Reasoning about Human-Friendly Strategies in Repeated Keyword Auctions

Francesco Belardinelli  
Université d’Evry  
Évry, France  
francesco.belardinelli@univ-evry.fr

Wojciech Jamroga  
SnT, University of Luxembourg  
Institute of Computer Science, PAS  
wojciech.jamroga@uni.lu

Vadim Malvone  
Télécom Paris  
Paris, France  
vadim.malvone@telecom-paris.fr

Munyque Mittelmann  
Université de Toulouse - IRIT  
Toulouse, France  
munyque.mittelmann@irit.fr

Aniello Murano  
University of Naples Federico II  
Naples, Italy  
nello.murano@gmail.com

Laurent Perrussel  
Université de Toulouse - IRIT  
Toulouse, France  
laurent.perrussel@irit.fr

ABSTRACT
In online advertising, search engines sell ad placements for keywords continuously through auctions. This problem can be seen as an infinitely repeated game since the auction is executed whenever a user performs a query with the keyword. As advertisers may frequently change their bids, the game will have a large set of equilibria with potentially complex strategies. In this paper, we propose the use of natural strategies for reasoning in such setting as they are processable by artificial agents with limited memory and/or computational power as well as understandable by human users. To reach this goal, we introduce a quantitative version of Strategy Logic with natural strategies in the setting of imperfect information. In a first step, we show how to model strategies for repeated keyword auctions and take advantage of the model for proving properties evaluating this game. In a second step, we study the logic in relation to the distinguishing power, expressivity, and model-checking complexity for strategies with and without recall.

KEYWORDS
Mechanism Design; Auctions; Strategic Reasoning

ACM Reference Format:
Francesco Belardinelli, Wojciech Jamroga, Vadim Malvone, Munyque Mittelmann, Aniello Murano, and Laurent Perrussel. 2022. Reasoning about Human-Friendly Strategies in Repeated Keyword Auctions. In Proc. of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), Online, May 9–13, 2022, IFAAMAS, 10 pages.

1 INTRODUCTION
In recent years a wealth of logic-based languages have been introduced to reason about the strategic abilities of autonomous agents in multi-agent systems (MAS), including Alternating-time Temporal Logic (ATL) [8], Strategy Logic (SL) [27, 54], and Game Logic [60], just to name a few. In conjunction with model checking techniques [10], these formal languages have allowed for the development of efficient verification tools [25, 36, 51], which have been successfully applied to the certification of MAS as different as voting protocols [13, 45], robot swarms [32, 49], and business processes [30, 38].

Still, verification tools and techniques are comparatively less developed for data-driven and data-intensive systems, that is, contexts where the data content of processes, or agents, is key to model and account for the evolution of the system [15, 55]. This is the case also for online advertising, where search engines sell ad placements for keywords continuously through auctions. This problem can be seen as an infinitely repeated game since the auction is executed whenever a user performs a query with the keyword. As advertisers may frequently change their bids, the game will have a large set of equilibria with potentially complex strategies, thus making the specification and verification of keyword auctions a complex problem to solve for current model checking methods.

In this paper, we investigate the use of natural strategies [46, 47] for reasoning about equilibria in keyword auctions. The work in [46, 47] followed some classical research on human concept learning [18, 35], social norms [67, 68], commonsense reasoning [29], automated planning [37, 66] and the psychology of planning [56], as well as robotics [58, 62]; in short, it proposed to model “human-friendly” strategies by lists of condition-action pairs with bounded complexity. This was in contrast to “combinatorial” strategies, defined as functions from (sequences of) states to actions, and typically used in the semantics of MAS logics [8, 27, 54, 60]. It was argued in [46, 47] that natural strategies provide better models of behaviour for agents with limited memory and computing capacity, such as humans or simple bots. The concept have been already used to redefine some security requirements for voting protocols in [45].

In our case, the bidding strategy in an auction should be executable for a simple artificial agent, as well as reasonably transparent to the human user, which makes natural strategies a good match. Moreover, natural strategies provide a way to define complexity (and hence also “simplicity”) metrics for various functionality, security, and usability properties in MAS. By focusing on simple strategies, one can make the verification of equilibrium properties

Proced is the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2022), P. Faliszewski, V. Mancardi, C. Pelachaud, M.E. Taylor (eds.), May 9–13, 2022, Online. © 2022 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.
decidable, or even tractable, despite the prohibitive complexity of the general problem. This is especially evident for strategies with memory, which normally make the synthesis and model checking problems undecidable [31, 72].

**Contribution** By leveraging on natural strategies, we introduce a quantitative semantics for SL with natural strategies and imperfect information. As a first contribution, we show how to represent popular strategies for repeated keyword auctions in the proposed framework, as well as prove properties pertaining to this game. Second, we analyse our novel variant of SL in relation with its distinguishing power, expressivity, and complexity of the model checking problem, for natural strategies with and without recall.

**Outline** In Section 2, we recall the basic definitions. Section 3 introduces the new logic NatSL[$F$]. In Section 4 we use it to analyse repeated keyword auctions. The expressivity of NatSL[$F$] is investigated in Section 5. Section 6 establishes the complexity of model checking, and Section 7 concludes the paper. The omitted proofs are available at http://arxiv.org/abs/2201.09616.

### 1.1 Related work

Recently, there have been efforts to apply formal methods to the (semi-)automatic verification of some decision-making mechanisms, including auctions and voting protocols. A number of works [12, 23, 48] expresses these mechanisms in high-level specification languages. However, in contrast with standard model checking techniques [10], their verification is not fully automated, but only assisted by a reasoner. Troquard et al. [69] introduce a framework for fully-automated verification of voting protocols. Still, their approach can only model one-shot mechanisms and thus does not capture multi-stage protocols and repeated auctions. In preliminary works, Pauly and Wooldridge [61] and Wooldridge et al. [75] advocate the use of ATL [8] to reason about decision-making mechanisms. As ATL lacks the expressivity to reason about quantitative aspects such as valuations and payments, and solution concepts such as equilibrium, Maubert et al. [53] introduce SLK[$F$], a quantitative and epistemic version of SL [27, 54], and show how it can be used for reasoning about notions such as Nash equilibrium and strategyproofness. Still, their approach considers strategies as functions from states to actions and cannot handle strategies with recall.

