two cases. First, assume  $M_{sa}, \delta, t \models \text{bid}_{i,j} \wedge \text{terminal}$ . Then  $\text{bid}_{i,j} \in \pi_\Phi(\delta[t])$  and  $\delta[t] \in T$ . By the update function,  $\delta[t + 1] = \delta[t]$  and  $M_{sa}, \delta, t + 1 \models \text{bid}_{i,j}$ , i.e.,  $M_{sa}, \delta, t \models \bigcirc \text{bid}_{i,j}$ . In the second case, assume  $\text{does}_i(p_1, \dots, p_m) \wedge p_j \neq 0$ . By the update function,  $\text{bid}_{i,j} \in \pi_\Phi(\delta[t + 1])$  and thus  $M_{sa}, \delta, t \models \bigcirc \text{bid}_{i,j}$ .

The remaining rules are verified in a similar way.

**Proposition 2** For each  $j \in G$  and each  $i, r \in N$  such that  $i \neq r$ ,

1.  $M_{sa} \models trade_{i,j} = 0 \vee trade_{r,j} = 0$
2.  $M_{sa} \models sold_j \rightarrow \bigcirc sold_j$
3.  $M_{sa} \models \neg sold_j \rightarrow price = price_j$

**Proof** Given a path  $\delta$  in  $M_{sa}$ , any stage  $t$  of  $\delta$  and a good type  $j \in G$ , let  $i, r \in N$ , such that  $i \neq r$ . Let us consider Statement 1. If  $\delta[t] = \bar{w}$ , then  $M_{sa}, \delta, t \models trade_{i,j} = 0 \wedge trade_{r,j} = 0$  (see Rule 1). Otherwise, by the path definition,  $\delta[j] = U(\delta[t - 1], \theta(\delta, t - 1))$ . Let us suppose for the sake of contradiction that  $M_{sa}, \delta, t \not\models trade_{i,j} = 0 \vee trade_{r,j} = 0$ . Since  $W$  construction defines  $trade_{i,j}, trade_{r,j} \in \{0, 1\}$ , we have  $M_{sa}, \delta, t \models trade_{i,j} = 1 \wedge trade_{r,j} = 1$ . Thus,  $M_{sa}, \delta, t - 1 \models \bigcirc trade_{i,j}$ . By Rule 4,  $M_{sa}, \delta, t - 1 \models \bigcirc (bid_{i,j} \wedge \bigwedge_{s \in N \setminus \{i\}} \neg bid_{s,j})$ . Thereby,  $M_{sa}, \delta, t - 1 \not\models \bigcirc (bid_{r,j} \wedge \bigwedge_{s \in N \setminus \{r\}} \neg bid_{s,j})$  and  $M_{sa}, \delta, t - 1 \not\models \bigcirc trade_{r,j}$ . Thus,  $M_{sa}, \delta, t \not\models trade_{r,j} = 1$ , which is a contradiction.

For Statement 2. Assume  $M_{sa}, \delta, t \models sold_j$ . Then,  $M_{sa}, \delta, t \models trade_{i,j} = 1$  for some agent  $i \in N$ . From Rule 6,  $\theta_i(\delta, t) = (p_1, \dots, p_m)$  with  $p_j = \pi_Y(\delta[t], price_j)$  and  $\theta_r(\delta, t) = (p'_1, \dots, p'_m)$  with  $p'_j = 0$ , for all agent  $r \neq i$ . By Rule 9, we have  $M_{sa}, \delta, t \models bid_{i,j} \wedge \bigwedge_{r \in N \setminus \{i\}} \neg bid_{r,j}$ . Thus, it follows from Rules 2 and 4 that  $M_{sa}, \delta, t \models \bigcirc trade_{i,j} = 1$  and  $M_{sa}, \delta, t \models \bigcirc sold_j$ .

For Statement 3, notice all prices in the initial state have the same value (Rule 1) and the current price and the price for unsold items are increased by the same amount in each turn (Rule 7 and 8). Assume  $M_{sa}, \delta, t \models \neg sold_j$ . Since  $M_{sa} \models sold_j \rightarrow \bigcirc sold_j$ , there is no stage  $t' < t$  such that  $M_{sa}, \delta, t' \models sold_j$  and thus  $M_{sa}, \delta, t \models price = price_j$ .

**Proposition 3**  $M_{sa} \models \bigcirc wbb$  and  $M_{sa} \not\models \bigcirc sbb$

**Proof** Suppose a path  $\delta$  in  $M_{sa}$  and a stage  $t$  in  $\delta$ . Note that each trade can be either 0 or 1 and the price for a good is at least 0, i.e.,  $\pi_Y(\delta[t], trade_{i,j}) \in \{0, 1\}$  and  $\pi_Y(\delta[t], price_j) \in I_{\geq 0}$ . It follows from Rule 10 that  $M_{sa}, \delta, t \models payment_i \geq 0$  for each agent  $i$  and  $M_{sa}, \delta, t \models \text{sum}_{i \in N}(payment_i) \geq 0$ . Thus,  $M_{sa} \models \bigcirc wbb$ .

We will prove  $M_{sa}$  is not strongly budget-balanced with a counter-example. Given an agent  $i$ , let  $\delta$  be a path in  $M_{sa}$  such that  $\theta_i(\delta, 0) = (\text{start}, 0, \dots, 0)$  and  $\theta_s(\delta, 0) = (0, \dots, 0)$  for each agent  $s \neq i$ . Since  $(\text{start}, 0, \dots, 0) \in L(\delta[0], i)$  and  $(0, \dots, 0) \in L(w, s)$ , there exists such path in  $M_{sa}$ . Since  $\delta[0] \notin T$ ,  $\pi_Y(\delta[0], price) = \pi_Y(\delta[0], price_j) = \text{start}$  and  $sold_j \notin \pi_\Phi(\delta[0])$  for each  $j$ , it follows from Rules 7 and 8, that all prices are increased by the constant  $\text{inc} > 0$ , that is  $M_{sa}, \delta, 1 \models price = \text{sum}(\text{start}, \text{inc}) \wedge \bigwedge_{j \in G} price_j = \text{sum}(\text{start}, \text{inc})$ . By Rule 9, we have that agent  $i$  is only bidding for the good 1, that is,  $M_{sa}, \delta, 1 \models bid_{i,1} \wedge \bigwedge_{j \in G \setminus \{1\}} \neg bid_{i,j}$ . All other agents are not bidding for any good, i.e.,  $M_{sa}, \delta, 1 \models \bigwedge_{j \in G} \neg bid_{s,j}$ , for each  $s \neq i$ . From Rules 4 and 10, we have  $M_{sa}, \delta, 1 \models trade_{i,1} = 1 \wedge payment_i = price_1 \wedge \bigwedge_{s \in N \setminus \{i\}} p_s = 0$ . Since  $\pi_Y(\delta[1], price_1) > 0$ , we have  $M_{sa}, \delta, 1 \not\models \text{sum}_{s \in N}(payment_s) = 0$  and  $M_{sa} \not\models \bigcirc sbb$ .

**Proposition 4** Given a joint action  $\beta \in \mathcal{B}^n$ ,

1.  $M_{sa} \not\models \text{does}(\beta) \rightarrow \bigcirc \text{ef}(\beta)$
2.  $M_{sa} \models \text{does}(\beta) \rightarrow \bigcirc (\text{terminal} \rightarrow \text{ef}(\beta))$

**Proof** Given a path  $\delta$  in  $M_{sa}$ , we first consider Statement 1. We prove  $M_{sa}$  is not efficient with a counter-example. Our purpose is to show that there exists a joint action  $\beta$  such that  $M_{sa}, \delta, 0 \models \text{does}(\beta) \wedge \neg \bigcirc \text{eff}(\beta)$ . That is,  $M_{sa}, \delta, 0 \models \text{does}(\beta) \wedge \neg \bigcirc \text{sum}_{i \in N} (v_i(\beta_i, (\text{trade}_j)_{j \in G}))$

$= \max_{\lambda \in \Lambda} (\text{sum}_{i \in N} (v_i(\beta_i, \lambda)))$   
 As we consider the initial state of  $\delta$ ,  $M_{sa}, \delta, 0 \models \text{price} = \text{start} \wedge \bigwedge_{j \in G} \text{price}_j = \text{start} \wedge \bigwedge_{i \in N} (\neg \text{bid}_{i,j} \wedge \text{trade}_{i,j} = 0)$ . Thus, no good was sold, that is,  $M_{sa}, \delta, 0 \models \bigwedge_{j \in G} \neg \text{sold}_j$ . Let  $i$  and  $r$  be two distinct agents in  $N$  with  $i \neq r$ . From the definition of  $L$ ,  $L(\delta[0], i) = L(\delta[0], r)$  and  $(p_1, \dots, p_m) \in L(\delta[0], i)$  if  $p_j = 0$  or  $p_j = \text{sum}(\text{start}, \text{inc})$  for each good  $j$ .

Let us assume  $M_{sa}, \delta, 0 \models \text{does}(\beta)$  for  $\beta \in \mathcal{B}^n$  such that  $\beta_i = \beta_r = (\text{start} + \text{inc}, 0, \dots, 0)$  and  $\beta_s = (0, \dots, 0)$  for each  $s \in N \setminus \{i, r\}$ . Since  $\beta_i \in L(\delta[0], i)$ ,  $\beta_r \in L(\delta[0], r)$  and  $\beta_s \in L(\delta[0], s)$ , there exists such path in  $M_{sa}$ . Thus,  $M_{sa}, \delta, 1 \models \text{bid}_{i,1} \wedge \text{bid}_{r,1}$ . Notice  $\text{bid}_{i',j}$  does not hold in  $\delta[1]$  for any other pair  $(i', j) \neq (i, 1)$  and  $(i', j) \neq (s, 1)$ . From Rules 4 and 5, we have  $M_{sa}, \delta, 1 \models \bigwedge_{i' \in N, j \in G} \text{trade}_{i',j} = 0$ .

Recall the valuation of each agent  $i'$  is  $v'_i((p_1, \dots, p_m), \lambda) = \sum_{j \in G} \lambda_{i',j} \cdot p_j$  for a trade  $\lambda \in I^{nm}$  and a bid  $(p_1, \dots, p_m) \in \mathcal{B}$ . Since each trade has the value 0 in  $\delta[1]$ , it follows that  $M_{sa}, \delta, 1 \models \text{sum}_{i \in N} (v_i(\beta_i, (\text{trade}_j)_{j \in G})) = 0$ .

However, let  $\lambda \in I^{nm}$  be a trade such that  $\lambda_{i,1} = 1$  and  $\lambda_{i',j} = 0$  for all other pair  $(i', j) \neq (i, 1)$ . It is easy to see that  $\lambda \in \Lambda$ . The value of this trade for  $i$  is  $v_i(\beta_i, \lambda) = \text{start} + \text{inc}$  and  $v'_i(\beta'_i, \lambda) = 0$  for each  $i' \neq i$ . Thus, we have  $\text{sum}_{i \in N} (v_i(\beta_i, \lambda)) = \text{start} + \text{inc}$ ,  $M_{sa}, \delta, 1 \models \neg \text{sum}_{i \in N} (v_i(\beta_i, (\text{trade}_j)_{j \in G})) = \max_{\lambda \in \Lambda} (\text{sum}_{i \in N} (v_i(\beta_i, \lambda)))$  and  $M_{sa}, \delta, 0 \models \neg \bigcirc \text{eff}(\beta)$ .

For Statement 2, let  $\beta \in \mathcal{B}^n$  be a joint action and  $t \geq 0$  be a stage of  $\delta$ . From the definition of  $M_{sa}$  we have that each trade can be only 0 or 1. That is, if  $\lambda \in \Lambda$  then  $\lambda_{i,j} \in \{0, 1\}$  for each agent  $i$  and good  $j$ . Assume  $M_{sa} \models \text{does}(\beta)$  and  $M_{sa}, \delta, t \models \bigcirc \text{terminal}$ . We intend to show that  $M_{sa} \models \bigcirc \text{eff}(\beta)$ , i.e.,  $M_{sa}, \delta, t \models \bigcirc \text{sum}_{i \in N} (v_i(\beta_i, (\text{trade}_j)_{j \in G})) = \max_{\lambda \in \Lambda} (\text{sum}_{i \in N} (v_i(\beta_i, \lambda)))$ .

We focus on the case where  $\delta[t] \notin T$ . Let  $i$  be an agent in  $N$  and let  $(p_1, \dots, p_m)$  denote  $i$ 's action in  $\beta$ . Recall function  $v_i((p_1, \dots, p_m), \lambda) = \sum_{j \in G} p_j \cdot \lambda_{i,j}$ , for a trade  $\lambda \in I^{nm}$ . Since the value for each good in  $v_i$  depends only on its reported value in  $i$ 's bid and its trade for  $i$ , we show that the part of  $v_i((p_1, \dots, p_m), \lambda)$  corresponding to each good  $j$  (i.e.,  $p_j \cdot \lambda_{i,j}$ ) is maximized when  $\lambda_{i,j} = \pi_{\forall}(\delta[t], \text{trade}_{i,j})$ .