A key assumption of the present contribution is that agents have only partial observability of the global state of the system, as it is often the case in real-life applications. Contexts of imperfect information have been extensively considered in the literature on formal verification [22, 31, 44, 50, 64]. Generally speaking, imperfect information immediately entails higher complexity of game solving. In multi-player games, the complexity can go up to being non-elementary [63], or even undecidability when considered in the context of memoryful strategies [31]. Hence, it is of interest to analyse imperfect information systems where agents have finite or bounded memory, in order to retrieve a decidable model checking problem. Works that are closest in spirit to our contribution concern modeling, specification, and reasoning about strategies of bounded-memory agents. We directly build on the research by Jamroga, Malvone, and Murano on natural strategies [46, 47]. We generalize the approach by considering quantitative semantics for both natural strategies and the logic, which is more suitable for reasoning about mechanisms with monetary transfer (e.g., auctions).

We also consider SL instead of ATL due to its expressive power. In a related vein, Ágones and Walther [2] investigate strategic abilities of agents with bounded memory, while Belardinelli et al. [14] consider bounded memory as an approximation of perfect recall. On a related direction, temporal and strategic logics have been extended to handle agents with bounded resources [5, 6, 20, 21]. Issues related to bounded rationality are also investigated in [11, 39, 43].

Also relevant for the present contribution are papers that study explicit representations of strategies. This category is much richer and includes extensions of ATL* with explicit reasoning about actions and strategies [1, 41, 70, 73], as well as logics that combine features of temporal and dynamic logic [40, 59]. Duijf and Broersen [33] present a variant of STIT logic, that enables reasoning about strategies and their performance in the object language. Also, plans in agent-oriented programming are in fact rule-based descriptions of strategies. In particular, reasoning about agent programs using strategic logics was investigated in [3, 4, 16, 28, 76].

## 2 PRELIMINARIES

We first recall basic notions. For the remainder of the paper, we fix a set of atomic propositions $Ap$, a set of agents $Ag$ and a set of strategy variables $Var$. We let $n$ be the number of agents in $Ag$. Finally, let $F \subseteq \{f : [-1,1]^m \rightarrow [-1,1] \mid m \in \mathbb{N}\}$ be a set of functions over $[-1,1]$ of possibly different arities.

### 2.1 Weighted Concurrent Game Structures

The semantics of natural strategies and NatSL[$F$] are interpreted over weighted concurrent game structures (wCGS). A difference from classical structures is that the labelling of atomic propositions is replaced by a weight function. We consider weighted propositions for easily handling quantitative aspects (such as prices).

**Definition 1.** A weighted concurrent game structure with imperfect information (wCGS) is a tuple $G = (Ac, V, L, \delta, \ell, V_0, \{\sim_a\}_{a \in Ag})$ where: (i) $Ac$ is a finite set of actions; (ii) $V$ is a finite set of states; (iii) $L : Ag \times V \rightarrow 2^{Ac}$ is a legality function, defining the availability of actions; (iv) $\delta$ is a transition function assigning a successor state $v' = \delta(v, (c_a)_{a \in Ag})$ to each state $v$ in $V$ and any tuple of actions $(c_a)_{a \in Ag}$, where $c_a \in L(a, v)$; (v) $\ell : V \times Ap \rightarrow [-1,1]$ is a weight function; (vi) $V_0 \subseteq V$ is a set of initial states; and (vii) $\sim_a \subseteq V \times V$ is an equivalence relation called the observation relation of agent $a$.

We require that the wCGS is uniform, that is $v \sim_a v'$ implies $L(a, v) = L(a, v')$. We write $o$ for a tuple of objects $(o_a)_{a \in Ag}$, one for each agent, and such tuples are called profiles. Given a profile $o$ and $a \in Ag$, we let $o_a$ be agent $a$’s component, and $o \sim a$ is $(o_i)_{i \in Ag \setminus \{a\}}$. Similarly, we let $Ag \sim a = Ag \setminus \{a\}$.

In a state $v \in V$, each player $a$ chooses an available action $c_a \in L(a, v)$, and the game proceeds to state $\delta(c, v) = (c_a)_{a \in Ag}$. A play $\pi = v_0 v_1 v_2 \ldots$ is an infinite sequence of states such that for every $i \geq 0$ there exists an action profile $c$ such that $\delta(v_i, c) = v_{i+1}$. We write $z_i = v_i$ for the state at index $i$ in play $\pi$. A history $h = v_0 v_1 v_2 \ldots v_n$ is a finite sequence of states. The last element of a history is denoted by $last(h) = v_n$. $H_G$ denotes the set of all histories in the wCGS $G$. 
2.2 Natural Strategies

In this section we recall the notion of uniform natural strategies from [47]. Natural strategies are conditional plans, presented through an ordered list of condition-action rules [47]. The intuition is that the first rule whose condition holds in the history of the game is selected, and the corresponding action is executed. As we are considering the setting of imperfect information, the conditions are regular expressions over weighted epistemic (WE) formulas. Given an agent $a$, the WE formulas over $AP$, denoted $WE(AP)$, are conditions on $a$’s knowledge and are expressed by the following Backus-Naur Form grammar:

$$
\psi ::= \top \mid K_a\psi \mid f(\psi, \ldots, \psi) \\
\phi ::= p \mid f(\phi, \ldots, \phi) \mid K_a\phi
$$

where $f \in F$ is a function, $p \in AP$ is an atomic proposition and $i \in AG$ is an agent.