From the definition of  $\mathcal{B}$ , we have that the reported value for  $j$  in  $i$ 's bid is at least zero, i.e.,  $p_j \geq 0$ . We check for two cases:

1. If  $p_j = 0$ , then  $\lambda_{i,j} \cdot p_j = 0$ , for any trade  $\lambda \in \Lambda$ . Thus,  $M_{sa}, \delta, t + 1 \models \text{times}(p_j, \text{trade}_{i,j}) = \max(\text{times}(p_j, 0), \text{times}(p_j, 1))$ ;
2. If  $p_j > 0$ , then  $M_{sa}, \delta, t + 1 \models \text{bid}_{i,j}$ . Since  $\delta[t + 1] \in T$ , it should be the case that  $M_{sa}, \delta, t + 1 \models \text{sold}_j$ . From Rule 2, we know the trade for good  $j$  is one for some agent, i.e.,  $M_{sa}, \delta, t + 1 \models \bigvee_{r \in N} \text{trade}_{r,j} = 1$ . Because  $M_{sa}, \delta, t + 1 \models \text{bid}_{i,j}$ , it should be the case that the agent who have a trade for  $j$  is  $i$  (see Rules 4 and 5), that is,  $M_{sa}, \delta, t + 1 \models \text{trade}_{i,j} = 1$ . Hence,  $M_{sa}, \delta, t + 1 \models \text{times}(p_j, 1) = \max(\text{times}(p_j, 0), \text{times}(p_j, 1))$ .

It follows that

$$M_{sa}, \delta, t + 1 \models \text{sum}_{i \in N} (v_i(\beta_i, (\text{trade}_j)_{j \in G})) = \max_{\lambda \in \Lambda} (\text{sum}_{i \in N} (v_i(\beta_i, \lambda)))$$

or simply,  $M_{sa}, \delta, t \models \bigcirc \text{eff}(\beta)$ .

The case where  $\delta[t] \in T$ , follows from the loop on the path definition.

**Proposition 5**  $M_{sa}$  is strategyproof.

**Proof** Given a path  $\delta$  in  $M_{sa}$  and a stage  $t \geq 0$  in  $\delta$ , such that  $L(\delta[t], i) \approx_i V_i$ . Let  $\vartheta \in \prod_{i \in N} V_i$  be a preference profile and  $\delta_\vartheta$  denote a path such that  $\delta[0, t] = \delta_\vartheta[0, t]$  and  $\theta(\delta_\vartheta, t) = \beta_\vartheta$ . We let  $\vartheta'_i \in V_i$  denote a preference of agent  $i$ ,  $\vartheta' = (\vartheta'_i, \vartheta_{-i})$  and  $\delta_{\vartheta'}$  be a path such that  $\delta[0, t] = \delta_{\vartheta'}[0, t]$ ,  $\theta_i(\delta_{\vartheta'}, t) = \beta_{\vartheta'_i}$  and  $\theta_r(\delta_{\vartheta'}, t) = \beta_{\vartheta_r}$  for each agent  $r \neq i$ .

In the stage  $t$  of  $\delta_\vartheta$ , the agents report their (truthful) preferences  $\vartheta$ , i.e.,  $M_{sa}, \delta_\vartheta, t \models \text{does}(\beta_\vartheta)$ . On the other hand, in the stage  $t$  of  $\delta_{\vartheta'}$ , the agent  $i$  reports her (untruthful) preference  $\vartheta'_i$  and each agent  $r \neq i$  reports  $\vartheta_r$ , i.e.,  $\bigwedge_{r \in N \setminus \{i\}} \text{does}_r(\beta_{\vartheta_r})$ .  $M_{sa}, \delta_{\vartheta'}, t \models \text{does}_i(\beta_{\vartheta'_i})$ .

For some  $x \in I$ , we have to show  $M_{sa}, \delta_\vartheta, t \not\models \text{Osub}(v_i(\beta_{\vartheta_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) = x$  and  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(v_i(\beta_{\vartheta_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) \leq x$ .

That is, agent  $i$ 's utility in  $\delta_{\vartheta'}$  is not better than in  $\delta_\vartheta$ . Remind the legal actions in  $\delta[t]$  represent her preference space, i.e.,  $L(\delta[t], i) \approx_i V_i$ . Notice the value of a bid  $(p_1, \dots, p_m)$  given a trade  $\lambda$  is  $v_i((p_1, \dots, p_m), \lambda) = \sum_{j \in G} \lambda_{i,j} \cdot p_j$ . Similarly, the payment in  $\delta[t]$  is  $\pi_Y(\delta[t], \text{payment}_i) = \sum_{j \in G} \pi_Y(\delta[t], \text{trade}_{i,j}) \cdot \pi_Y(\delta[t], \text{price}_j)$ . Since there is no dependence among goods in  $v_i$  and in  $i$ 's payment, we consider the part of  $i$ 's utility corresponding to  $j$  in  $\delta_\vartheta$  and  $\delta_{\vartheta'}$ . Thus, we need to show that

- (i)  $M_{sa}, \delta_\vartheta, t \not\models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = x_j$
- (ii)  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) \leq x_j$

for some  $x_j \in I$  and each good  $j \in G$ .

We denote  $\beta_{\vartheta_i} = (p_1, \dots, p_m)$  and  $\beta_{\vartheta'_i} = (p'_1, \dots, p'_m)$ . According to the legality definition, the values of  $p_j$  and  $p'_j$  can be either the price of  $j$  in  $\delta[t]$ , zero, or the price in  $\delta[t]$  incremented by  $\text{inc} > 0$ . Let us consider each case:

- Assume  $p_j = \pi_Y(\delta[t], \text{price}_j)$  iff  $M_{sa}, \delta, t \models \text{trade}_{i,j} = 1$  (w.r.t the definition of  $L$ ). That is,  $p_j = \pi_Y(\delta[t], \text{price}_j)$  when good  $j$  was already bought by agent  $i$ . Thus, it is not legal for  $i$  to bid any other value for  $j$ , i.e.,  $(p'_1, \dots, p'_j) \in L(\delta[t], i)$  iff  $p_j = p'_j$ . Thereby, it is easy to see that  $M_{sa}, \delta_\vartheta, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = p_j$  and also  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = p_j$ .
- Assume agent  $i$  declines to raise her bid for good  $j$ , that is,  $p_j = 0$ . By Rules 5 and 9, we have  $M_{sa}, \delta_\vartheta, t \models \text{Otrade}_{i,j} = 0$  and  $M_{sa}, \delta_\vartheta, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = 0$ . By the legality definition, it should be the case that  $M_{sa}, \delta, t \models \neg \text{trade}_{i,j} = 0$  and  $p'_j$  is either 0 or  $\pi_\Phi(\delta[t], \text{price}_j) + \text{inc}$ . If  $p'_j = 0$ , then  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = 0$ . Otherwise, the value of  $\text{trade}_{i,j}$  in  $\delta_{\vartheta'}[t + 1]$  will be either 0 or 1, depending on the joint bid  $\beta_{\vartheta'}$ .
  - If  $M_{sa}, \delta_{\vartheta'}, t \models \text{Otrade}_{i,j} = 1$ , then  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = \text{sub}(0, \text{price}_j)$ . Since  $\pi_Y(\delta_{\vartheta'}[t + 1], \text{price}_j) \geq 0$ , we have  $\text{sub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) \leq 0$ .
  - Otherwise, the trade for good  $j$  is zero (i.e.,  $M_{sa}, \delta_{\vartheta'}, t \models \text{Otrade}_{i,j} = 0$ ) and  $M_{sa}, \delta_{\vartheta'}, t \models \text{Osub}(\text{times}(\text{trade}_{i,j}, p_j), \text{times}(\text{trade}_{i,j}, \text{price}_j)) = 0$ .
- The proof for the case  $p_j = \pi_Y(\delta[t], \text{price}_j) + \text{inc}$  is similar to the previous case.



It follows that  $M_{sa}, \delta_g, t \models \text{Osub}(v_i(\beta_{g_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) = x$  and  $M_{sa}, \delta_{g'}, t \models \text{Osub}(v_i(\beta_{g_i}, (\text{trade}_j)_{j \in G}), \text{payment}_i) \leq x$ , where  $x \in \mathbb{I}$ .

**Lemma 2** For each agent  $i \in \mathbb{N}$ , each bid-tree  $\beta \in \mathcal{L}_{\text{TBBL}}$  and each  $\lambda \in \mathbb{I}^m$ ,  $v_i(\text{noop}, \lambda) = 0$ .

**Proof** Remind  $\text{noop}$  denotes a leaf bid  $\langle 0, j, 0 \rangle$ , where  $j \in G$ . Thus,  $b_i(\text{noop}) = 0$ . Let  $\lambda \in \mathbb{I}^m$ . The value of  $\lambda$  given  $\text{noop}$ , i.e.,  $v_i(\text{noop}, \lambda)$ , is the maximal sum of  $b_i(\beta) \cdot \text{sat}_i(\beta)$  in a solution  $\text{sat}_i$ , for all  $\beta \in \text{Node}(\text{noop})$ . Since  $\text{Node}(\text{noop}) = \{\text{noop}\}$  and  $b_i(\text{noop}) = 0$ , for any solution  $\text{sat}_i$ ,  $v_i(\text{noop}, \lambda) = 0$ .

**Lemma 3**  $M_{vcg}$  is an ST-model and it is a model of  $\Sigma_{vcg}$ .

**Proof (Sketch)** It is routine to check that  $M_{vcg}$  is actually an ST-model. Given a path  $\delta$  in  $M_{vcg}$  and a stage  $t$  of  $\delta$ , we need to show that  $M_{vcg}, \delta, t \models \varphi$ , for each  $\varphi \in \Sigma_{vcg}$ .

Let us verify Rule 1. Assume  $M_{vcg}, \delta, t \models \text{initial}$ , then  $\delta[t] = \bar{w}$ , i.e.,  $\delta[t] = \langle 1, 0, \dots, 0, 0, \dots, 0 \rangle$ . By the definitions of  $\pi_Y$  and  $\pi_\Phi$ ,  $\pi_\Phi(\bar{w}) = \{\text{bidRound}\}$ ,  $\pi_Y(\bar{w}, \text{payment}_i) = 0$  and  $\pi_Y(\bar{w}, \text{trade}_{i,j}) = 0$  for all  $i \in \mathbb{N}$  and  $j \in G$ . Thus,  $M_{vcg}, \delta, t \models \text{bidRound} \wedge \bigwedge_{i \in \mathbb{N}} \text{payment}_i = 0 \wedge \bigwedge_{j \in G} \text{trade}_{i,j} = 0$ .

Now we verify Rule 4. Assume  $M_{vcg}, \delta, t \models \text{initial}$ , then  $\delta[t] = \bar{w}$  and for all  $i \in \mathbb{N}$  and  $\beta \in \mathcal{B}$ ,  $(\bar{w}, i, \beta) \in L$ . Thus,  $M_{vcg}, \delta, t \models \text{legal}_i(\beta)$ .

Then we consider Rule 5.  $M_{vcg}, \delta, t \models \text{does}(\beta) \wedge \text{initial}$ , for  $\beta \in \mathcal{B}^m$ , i.e.,  $M_{vcg}, \delta, t \models \bigwedge_{i \in \mathbb{N}} \text{does}(\beta_i)$  and  $M_{vcg}, \delta, t \models \text{initial}$ . Thus,  $\theta_i(\delta, t) = \beta_i$  for all  $i \in \mathbb{N}$ . The update function  $U$  defines  $\delta[t + 1]$  such that  $\pi_Y(\delta[t + 1], \text{trade}_{i,j}) = \text{WD}_{\lambda_{i,j}}(\beta, X)$ , for each  $i \in \mathbb{N}$  and  $j \in G$ . Thus,  $M_{vcg}, \delta, t + 1 \models \bigwedge_{i \in \mathbb{N}, j \in G} \text{trade}_{i,j} = \text{WD}_{\lambda_{i,j}}(\beta, X)$  and also  $M_{vcg}, \delta, t \models \text{O}(\bigwedge_{i \in \mathbb{N}, j \in G} \text{trade}_{i,j} = \text{WD}_{\lambda_{i,j}}(\beta, X))$ . Using the abbreviation,  $M_{vcg}, \delta, t \models \text{O}(\bigwedge_{i \in \mathbb{N}} \text{trade}_i = \text{WD}_{\lambda_i}(\beta, X))$ .

The remaining rules are verified in a similar way.

**Lemma 4** For each agent  $i \in \mathbb{N}$ ,  $M_{vcg} \models \text{initial} \wedge \text{does}_i(\text{noop}) \rightarrow \text{Opayment}_i = 0 \wedge \bigwedge_{j \in G} \text{trade}_{i,j} \geq 0$

**Proof** Straightforward from Lemma 2, Rule 6 from  $\Sigma_{vcg}$  and Rule R2 from the definition of  $b_i$ .

**Proposition 6**  $M_{vcg} \not\models \text{Osbb}$  and  $M_{vcg} \not\models \text{Owbb}$ .

**Proof** We show  $M_{vcg} \not\models \text{Owbb}$  and  $M_{vcg} \not\models \text{Osbb}$  by showing a counter example. Given two distinct agents  $i, r$  in  $\mathbb{N}$  and a good  $j$  in  $G$ , let  $\delta$  be a path in  $M_{vcg}$  such that  $\theta_i(\delta, 0) = \langle 1, j, 5 \rangle$  and  $\theta_r(\delta, 0) = \langle -1, j, -3 \rangle$ . For any other agent  $s \in \mathbb{N} \setminus \{i, r\}$ ,  $\theta_s(\delta, 0) = \text{noop}$ . Since this actions are legal in the initial state  $\delta[0]$ , there exists such path.