Given a wCGS $G$, a state $v \in V$ and a $WE(AP)$ formula $\phi$, we inductively define the satisfaction value of $\phi$ in $v$, denoted $[\phi](v)$:

$$
[\emptyset](v) = \top(v, p) \\
[K_a\phi](v) = \min_{\sigma' = a^\sigma} [\phi](v') \\
f(\phi_1, \ldots, \phi_m)(v) = f([\phi_1](v), \ldots, [\phi_m](v))
$$

The semantics for the knowledge modality is the standard in the literature on fuzzy epistemic logic (e.g. [52]). Let $Reg(WE(AP))$ be the set of regular expressions over the weighted epistemic conditions $WE(AP)$, defined with the constructors $\cdot, \cup, \cdot^*$ representing concatenation, nondeterministic choice, and finite iteration, respectively. Given a regular expression $r$ and the language $L(r)$ on words generated by $r$, a history $h$ is consistent with $r$ iff there exists $b \in L(r)$ such that $|h| = |b|$ and $[b[i]]([h[i]]) = 1$, for all $0 \leq i \leq |h|$. Intuitively, a history $h$ is consistent with a regular expression $r$ if the $i$-th weighted epistemic condition in $r$ “holds” in the $i$-th state of $h$ (for any position in $i$).

A uniform natural strategy with recall $s_a$ for agent $a$ is a sequence of pairs $(r, c)$, where $r \in Reg(WE(AP))$ is a regular expression, and $c$ is an action available in last$(h)$, for all histories $h \in H_G$ consistent with $r$ and the last pair on the sequence is required to be $(\top, c)$, with $c \in L(a, v)$ for every $v \in V$ and some $c \in Ac$.

A uniform memoryless natural strategy is a special case of natural strategy in which each condition is a weighted epistemic formula (i.e. no regular operators are allowed).

Natural strategies are uniform in the sense they specify the same actions in indistinguishable states (see [47]). We define $Str^a_\rho$ to be the set of uniform natural strategies for agent $a$ and $Str^\rho = \cup_{a \in AG} Str^a_\rho$, where $\rho \in \{ir, iR\}$.

Let $size(s_a)$ denote the number of guarded actions in $s_a$, $cond_i(s_a)$ be the $i$-th guarded condition on $s_a$, $cond_i[j]$ the $j$-th WE formula of the guarded condition $s_a$, and $act_i(s_a)$ be the corresponding action. Finally, $match(h, s_a)$ is the smallest index $i \leq size(s_a)$ such that for all $0 \leq j \leq |last(h)|$, $cond_i[j][h[j]] = 1^4$ and $act_i(s_a) \in L(a, last(h))$.

3 NATURAL STRATEGY LOGIC

SL$[F]$ [19] proposes a quantitative semantics for Strategy Logic, in which strategies are functions mapping histories to actions. For reasoning about intuitive and simple strategies, we introduce SL$[F]$ with natural strategies and imperfect information, denoted NatSL$[F]$. Throughout this section, let $\rho \in \{ir, iR\}$ denote whether the semantics considers memoryless or recall strategies.

An assignment $\chi : AG \cup Var \rightarrow Str^\rho$ is a function from players and variables to strategies. For an assignment $\chi$, an agent $a$ and a strategy $\sigma$ for $a$, $\chi[a \mapsto \sigma]$ is the assignment that maps $a$ to $\sigma$ and is otherwise equal to $\chi$, and $\chi[s \mapsto \sigma]$ is defined similarly, where $s$ is a variable. For an assignment $\chi$ and a state $v$ we let $Out(\chi, v)$ be the unique play that starts in $v$ and follows the strategies assigned by $\chi$. Formally, $Out(\chi, v)$ is the play $v_0v_1\ldots$ such that $v_0 = v$ and for all $i \geq 0$, $v_{i+1} = \delta(v_i, c)$ where for all $a \in AG$, $c_a = act_{match(v_i, x(a))}(\chi(a))$.

3.1 NatSL$[F]$ Syntax

Definition 2. The syntax of NatSL$[F]$ is defined as follows:

$$
\phi ::= p \mid \exists_a^{\leq k} \cdot \phi \mid (a, s_a)\phi \mid f(\phi, \ldots, \phi) \mid X\phi \mid \phi U \phi
$$

where $p \in AP$, $s_a \in Var \cup Str^a_\rho$, $a \in AG$, and $f \in F$.

The intuitive reading of the operators is as follows: $\exists_a^{\leq k} \cdot \phi$ means that there exists a strategy with complexity less or equal than $k$ for agent $a$ such that $\phi$ holds; $(a, s_a)\phi$ means that when strategy $s_a$ is assigned to agent $a$, $\phi$ holds; $X$ and $U$ are the usual temporal operators “next” and “until”. The meaning of $f(\phi_1, \ldots, \phi_n)$ depends on the function $f$. We use $T$, $V$, and $\neg$ to denote, respectively, function 1, function $x, y \mapsto \max(x, y)$ and function $x \mapsto -x$.

A variable is free in formula $\phi$ if it is bound to an agent without being quantified upon, and an agent $a$ is free in $\phi$ if $\phi$ contains a temporal operator ($U$ or $X$) that is not in the scope of any binding for $a$. The set of free variables and agents in $\phi$ is written free$(\phi)$, and a formula $\phi$ is a sentence if free$(\phi) = \emptyset$. The strategy quantifier $\exists_a^{\leq k} \cdot \phi$ quantifies on strategies for agent $a$.

3.2 NatSL$[F]$ Semantics

Definition 3. Let $G = (A, V, \delta, \ell, V_f, (\sim_a)_{a \in AG})$ be a wCGS, and $\chi$ an assignment. The satisfaction value $[\phi]^\chi_X(\sigma) \in [-1, 1]$ of a NatSL$[F]$ formula $\phi$ in a state $\sigma$ is defined as follows, where $\pi$ denotes $Out(\chi, \sigma)$:

$$
[\emptyset]^\chi_X(\sigma) = \ell(\emptyset, p) \\
[\exists_a^{\leq k} \cdot \phi]^\chi_X(\sigma) = \max_{\sigma \in \{a \in AG | Str^\rho \cap \text{comp}(\chi(a) \leq s_a) \}} [\phi]^\chi_X(\sigma) \\
[(a, s_a)\phi]^\chi_X(\sigma) = [\phi]^\chi_X(\sigma \mapsto (a, s_a)) \\
[(a, s_a)\phi]^\chi_X(\sigma) = [\phi]^\chi_X(\sigma \mapsto (a, s_a))$

if $s_a \in Var$

if $s_a \notin Var$
When a user submits a query, an auction is run to determine the allocation of slots in decreasing order of bids and the payment for the bid of the agent allocated to the slot. Each slot has a click-through rate \( \alpha \) for a click, where \( \alpha \) is the probability that the user will click on the advertisement in slot \( i \) for \( i \in \{1, \ldots, n\} \) with \( i \neq j \). In this auction, the prices are 0 and the slots are allocated to agents in decreasing order of bids, with the payment for the bid of the agent allocated to the slot. For a keyword \( V \), the atomic propositional set is \( \mathcal{AP} = \{ a \} \) by the notation \( F = \{ \top, \vee, \neg \} \). NatSL [\( F \)] corresponds to a Boolean-valuated extension of SL with Natural Strategies.

We define the classic abbreviations: \( \bot := \neg \top, \phi \rightarrow \phi' := \neg \phi \lor \phi' \), \( \phi \land \phi' := \neg(\neg \phi \lor \neg \phi') \). For instance, \( \land \) corresponds to \( \land \), \( \top \) computes the supremum of the satisfaction value of \( \phi \) over all future points in time, \( \neg \) computes the infimum of these values, and \( \forall \). \( \exists \). \( \phi \) minimizes the value of \( \phi \) over all possible strategies \( s \).

### 4 REPEATED KEYWORD AUCTIONS

Modeling mechanisms with monetary transfer and private valuations requires handling quantitative features and imperfect information. Memoryless strategies are enough for mechanisms in which the satisfaction value does not depend on the assignment, and we write \( \langle \phi \rangle_{X}^{G}_{\rho}(v) \) for \( \langle \phi \rangle_{X}^{G}_{\rho}(v) \) where \( X \) is any assignment. We also let \( \langle \phi \rangle_{X}^{G}_{\rho} = \min_{v \in V} \langle \phi \rangle_{X}^{G}_{\rho}(v) \).

**Remark 1.** When propositions only take values in \( \{-1, 1\} \) and \( F = \{ \top, \vee, \neg \} \), NatSL [\( F \)] corresponds to a Boolean-valuated extension of SL with Natural Strategies.

For a keyword \( V \), the atomic propositional set is \( \mathcal{AP} = \{ a \} \) by the notation \( F = \{ \top, \vee, \neg \} \). NatSL [\( F \)] corresponds to a Boolean-valuated extension of SL with Natural Strategies.

We define the classic abbreviations: \( \bot := \neg \top, \phi \rightarrow \phi' := \neg \phi \lor \phi' \), \( \phi \land \phi' := \neg(\neg \phi \lor \neg \phi') \). For instance, \( \land \) corresponds to \( \land \), \( \top \) computes the supremum of the satisfaction value of \( \phi \) over all future points in time, \( \neg \) computes the infimum of these values, and \( \forall \). \( \exists \). \( \phi \) minimizes the value of \( \phi \) over all possible strategies \( s \).

### 4.1 Solution concepts for \( \mathcal{G}_{GSP} \)

In this section, we show how NatSL [\( F \)] can be used for the verification of mechanisms with natural strategies. In sight of our motivating example, we aim at rephrasing conditions and properties usually considered in the analysis of keyword auctions [24, 34, 71].

**Nash equilibrium** Since auctions are noncooperative, the solution concept in the pure strategy setting usually considered is the Nash equilibrium (NE). The NE captures the notion of stable solution: a strategy profile is NE if no player can improve her utility through an unilateral change of strategy [65]. With NatSL [\( F \)], we restrict the range of strategies to simple ones, as it enables us to reason about artificial agents with limited capabilities and human-friendly strategies. Let \( \sigma = (\sigma_{a})_{a \in \mathcal{A}} \) be a profile of strategies and \( k > 0 \) denotes the price of slot \( s \) and \( \theta_{a} \) denotes \( a \)'s valuation. Define \( \mathcal{G}_{GSP} = \langle Ac, V, L, \delta, \ell, V_{c}, \{\neg a \}_{a \in \mathcal{A}} \rangle \), where:

- \( Ac = \{ 0 + x \times inc : 0 \leq x \leq \frac{1}{inc} \} \), where \( b \in Ac \) denotes a bid with price \( b \) for a click; given \( c = (c_{a})_{a \in \mathcal{A}} \) let \( rank_{c} = (a_{1}, \ldots, a_{n}) \) be the sequence of distinct agents in \( Ag \) ordered by their bid, that is, \( i < j \) if \( c_{a_{i}} > c_{a_{j}} \) or \( c_{a_{i}} = c_{a_{j}} \) and \( a_{i} < a_{j} \) for \( i, j \in \{1, \ldots, n\} \) with \( i \neq j \). In case of draws, the sequence is determined with respect to \( < \). We let \( rank_{c}(i) \) denote the agent in the \( i \)-th position of the sequence \( rank_{c} \).
- \( V = \{ (a_{1}, \ldots, a_{m}, p_{r_{1}}, \ldots, p_{r_{m}}, (\nu_{a}), a \}_{a \in \mathcal{A}}) : a_{i} \in \mathcal{A}_{G} \} \), where each slot represents the current slot allocation and prices, with \( a_{i} \), \( p_{r_{i}} \), and \( \nu_{a} \) denoting the winner of slot \( s \), the price per click of \( s \) and \( a \)'s valuation, resp.
- For each \( a \in \mathcal{A} \) and \( v \in \mathcal{V} \), \( A(L, v, a) = Ac \);
- For each \( v \in \mathcal{V} \) and \( c = (c_{a})_{a \in \mathcal{A}} \) such that \( c_{a} \in L(a, v) \), the transition function uses the agent’s bids to choose the next allocations and prices, and is defined as follows: \( \delta(v, (c_{a})_{a \in \mathcal{A}}) = (a_{1}', \ldots, a_{m}', p_{r_{1}}', \ldots, p_{r_{m}}', (\nu_{a}), (a_{i} \in \mathcal{A}_{G})) \), where for each agent \( a \) and slot \( s \), (i) \( a_{s} = rank_{c}(s) \) if \( s \leq n \), and \( a_{s} = none \) otherwise; (ii) \( p_{r_{s}} = c_{rank_{c}(s+1)} \) if \( s+1 \leq n \), and \( p_{r_{s}} = 0 \) otherwise.
- For each agent \( a \), slot \( s \in S \) and state \( v = (a_{1}, \ldots, a_{m}, p_{r_{1}}, \ldots, p_{r_{m}}, (\nu_{a}), (a_{i} \in \mathcal{A}_{G})) \), the weight function is defined as follows: (i) \( \ell(v, a_{s}) = 1 \) if \( a_{s} = a \), and \( \ell(v, none) = 0 \) otherwise; (ii) \( v(a_{s}) = p_{r_{s}} \); and (iii) \( \ell(v, none) = v(a_{s}) \).
- In an initial state, the prices are 0 and the slots are allocated to none, that is, \( V_{i} = \{ none, none, none, 0, 0, 0, \ldots, 0 \} \in \mathcal{V} \).
- For each agent \( a \) and two states \( v = (a_{1}, \ldots, a_{m}, p_{r_{1}}, \ldots, p_{r_{m}}, (\nu_{a}), (a_{i} \in \mathcal{A}_{G})) \) and \( v' = (a_{1}', \ldots, a_{m}', p_{r_{1}}', \ldots, p_{r_{m}}', (\nu_{a}'), (a_{i} \in \mathcal{A}_{G})) \) in \( V \), the observation relation \( \sim \) is such that if \( v \sim v' \) then \( i) \( a_{s} = a_{s}', \) for each \( 1 \leq s \leq m \); (ii) \( p_{s} = p_{s}' \), for each \( 1 \leq s \leq m \); (iii) \( \nu_{a} = \nu_{a}' \).

Notice there is exactly one initial state for each possible valuation profile in \( \Pi_{a \in \mathcal{A}} V_{a} \). Additionally, valuations remain unchanged after the initial state. We use the formula \( s^{-1} = 1 \) when it is convenient to obtain a value in \( \{-1, 1\} \) for representing a slot \( s \). The utility of agent \( a \) when she is assigned to slot \( s \) is denoted by the formula \( ut_{a,s} = \theta_{a} - (\theta_{a} - p_{s}) \). The expected utility for agent \( a \) depends on her actual allocation, that is, \( ut_{a} = \sum_{s \in S} all_{a,s} \times ut_{a,s} \).
and define the formula

$$\text{NE}(\sigma, k) := \bigwedge_{a \in Ag} \forall t \leq k. \left( (Ag_{-a}, \sigma_{-a})(a, t)Xut_a \leq (Ag, \sigma)Xut_a \right)$$

The formula \(\text{NE}(\sigma, k)\) means that, for every agent and alternative strategy \(t\) of complexity at most \(k\), binding to \(t\) when everyone else binds to their strategies in \(\sigma\) leads to at most the same utility as when she also binds to her strategy in \(\sigma\). In relation to strategies with complexity at most \(k\), the strategy profile \(\sigma\) leads to a NE in the next state of \(v\) if \(\text{NE}(\sigma, k))^G_{\text{GSP}}(v) = 1\).

Predicting outcomes of a keyword auction is a difficult task given the infinite nature of NE continuum [77]. For this reason, refined solution concepts have been proposed to reduce the NE continuum to subsets. Edelman et al. [34] studied the subset called locally envy-free equilibrium (LEFE), in which no advertiser can improve to subsets. Edelman et al. [24] asks the problem on whether there exists a "natural bidding strategy" for the advertisers that would lead to equilibrium.

**Convergence** The concept of convergence or stabilization can be easily encoded in NatSL[\(\mathcal{F}\)] : we say a wCGS \(\mathcal{G}\) converge to a property \(\varphi\) if the initial states lead to \(\varphi\) being eventually always the case. Formally, a wCGS converge to a condition \(\varphi\) if \(\varphi \in \mathcal{F}\) for all initial state \(s_i \in V_i\) since it holds that if \(\varphi\) is true for the initial states, then it will be true for all future states.

**4.2 Natural Strategies for \(\mathcal{G}_{\text{GSP}}\)**

Given agent \(a\) and the wCGS \(\mathcal{G}_{\text{GSP}}\), we exemplify strategies for \(a\) in a repeated keyword auction. For readability, we omit the epistemic operator \(K_a\) from an epistemic condition \(K_a \varphi\) when the satisfaction value of \(\varphi\) is known by \(a\) in all states. A common approach for an advertiser is to assume that all the other bids will remain fixed in the next round and target the slot that maximizes her utility at current prices. This mechanism allows a range of bids that will result in the same outcome from \(a\)'s perspective, so a number of strategies are distinguished by the bid choice within this range.