By Rule 5 and the definition of function  $\text{WD}_\lambda$ , we have that  $M_{vcg}, \delta, 1 \models (\text{trade}_{i,j} = 1 \wedge \text{trade}_{r,j} = -1)$  and  $M_{vcg}, \delta, 1 \models \text{trade}_{s,j'} = 0$  for all pairs  $(s, j') \neq (i, j)$  and  $(s, j') \neq (r, j)$ . That is, the good  $j$  is sold by  $r$  and bought by  $i$ . The social welfare of all agents other than  $i$  is  $-3$  and the social welfare of all agents other than  $r$  is  $5$ . If either  $i$  or  $r$  did not participate, there would be no trade and the social welfare would be zero. Thus, by Rule 6, the payments for  $i$  and  $r$  are  $3$  and  $-5$ , resp., that is,  $M_{vcg}, \delta, 1 \models \text{payment}_i = 3 \wedge \text{payment}_r = -5$ . The other agents' do not have payments on  $\delta[1]$ , i.e.,  $M_{vcg}, \delta, 1 \models \bigwedge_{s \in \mathbb{N} \setminus \{i, r\}} \text{payment}_s = 0$ . Then,  $M_{vcg}, \delta, 1 \models \text{sum}_{i \in \mathbb{N}}(\text{payment}_i) < 0$ . Thus we have that  $M_{vcg}, \delta, 0 \not\models \text{Osbb}$  and  $M_{vcg}, \delta, 0 \not\models \text{Owbb}$ .

**Proposition 7** Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{vcg} \models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$

**Proof (Sketch)** Since  $M_{vcg} \models \text{initial} \rightarrow \text{Oterminal}$  and  $\delta[t + 1] = \delta[t]$  whenever  $\delta[t] \in T$ , it suffices to show that  $M_{vcg}, \delta, 0 \models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$ , for any joint action  $\beta \in \mathcal{B}^n$ . Assume  $M_{vcg}, \delta, 0 \models \text{does}(\beta)$ , by Rule 5 we have that  $M_{vcg}, \delta, 1 \models \bigwedge_{i \in N, j \in G} \text{trade}_{ij} = \text{WD}_{\lambda_{ij}}(\beta, X)$ . By the definition of the winner determination function  $\text{WD}_{\lambda}(\beta, X)$  (and consequently function  $\text{WD}_{\lambda_{ij}}$  for each agent  $i$  and good  $j$ ), we have that the trade  $(\pi_Y(\delta[1], \text{trade}_{ij}))_{j \in G, i \in N}$  maximizes the cumulative value among the agents. That is,  $M_{vcg}, \delta, 1 \models \text{sum}_{i \in N}(v_i(\beta_i, (\text{trade}_j)_{j \in G})) = \max_{\lambda \in \Lambda}(\text{sum}_{i \in N}(v_i(\beta_i, \lambda)))$  and  $M_{vcg}, \delta, 0 \models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$ .

**Proposition 8** Given a joint action  $\beta \in \mathcal{B}^n$ , if  $\vartheta_i(\lambda) \geq 0$  for all  $\lambda \in I^{nm}$ ,  $\vartheta_i \in V_i$  and  $i \in N$ , then  $M_{vcg} \models \text{does}(\beta) \rightarrow \text{Oir}(\beta)$ .

**Proof** Let  $\beta \in \mathcal{B}^n$  be a joint action and  $\delta$  be a path in  $M_{vcg}$ . Since  $M_{vcg} \models \text{initial} \rightarrow \text{Oterminal}$  and  $\delta[t + 1] = \delta[t]$  whenever  $\delta[t] \in T$ , it suffices to show that  $M_{vcg}, \delta, 0 \models \text{does}(\beta) \rightarrow \text{Oir}(\beta)$ . Assume  $M_{vcg}, \delta, 0 \models \text{does}(\beta)$  and that the preferences represented by  $L(\delta[1], i)$  are non-negative for every agent and possible trade. That is,  $v_i(\beta, \lambda) \geq 0$  for any bid  $\beta$ , trade  $\lambda$  and agent  $i$ .

Let  $\lambda = (\lambda_j)_{j \in G}$  denote the trade performed after the agents report  $\beta$ , where  $\lambda_{ij} = \pi_Y(\delta[1], \text{trade}_{ij}) = \text{WD}_{\lambda_{ij}}(\beta, X)$  for each good  $j$  and agent  $i$ . We denote by  $\lambda^{-i} = \text{WD}_{\lambda}^{-i}(\beta, X)$  the trade that would happen if  $i$  did not participate.

The utility of agent  $i$  in  $\delta[1]$  is  $u_i = v(\beta_i, \lambda) - \pi_Y(\delta[1], \text{payment}_i)$ . According to the payment definition (Rule 6), agent  $i$ 's utility in  $\delta[1]$  is

$$u_i = v_i(\beta_i, \lambda) - \left( \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i}) - \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda) \right)$$

or

$$u_i = v_i(\beta_i, \lambda) - \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i}) + \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda)$$

or simply

$$u_i = \sum_{r \in N} v_r(\beta_r, \lambda) - \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i})$$

Thus, agent  $i$ 's utility is non-negative if

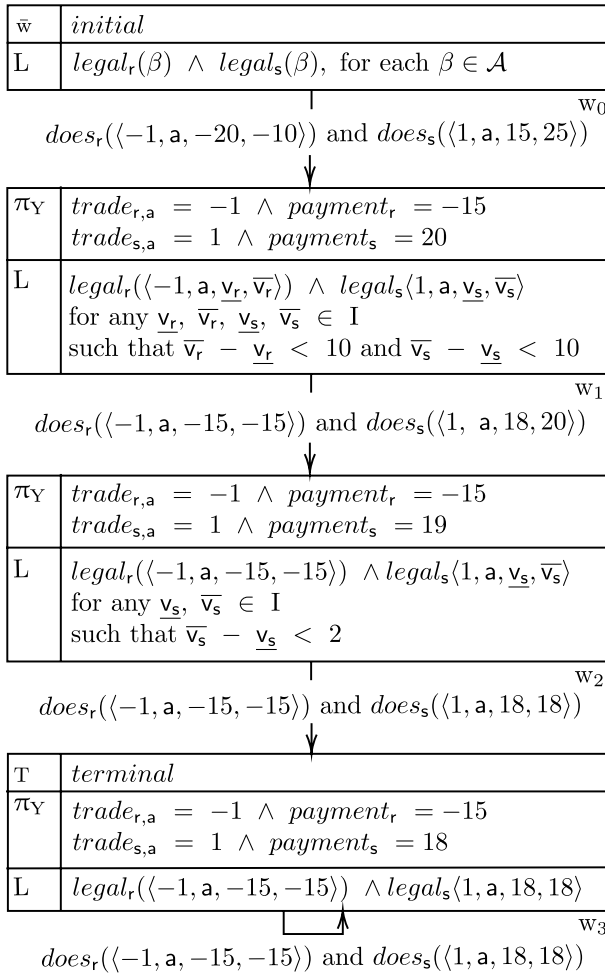
$$\sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda) \geq \sum_{r \in N} v_r(\beta_r, \lambda^{-i})$$

Assume for contradiction that this is not the case, that is,

$$\sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i}) > \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda)$$

Since  $v_r(\beta_r, \lambda^{-i}) \geq 0$ , then we have

$$\sum_{r \in N} v_r(\beta_r, \lambda^{-i}) \geq \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda^{-i}) > \sum_{r \in N \setminus \{i\}} v_r(\beta_r, \lambda)$$



**Fig. 7** A path in  $M_{ice}$  where two agents trade a good

which is a contradiction since  $\lambda$  is the efficient trade (see Proposition 7). Thus,  $M_{vcg}, \delta, 1 \models \bigwedge_{r \in N} \text{sub}(v_r(\beta_r, (\text{trade}_j)_{j \in G}), payment_r) \geq 0$  and  $M_{vcg}, \delta, 0 \models \text{Oir}(\beta)$ .