**Balanced bidding** In the balanced bidding strategy (BB) [24], the agent bids so as to be indifferent between successfully winning the targeted slot at its current price, or winning a slightly more desirable slot at her bid price. The natural strategy representing balanced bidding for agent \(a\) is denoted \(BB_{a,1}(b)\) and is constructed in three parts. First, include the guarded actions \(BB_{a,2}(b, s)\) for each action \(b \in Ac\). Second, include \(BB_{a,2}(b, s)\) for each \(b \in Ac\) and \(1 < s \leq m\). Third, the last guarded action is \(\langle T, 0 \rangle\). The condition \(BB_{a,1}(b)\) refers to the case in which the slot maximizing \(a\)'s utility is the top slot and \(b\) is \((\varphi_{a} + p_1)/2\):

$$BB_{a,1}(b) := b = \frac{\varphi_{a} + p_1}{2} \land (\max_{s \leq m}(u_{s,a})^{-1} = 1$$

Condition \(BB_{a,2}(b, s)\) denotes the case in which the slot \(s \neq 1\) maximizes \(a\)'s utility and \(b\) is the bid value that is high enough to force the prices paid by her competitors to rise, but not so high that she would mind getting a higher slot at a price just below \(b\):

$$BB_{a,2}(b, s) := u_{s,a} = \varphi_{a+1} \land (\max_{s \leq m}(u_{s,a})^{-1} = s^{-1}$$

Notice the guarded action \(BB_{a,2}(b, s)\) is defined for \(s > 1\) since it compares the utility with the one for \(s = 1\). The case \(s = 1\) is treated by the guarded action \(BB_{a,1}(b)\).
Given a valuation profile $\mathbf{v} = (v_a)_{a \in A}$, let $\eta_x$ be the agent in the $x$-th position of $\text{rank}(a)$ (that is, $\eta_x$ is the agent with $x$-th highest valuation). We let $b_{\eta_x}(\mathbf{v})$ be a function in $\mathcal{F}$ defined as follows:

$$b_{\eta_x}(\mathbf{v}) = \begin{cases} \frac{\partial_v}{\partial x-1} b_{\text{rank}(x+1)}(\mathbf{v}) + (1 - \frac{\partial_v}{\partial x-1}) v_{\eta_x}, & \text{if } x \geq m + 1 \\
v_{\eta_x}, & \text{if } 2 \leq x \leq m \\ 0, & \text{if } x = 1 \end{cases}$$

If $BB = (BB_a)_{a \in A}$ converges to the equilibrium with VCG outcomes, the agent with the highest valuation bids any value above $b_{\eta_x}(\mathbf{v})$. The equilibrium bid for $a \neq \eta_1$ is $b_a(\mathbf{v})$ [24]. When there are two slots and all players update their bids according to BB, the game converges to the equilibrium with VCG outcome. However, this is not the case for more than two slots [24].

**Proposition 3.** For any initial state $v_i \in V$, state $v \in V$, and $1 < x \leq n$, the following holds, where $\mathbf{v} = ((v(x, \eta_x))_{x \in A})$:

1. If $\mathbf{v} = (\mathbf{v}(x, \eta_x))_{x \in A}$, and $\text{actmatch}(v, RR_B) = b_{\eta_x}(\mathbf{v})$

2. If $m = 2$, then $\mathbf{v} = (\mathbf{v}(x, \eta_x))_{x \in A}$, and $\text{actmatch}(v, BB_B) = b_{\eta_x}(\mathbf{v})$

3. If $m > 2$, then $\mathbf{v} = (\mathbf{v}(x, \eta_x))_{x \in A}$.

**Restricted BB**

The restricted balanced bidding strategy (BBB) [24] is a variation of BB in which the agent only targets slots that are no better than her current slot. The natural strategy representing BBB for agent $a$ is denoted $BB_B$, and is constructed as follows.

First, include the guarded actions $BB_B(b) (b, s)$ for each action $b \in Ac$. Second, include $(BB_B(b))_{b \in Ac}$ for each $b \in Ac$ and $1 < s \leq m$. Finally, the last guarded action is $(\top, 0)$. Let $s_a = \min(m, \sum_{t \in \mathcal{S}} x^t \cdot a_{all}^t)$ be the slot assigned to agent $a$ or the last slot if there is no such slot. Define $BB_B^1(b)$ and $BB_B^2(b, s)$:

$$BB_B^1(b) := \frac{\partial_v}{\partial x-1} + p_t \land \underset{a \in A}{\text{argmax}} \quad \frac{\partial_v}{\partial \text{all}^t} = 1$$

$$BB_B^2(b, s) := \text{actmatch}(v, BB_B) = b_{\eta_x}(\mathbf{v}) \land \text{actmatch}(v, \text{BB_B}) > b_{\eta_x}(\mathbf{v})$$

Similar to the results in [24], we have that if all agents follow the restricted balanced bidding strategy, the auction converges to the VCG equilibrium outcome. Always BB converges:

**Proposition 4.** For any initial state $v_i \in V$, state $v \in V$, and $1 < x \leq n$, the following holds, where $\mathbf{v} = ((v(x, \eta_x))_{x \in A})$:

1. If $\mathbf{v} = (\mathbf{v}(x, \eta_x))_{x \in A}$, and $\text{actmatch}(v, BB_B) = b_{\eta_x}(\mathbf{v})$

2. If $m > 2$, then $\mathbf{v} = (\mathbf{v}(x, \eta_x))_{x \in A}$.

**Knowledge grounded BBB**

The knowledge grounded BBB strategy (KBB) is a variation of BBB in which the agent uses her knowledge about the valuations of the player currently at her target slot to ground her bid value. The idea is to avoid bidding more than what she knows her opponent valuates the slot. The natural strategy representing KBB for agent $a$ is denoted $KBB_A$ is constructed in three steps. First, include the guarded actions $KBB_A(b_i, b, c, i)$ for each $b, c \in Ac$ and agent $i \neq a$. Second, include $(KBB_A(b, c, i))_{b, c \in Ac}$ for each $b, c \in Ac$, slot $1 < s \leq m$ and agent $i \neq a$. Finally, include the guarded actions from $BB_B$. The conditions $KBB_A(b, c, i)$ and $KBB_A^2(b, c, i)$ are defined as follows:

$$KBB_A^1(b, c, i) := K_a(BBB_A^1(b, s) \land \text{all}_A = 1 \land c = \min(b_i, b))$$

The prices under KBB are at most the same as under BBB:

**Proposition 5.** For any state $v \in V$, slot $s \in S$ and agent $a$, $[\langle (Ag, \text{KBB}) \rangle p_s \leq \langle (Ag, \text{BBB}) \rangle p_s]$
5 EXPRESSIVITY

In relation to SL[\mathcal{F}] with combinatorial strategies, NatSL[\mathcal{F}] introduces a new, broader class of human-friendly strategies and a language for expressing properties of agents that use such strategies. Clearly, strategies with quantitative conditions can be used to obtain goals that would not be achievable otherwise. On the other hand, bounded natural strategies of NatSL[\mathcal{F}] may not achieve some goals that can be enforced with combinatorial strategies of SL[\mathcal{F}]. In this section, we show that the expressive power of NatSL[\mathcal{F}] is incomparable to that of SL[\mathcal{F}]. In other words, there are properties of quantitative games that cannot be equivalently translated to properties based on combinatorial strategies, and vice versa. From this, we conclude that reasoning about human-friendly strategies differs inherently from the "standard" one.