**Lemma 5**  $M_{ice}$  is an ST-model and it is a model of  $\Sigma$ .

**Proof** The proof is similar to those for Lemmas 1 and 3.

**Proposition 9**  $M_{ice} \not\models \text{Osb}$  and  $M_{ice} \models \text{Owbb}$ .

**Proof (Sketch)** Considering strong budget-balance, notice the path  $\delta$  illustrated at Fig. 7 is a counterexample. For instance, in the last stage we have  $M_{ice}, \delta, 3 \models \text{sum}_{i \in N} (payment_i) = 3$  and thus  $M_{ice} \not\models \text{Osum}_{i \in N} (payment_i) = 0$  and  $M_{ice} \not\models \text{Osb}$ .

Now we consider weak budget-balance. Let  $\delta$  be a path in  $M_{ice}$ ,  $t \geq 0$  a stage in  $\delta$  and  $i \in N$ . By the definition of  $v_i$ , we have that an empty trade is valued zero, i.e.,

$v_i(\theta_i(\delta, 0), (0, \dots, 0)) = 0$ . Thus,  $\sum_{r \in N} v_r(\theta_r(\delta, 0), (0, \dots, 0)) = 0$ . Notice the empty trade  $(0, \dots, 0)$  satisfies Constraints C1–C4 from the winner determination. Among the trades satisfying those constraints,  $WD^\epsilon$  selects the trade  $(\lambda_j)_{j \in G}$  that maximizes the cumulative value among all agents. Thus, the cumulative value cannot be smaller than zero. Since the agents' pay the value of their bids, it follows that  $M_{ice}, \delta, t \models \text{Osub}_{i \in N}(\text{payment}_i) \geq 0$ .

**Proposition 10** Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{ice} \models \text{does}(\beta) \rightarrow \text{Oir}(\beta)$ .

**Proof** The proof is straightforward since  $M_{ice} \models \text{does}_i(\beta_i) \rightarrow \text{Opayment}_i = v_i(\beta_i, (\text{trade}_j)_{j \in G})$ , for each  $i \in N$  and  $\beta_i \in \mathcal{B}$ .

**Proposition 11** Given a joint action  $\beta \in \mathcal{B}^n$ ,  $M_{ice} \models \text{does}(\beta) \rightarrow \text{Oef}(\beta)$

**Proof (Sketch)** The proof is similar to the proof for Proposition 11.

**Proposition 12**  $M_{ice}$  is not strategyproof.

**Proof (Sketch)** We show  $M_{ice}$  is not strategyproof with a counterexample. Assume the path  $\delta$  illustrated in Fig. 7 and consider stage 2. We have  $\theta_s(\delta, 2) = \langle 1, a, 18, 18 \rangle$ . Since the agents pay their reported valuation, in  $\delta[3]$ , the utility of agent  $s$  given her bid is zero, i.e.,  $M_{ice}, \delta, 3 \models \text{sub}(v_s(\langle 1, a, 18, 18 \rangle, (\text{trade}_j)_{j \in G}), \text{payment}_s) = 0$ . Let  $\delta'$  be a path in  $M_{ice}$  defined exactly like  $\delta$ , except by the actions performed by  $s$  in each stage  $t \geq 2$ , which is defined as  $\theta_s(\delta', t) = \langle 1, a, 16, 16 \rangle$ . Then,  $M, \delta', 3 \models \text{Osub}(v_s(\langle 1, a, 18, 18 \rangle, (\text{trade}_j)_{j \in G}), \text{payment}_s) = 2$  and  $M, \delta', 3 \not\models \text{Osub}(v_s(\langle 1, a, 18, 18 \rangle, (\text{trade}_j)_{j \in G}), \text{payment}_s) \leq 0$ .

**Acknowledgements** This research is supported by the ANR Project AGAPE ANR-18-CE23-0013. The authors would like to thank the reviewers for carefully reading of the manuscript and for providing insightful comments.

## References

1. Alur, R., Henzinger, T. A., & Kupferman, O. (2002). Alternating-time temporal logic. *Journal of the ACM*, 49(5), 672–713.
2. Badica, A., & Badica, C. (2010). Specification and verification of an agent-based auction service. In G. A. Papadopoulos, W. Wojtkowski, G. Wojtkowski, S. Wrycza, & J. Zupancic (Eds.), *Information systems development: Towards a service provision society* (pp. 239–248). Boston: Springer.
3. Bădică, C., Ganzha, M., & Paprzycki, M. (2006). Rule-based automated price negotiation: Overview and experiment. In L. Rutkowski, R. Tadeusiewicz, L. A. Zadeh, & J. M. Żurada (Eds.), *Artificial intelligence and soft computing—ICAISC 2006* (pp. 1050–1059). Berlin, Heidelberg: Springer Berlin Heidelberg.
4. Baral, C., & Uyan, C. (2001). Declarative specification and solution of combinatorial auctions using logic programming. In T. Eiter, W. Faber, & M. I. Truszczynski (Eds.), *Logic programming and non-monotonic reasoning* (pp. 186–199). Berlin: Springer Berlin Heidelberg.
5. Barthe, G., Gaboardi, M., Arias, E., Hsu, J., Roth, A., & Strub, P. Y. (2016). Computer-aided verification for mechanism design. In *Conference on web and internet economics*.
6. Belardinelli, F. (2014). Model checking auctions as artifact systems: Decidability via finite abstraction. In *Proceedings of the European conference on artificial intelligence (ECAI'14)* (pp. 81–86).