5.1 Expressive and Distinguishing Power

We first adapt the notions of distinguishing power and expressive power to the quantitative case as follows.\footnote{Cf., e.g., [74] for a detailed discussion of standard notions of expressivity.}

**Definition 4** (Distinguishing power of real-valued logics). Let \mathcal{L}_1 = (L_1, [\cdot]_1) and \mathcal{L}_2 = (L_2, [\cdot]_2) be two logical systems with syntax L_1, L_2 and real-valued semantics [\cdot]_1, [\cdot]_2 over the same class of models M. We say that \mathcal{L}_2 is at least as distinguishing as \mathcal{L}_1 (written: \mathcal{L}_1 \preceq \mathcal{L}_2) if for every pair of models M, M' \in M, if there exists a formula \varphi_1 \in L_1 such that \varphi_1^M \neq \varphi_1^{M'} \text{, then there is also } \varphi_2 \in L_2 with \varphi_2^M \neq \varphi_2^{M'}. In other words, if there is a formula \varphi_1 of \mathcal{L}_1 discerning M from M', then there must be also a formula of \mathcal{L}_2 doing the same.

**Definition 5** (Expressive power of real-valued logics). \mathcal{L}_2 is at least as expressive as \mathcal{L}_1 (written: \mathcal{L}_1 \preceq \mathcal{L}_2) if for every \varphi_1 \in L_1 there exists \varphi_2 \in L_2 such that, for every model M \in M, we have \varphi_1^M = \varphi_2^M. In other words, every formula of \mathcal{L}_1 has a translation in \mathcal{L}_2 that produces exactly the same truth values on models in M.

It is easy to see that \mathcal{L}_1 \preceq \mathcal{L}_2 implies \mathcal{L}_1 \preceq \mathcal{L}_2. Thus, by transposition, we also get that \mathcal{L}_1 \preceq \mathcal{L}_2 implies \mathcal{L}_1 \preceq \mathcal{L}_2.

In the remainder, M is the class of pointed weighted games, i.e., pairs (G, v) where G is a wCGS and v is a state in G.

5.2 Expressivity of \textit{NatSL}[\mathcal{F}] vs. \textit{SL}[\mathcal{F}]

NatSL[\mathcal{F}] and SL[\mathcal{F}] are based on different notions of strategic ability. The former refers to "natural" strategies, represented as mappings from regular expressions over atomic propositions to actions. The latter uses "combinatorial" strategies, represented by mappings from sequences of states to actions. Each natural strategy can be translated to a combinatorial one, but not vice versa. Consequently, SL[\mathcal{F}] can express that a given coalition has a combinatorial strategy to achieve their goal (which is not expressible in NatSL[\mathcal{F}]!). On the other hand, NatSL[\mathcal{F}] allows expressing that a winning natural strategy does not exist (which cannot be captured in SL[\mathcal{F}]). Now we show that NatSL[\mathcal{F}] allows to express properties that cannot be captured in SL[\mathcal{F}], and vice versa.

**Proposition 8.** NatSL[\mathcal{F}] \preceq d SL[\mathcal{F}] in both ir and iR semantics.

**Figure 1:** Model \mathcal{G}_1. Its counterpart \mathcal{G}'_1 is obtained by fixing p to hold only in \varrho_1, \varrho'_1. Underscore fits any action label.

**Proof sketch.** Consider model \mathcal{G}_1 in Figure 1, with agents Ag = {1, 2}, actions Ac_1 = {a_1, b_1, c_1, d_1, e_1}, and Ac_2 = {a_2, b_2} available at all positions, and propositions AP = {p, win}. Both propositions are qualitative (that is, the propositions have only values in \{-1, 1\}). For each proposition, the states where it evaluates to 1 are indicated; otherwise its truth value is assumed to be \(-1\). The outgoing transitions in \varrho_1, \varrho'_1 (resp. \varrho_2) are exact copies of those at \varrho_1 (resp. \varrho_2). Moreover, model \mathcal{G}'_1 is obtained by fixing proposition p to hold only in \varrho_1, \varrho'_1, but not in \varrho'_2. As all the propositions are qualitative, formulas of NatSL[\mathcal{F}] and SL[\mathcal{F}] evaluate to \(-1\) or \(1\). Note also that the sets of ir and iR strategies in each model coincide, so we can concentrate on the ir case w.l.o.g.

Let \mathcal{G} \uparrow \sigma denote the model obtained by fixing the (memoryless) strategy \sigma in G. In order to prove that (\mathcal{G}_1, q_0) and (\mathcal{G}'_1, q_0) satisfy the same formulas of SL[\mathcal{F}], it suffices to observe that:

1. For every strategy \sigma_1 of agent 1 in \mathcal{G}_1, there is \sigma'_1 in \mathcal{G}'_1 such that agent 2 has the same strategic abilities in (\mathcal{G}_1 \uparrow \sigma_1, q_0) and (\mathcal{G}'_1 \uparrow \sigma'_1, q_0) (and vice versa). For instance, playing c_1 in \mathcal{G}_1 obtains the same abilities of 1 as playing \varrho_1 in \mathcal{G}'_1.