7. Bellosta, M., Kornman, S., & Vanderpooten, D. (2005). A framework for multiple criteria English reverse auctions. In *IEEE/WIC/ACM international conference on intelligent agent technology* (pp. 633–639).
8. Bellosta, M. J., Kornman, S., & Vanderpooten, D. (2008). A unified framework for multiple criteria auction mechanisms. *Web Intelligence and Agent Systems*, 6(4), 401–419.
9. Boutilier, C., & Hoos, H. H. (2001). Bidding languages for combinatorial auctions. In B. Nebel (Ed.), *Proceedings of the 7th international joint conference on artificial intelligence (IJCAI-01)* (pp. 1211–1217). Seattle, Washington: Morgan Kaufmann.
10. Brihaye, T., Da Costa, A., Laroussinie, F., & Markey, N. (2009). Atl with strategy contexts and bounded memory. In *International symposium on logical foundations of computer science* (pp. 92–106). Springer.
11. Burghardt, J. (2002). Axiomatization of finite algebras. In *Annual conference on artificial intelligence* (pp. 222–234). Springer.
12. Caminati, M., Kerber, M., Lange, C., & Rowat, C. (2015). Sound auction specification and implementation. In *ACM conference on economics and computation*.
13. Cavallo, R., Parkes, D. C., Juda, A. I., Kirsch, A., Kulesza, A., Lahaie, S., Lubin, B., Michael, L., & Shneidman, J. (2005). TBBL: A tree-based bidding language for iterative combinatorial exchanges. In *Multidisciplinary workshop on advances in preference handling*, Edinburgh.
14. Chatterjee, K., Henzinger, T. A., & Piterman, N. (2010). Strategy logic. *Information and Computation*, 208(6), 677–693.
15. Chevaleyre, Y., Dunne, P. E., Endriss, U., Lang, J., Lemaître, M., Maudet, N., et al. (2006). Issues in multiagent resource allocation. *Informatica*, 30(1), 3–31.
16. Conitzer, V., & Sandholm, T. (2002). Complexity of mechanism design. In A. Darwiche, N. Friedman (Eds.), *UAI'02, proceedings of the 18th conferences in uncertainty in artificial intelligence*, University of Alberta, Edmonton, Alberta, Canada, August 1–4, 2002 (pp. 103–110). Morgan Kaufmann.
17. Cramton, P. (2011). Simultaneous ascending auctions, chap. 4. American Cancer Society.
18. Dobricianu, A., Biscu, L., & Badica, C. (2007). Adding a declarative representation of negotiation mechanisms to an agent-based negotiation service. In *2007 IEEE/WIC/ACM international conferences on web intelligence and intelligent agent technology—workshops* (pp. 471–474).
19. Feige, U., Feldman, M., Immorlica, N., Izsak, R., Lucier, B., & Syrgkanis, V. (2015). A unifying hierarchy of valuations with complements and substitutes. In *Proceeding of the AAAI conference on artificial intelligence*, vol. 29(1).
20. Finnsson, H. (2012). Generalized Monte-Carlo tree search extensions for general game playing. In *Twenty-sixth AAAI conference on artificial intelligence*.
21. Genesereth, M., & Thielscher, M. (2014). *General game playing. Synthesis lectures on artificial intelligence and machine learning*. Morgan & Claypool Publishers.
22. Giovannucci, A., Cerquides, J., Endriss, U., & Rodríguez-Aguilar, J. A. (2010). A graphical formalism for mixed multi-unit combinatorial auctions. *Autonomous Agents and Multi-Agent Systems*, 20(3), 342–368.
23. Green, J., & Laffont, J. J. (1979). *Incentives in public decision-making*. North-Holland: Elsevier.
24. Hudert, S., Eymann, T., Ludwig, H., & Wirtz, G. (2009). A negotiation protocol description language for automated service level agreement negotiations. In *2009 IEEE conference on commerce and enterprise computing, CEC 2009* (pp. 162–169).
25. Hudert, S., Ludwig, H., & Wirtz, G. (2009). Negotiating SLAs—An approach for a generic negotiation framework for WS-agreement. *Journal of Grid Computing*, 7(2), 225–246.
26. Jackson, M. O. (2009). *Optimization and operations research -Volume III, chap. Mechanism Theory*. EOLSS Publications.
27. Jiang, G., Perrussel, L., Zhang, D., Zhang, H., & Zhang, Y. (2019). Game equivalence and bisimulation for game description language. In A. C. Nayak & A. Sharma (Eds.), *PRICAI 2019: Trends in artificial intelligence—16th Pacific Rim international conference on artificial intelligence* (Vol. 11670, pp. 583–596). Berlin: Springer.
28. Jiang, G., Zhang, D., Perrussel, L., & Zhang, H. (2016). Epistemic GDL: A logic for representing and reasoning about imperfect information games. In *IJCAI International Joint Conference on Artificial Intelligence*.
29. Jiang, G., Zhang, D., Perrussel, L., & Zhang, H. (2021). Epistemic GDL: A logic for representing and reasoning about imperfect information games. *Artificial Intelligence*, 294, 103453.
30. de Jonge, D., & Zhang, D. (2021). GDL as a unifying domain description language for declarative automated negotiation. *Autonomous Agents and Multi-Agent Systems*, 35(1), 13.
31. Jonge, D. D., & Zhang, D. (2016). Using GDL to represent domain knowledge for automated negotiations. In N. Osman, C. Sierra (Eds.), *Autonomous agents and multiagent systems: AAMAS 2016*