2. Analogously for strategies of agent 2, e.g., strategy \varrho_2^a \varrho_2^b \varrho_2^b \varrho_2^b in \mathcal{G}_1 can be simulated by strategy \varrho_2^a \varrho_2^b \varrho_2^b \varrho_2^b in \mathcal{G}'_1.

On the other hand, the formula \exists \varrho_2 \forall \varrho_1 (\varrho_1(1, s_1)(2, s_2) \mathbf{F} \text{win of NatSL}[\mathcal{F}] holds in (\mathcal{G}_1, q_0), but not in (\mathcal{G}'_1, q_0). The winning natural strategy for agent 2 in \mathcal{G}_1 is ((\top^*, p, a_2), (\top^*, b_2)); clearly, it does not succeed in \mathcal{G}'_1.

**Proposition 9.** SL[\mathcal{F}] \preceq d NatSL[\mathcal{F}] in both ir and iR semantics.

**Proof sketch.** Consider models \mathcal{G}_2 and \mathcal{G}'_2 in Figure 2. They have isomorphic action/transition structures, the only difference being the indistinguishability of states \varrho_1, \varrho_2 in \mathcal{G}'_2 (but not in \mathcal{G}_2). Since the two states have the same valuations of propositions, each natural strategy must specify the same decision in \varrho_1, \varrho_2. Thus, both players have exactly the same available natural strategies in \mathcal{G}_2 and \mathcal{G}'_2, and hence (\mathcal{G}_2, q_0) and (\mathcal{G}'_2, q_0) produce the same valuations of NatSL[\mathcal{F}] formulas.

On the other hand, we have that \exists \varrho_2 \forall \varrho_1 \forall \varrho_2 \forall \varrho_1 (\mathbf{F} \text{win of SL}[\mathcal{F}] holds in (\mathcal{G}_2, q_0), but not in (\mathcal{G}'_2, q_0).
The following is an immediate consequence.

**Theorem 1.** NatSL\([F]\) and SL\([F]\) have incomparable distinguishing power over the class of pointed wCGS (in both ir and irR semantics).

**Corollary 2.** NatSL\([F]\) and SL\([F]\) have incomparable expressive power over the class of pointed wCGS (in both ir and irR semantics).

6 MODEL CHECKING

In this section we show that the model checking problem for NatSL\([F]\) with imperfect information is no harder than model checking LTL or classic SL with memoryless agents. First of all, we define the quantitative model-checking problem for NatSL\([F]\).

**Definition 6.** Given \(\rho \in \{ir, irR\}\), the model-checking problem for NatSL\([F]\) consists in deciding, for a given sentence \(\varphi\), wCGS \(G\) at a state \(v \in V\) and predicate \(P \subseteq \{-1, 1\}\), whether \([\varphi]^{G, \rho}(v) \in P\).

Now, we have all the ingredients to prove the following result.

**Theorem 2.** Assuming that functions in \(F\) can be computed in polynomial space, model checking NatSL\([F]\) with imperfect information, natural strategies with recall, and \(k\) as parameter of the problem is PSPACE-complete.

Proof. For the lower-bound we recall that LTL\([F]\) model checking is PSPACE-complete [7]. For the upper-bound, to verify that a given NatSL\([F]\) formula \(\varphi\) is satisfied over a wCGS \(G\) at a state \(v \in V\) under assignments \(\chi\) over uniform natural strategies with recall, we make use of a recursive function as is done in [26]. We start by showing that each recursive call only needs at most polynomial space. First, observe that each assignment \(\chi\) has a strategy \(s_0\) for each agent \(a \in Ag\).

We know that each strategy \(s_0\) that can be assigned to agent \(a\) is bounded, and we have that \(\text{comp}(s_0) \leq k\). Thus, each strategy can be stored in \(O(k \cdot |Act|)\) and, by consequence, any assignment can be stored in space \(O((|Ag| \cdot |\text{free}(\varphi)|) \cdot (k \cdot |Act|))\).

Now, we can analyse the recursive function. For the base case, \([p]^{G, \rho}(v)\) can be computed in constant space via the weight function.

For strategy quantification \(\exists s_0^k\cdot \varphi\), we need to compute \(\exists s_0^k\cdot \varphi\) over assignments \((\cdot, \cdot)\) to store the current strategy and the current maximum value computed.

This can be computed by a simple loop that increases \(i\), computes \([\varphi]^{G, \rho}(\pi_i)\) and \(\min\) of \([\varphi]^{G, \rho}(\pi_j)\) over each \(j\) and, by consequence, any assignment can be stored in space \(O((|Ag| \cdot |\text{free}(\varphi)|) \cdot (k \cdot |Act|))\).

7 CONCLUSION

In this work we have introduced Natural Strategy Logic with quantitative semantics and imperfect information (NatSL\([F]\)) for reasoning about strategic ability in auctions. NatSL\([F]\) provides a tool for mechanism design and offers a new perspective for formal verification and design of novel mechanisms and strategies. We demonstrated the usefulness of our approach by modelling and evaluating strategies for repeated keyword auctions.

In terms of technical results, we proved that the model checking problem for NatSL\([F]\) is PSPACE-complete, that is, no harder than model checking for the much less expressive language of quantitative LTL (LTL\([F]\)). We also showed that NatSL\([F]\) has incomparable distinguishing and expressive power to SL\([F]\). This means that the characterizations based on simple bounded strategies offer an inherently different view of auctions and mechanism design from characterizations using combinatorial strategies of arbitrary complexity. Amazingly, this aspect has never been studied for natural strategies, not even for the original proposal of NatATL [46].

We consider several directions for future work. First, a probabilistic version of Strategy Logic would allow handling mechanisms in stochastic settings with mixed strategies. Another direction is to investigate the use of strategies with recall for learning other players’ valuations based on their behaviour. Finally, the implementation of a model checker for NatSL\([F]\) will enable the empirical evaluation of auctions with natural strategies.

ACKNOWLEDGMENTS

This research has been supported by the ANR project AGAPE ANR-18-CE23-0013 and the FNR-CORE/NCBR-POLLUX project STV (POLLUX-VI/1/2019).