- workshops, visionary papers*, Singapore, Singapore, May 9–10, 2016, Revised Selected Papers (pp. 134–153). Cham: Springer.
32. Jonge, D. D., & Zhang, D. (2017). Automated negotiations for general game playing. In K. Larson, M. Winikoff, S. Das, E. Durfee (Eds.), *Proceedings of the 16th conference on autonomous agents and multiagent systems*, AAMAS 2017, São Paulo, Brazil, May 8–12, 2017 (pp. 371–379). ACM.
  33. Kalagnanam, J., & Parkes, D. C. (2004). Auctions, bidding and exchange design. In D. Simchi-Levi, S. D. Wu, & Z. J. Shen (Eds.), *Handbook of quantitative supply chain analysis: Modeling in the e-business era* (pp. 143–212). Boston: Springer.
  34. Kerber, M., Lange, C., & Rowat, C. (2016). An introduction to mechanized reasoning. *Journal of Mathematical Economics*, 66, 26–39.
  35. Klemperer, P. (1999). Auction theory: A guide to the literature. *Journal of Economic Surveys*, 13(3), 227–286.
  36. Krishna, V. (2009). *Auction theory*. London: Academic Press.
  37. Larsen, G. K. H., van Foreest, N. D., & Scherpen, J. M. A. (2013). Distributed control of the power supply-demand balance. *IEEE Transactions on Smart Grid*, 4(2), 828–836.
  38. Lee, H. G., & Lee, R. (1997). A hybrid approach of linear programming and logic modeling for the market core of sealed bid auctions. *Annals of Operations Research*, 75, 1997.
  39. Lochner, K. M., & Wellman, M. P. (2004). Rule-based specification of auction mechanisms. In *Proceedings of the third international joint conference on autonomous agents and multiagent systems*, AAMAS 2004, 818–825.
  40. Love, N., Genesereth, M., & Hinrichs, T. (2006). *General game playing: Game description language specification*. Technical Report, LG-2006-01, Stanford University, Stanford, CA.
  41. Lubin, B., Juda, A. I., Cavallo, R., Lahaie, S., Shneidman, J., & Parkes, D. C. (2008). Ice: An expressive iterative combinatorial exchange. *Journal of Artificial Intelligence Research*, 33, 33–77.
  42. Maubert, B., Mittelmann, M., Murano, A., & Perrussel, L. (2021). Strategic reasoning in automated mechanism design. In *Proceedings of the eighteen conference on principles of knowledge representation and reasoning*.
  43. Meyer, T., Foo, N., Kwok, R., & Zhang, D. (2004). Logical foundations of negotiation: Outcome, concession and adaptation. *Proceedings of the national conference on artificial intelligence* (pp. 293–298).
  44. Mishra, D., & Sharma, T. (2018). A simple budget-balanced mechanism. *Social Choice and Welfare*, 50(1), 147–170.
  45. Mittelmann, M., Bouveret, S., & Perrussel, L. (2021). A general framework for the logical representation of combinatorial exchange protocols. In F. Dignum, A. Lomuscio, U. Endriss, A. Nowé, A. (Eds.), *AAMAS'21: 20th international conference on autonomous agents and multiagent systems* (pp. 1602–1604). ACM.
  46. Mittelmann, M., & Perrussel, L. (2020). Auction description language (ADL): A general framework for representing auction-based markets. In G. de Giacomo (Eds.), *ECAI 2020*. Santiago de Compostela: IOS Press.
  47. Mogavero, F., Murano, A., Perelli, G., & Vardi, M. (2014). Reasoning about strategies: On the model-checking problem. *ACM Transactions on Computational Logic*, 15(4), 2014.
  48. Mogavero, F., Murano, A., Perelli, G., & Vardi, M. Y. (2014). Reasoning about strategies: On the model-checking problem. *ACM Transaction on Computational Logic (TOCL)*, 15(4), 1–47.
  49. Myerson, R. B., & Satterthwaite, M. A. (1983). Efficient mechanisms for bilateral trading. *Journal of Economic Theory*, 29(2), 265–281.
  50. Nisan, N. (2000). Bidding and allocation in combinatorial auctions. In *ACM conference on electronic commerce* (pp. 1–12).
  51. Nisan, N. (2004). Bidding languages. *Combinatorial Auctions* (pp. 1–19).
  52. Nisan, N., Roughgarden, T., Tardos, É., & Vazirani, V. (2007). *Algorithmic game theory*. Cambridge: Cambridge University Press.
  53. Parkes, D. C., Cavallo, R., Elprin, N., Juda, A., Lahaie, S., Lubin, B., Michael, L., Shneidman, J., & Sultan, H. (2005). ICE: An Iterative Combinatorial Exchange. In *Proceedings of the 6th ACM conference on electronic commerce, EC'05* (pp. 249–258). Association for Computing Machinery, New York, NY, USA.
  54. Parkes, D. C., & Ungar, L. H. (2001). *Iterative combinatorial auctions: Achieving economic and computational efficiency*. Philadelphia, PA: University of Pennsylvania.
  55. Parsons, S., Rodriguez-Aguilar, J. A., & Klein, M. (2011). Auctions and bidding: A guide for computer scientists. *ACM Computing Surveys*, 43(2), 2011.
  56. Pauly, M., & Wooldridge, M. (2003). Logic for mechanism design—A manifesto. In *Workshop on game theory and decision theory in agent systems (GTDT)*.

57. Porello, D., & Endriss, U. (2010). Modelling combinatorial auctions in linear logic. In *Proceedings of the twelfth international conference on principles of knowledge representation and reasoning, KR'10* (pp. 71–78). AAAI Press.
58. Rabe, F. (2006) First-order logic with dependent types. In *International joint conference on automated reasoning* (pp. 377–391). Springer.
59. Rolli, D., Luckner, S., Gimpel, H., & Weinhardt, C. (2006). A descriptive auction language. *Electronic Markets*, 16(1), 51–62.
60. Ruan, J., Van Der Hoek, W., & Wooldridge, M. (2009). Verification of games in the game description language. *Journal of Logic and Computation*, 19(6), 1127–1156.
61. Saffidine, A. (2014). The game description language is turing complete. *IEEE Transactions on Computational Intelligence and AI in Games*, 6(4), 320–324.
62. Sandholm, T. (2003). Automated mechanism design: A new application area for search algorithms. In F. Rossi (Ed.), *Principles and Practice of Constraint Programming - CP 2003* (pp. 19–36). Berlin, Heidelberg: Springer Berlin Heidelberg.
63. Schobbens, P. Y. (2004). Alternating-time logic with imperfect recall. *Electronic Notes in Theoretical Computer Science*, 85(2), 82–93.
64. Thielscher, M. (2011). GDL-II. *KI - Künstliche Intelligenz*, 25(1), 63–66.
65. Thielscher, M. (2017). GDL-III: A description language for epistemic general game playing. In *IJCAI international joint conference on artificial intelligence* (pp. 1276–1282).
66. Thielscher, M., & Zhang, D. (2010). *From general game descriptions to a market specification language for general trading agents* (pp. 259–274). Springer Berlin Heidelberg.
67. Voorneveld, M. (2003). Characterization of pareto dominance. *Operations Research Letters*, 31(1), 7–11.
68. Wang, H., Tang, Y., Liu, J., & Chen, W. (2018). A search optimization method for rule learning in board games. In *Pacific Rim international conference on artificial intelligence* (pp. 174–181). Springer.
69. Wooldridge, M., Ågotnes, T., Dunne, P., & Van der Hoek, W. (2007). Logic for automated mechanism design—a progress report. In *Proceedings of AAAI conference on artificial intelligence*.
70. Wooldridge, M., & Parsons, S. (2000). Languages for negotiation. In *Proceedings of the 14th European conference on artificial intelligence, ECAI'00* (pp. 393–397). IOS Press, NLD.
71. Wooldridge, M., & Parsons, S. (2000). On the use of logic in negotiation. In *Proceedings of the autonomous agents workshop on agent communication languages and conversation protocols*.
72. Xia, M., Stallaert, J., & Whinston, A. B. (2005). Solving the combinatorial double auction problem. *European Journal of Operational Research*, 164(1), 239–251.
73. Xu, H., Bates, C., & Shatz, S. (2009). Real-time model checking for skill detection in live online auctions. In *Software engineering research and practice* (pp. 134–140).
74. Xu, H., & Cheng, Y. T. (2007). Model checking bidding behaviors in internet concurrent auctions. *International Journal of Computer Systems Science & Engineering*, 22(4), 179–191.
75. Zhang, D. (2018). A logic for reasoning about game descriptions. In T. Mitrovic, B. Xue, & X. Li (Eds.), *AI 2018: Advances in artificial intelligence* (pp. 38–50). Cham: Springer.
76. Zhang, D. (2020). Behavioural equivalence of game descriptions. In *Australasian joint conference on artificial intelligence* (pp. 307–319). Springer.
77. Zhang, D., & Thielscher, M. (2015). A logic for reasoning about game strategies. In B. Bonet, S. Koenig (Eds.), *Proceedings of the twenty-Ninth AAAI conference on artificial intelligence*, January 25–30, 2015, Austin, Texas, USA (pp. 1671–1677). AAAI Press.
78. Zhang, D., & Thielscher, M. (2015). Representing and reasoning about game strategies. *Journal of Philosophical Logic*, 44(2), 203–236.