Intelligent Information Systems VIII
Proceedings of the Workshop held in
Ustroń, Poland, June 14-18 , 1999

Modelling Artificial Intelligence on a Case of Bridge Card Play Bidding

Wojciech Jamroga

Division of Computer Science
University of Gdańsk
e-mail: `matwj@univ.gda.pl`

Abstract. In this paper a model is constructed for games with communication under incomplete information. The model, based on natural language processing techniques, is developed and tested on a case of Bridge card play bidding. The approach proposed here emphasizes the information processing nature of such games. Finally, some conclusions from implementation of a prototype bidding system and a new methodology for performance evaluation of such systems are presented.

Keywords: knowledge based systems, natural language processing, game solving, games with incomplete information, conversational agent, reasoning, performance evaluation, Horn clauses.

1 Introduction.

Bridge is a game that requests multiple skills and knowledge. The knowledge must include: game theory, decision theory, a bit of psychology, and also information and language processing — especially in the phase of bidding. This paper presents an attempt to model Bridge card play bidding through players' language activity, as a special case of dialogue.

This work owes a great debt to several papers and books. General model of language communication is based on concepts presented in *"Language as a Cognitive Process"* by Terry Winograd (1983). Semantic structure of a bid is modelled using the utterance functions scheme proposed by Roman Jakobson (1960) in *"Linguistics and Poetics"*. Many ideas are taken from a book by James Allen (1995) *"Natural Language Understanding"*, including questions connected with multilevel semantic representation with first order predicate calculus, speech act planning, and speech act theory (an alternative to Jakobson's function scheme) as well.

A state-of-the-art of research on Bridge implementation was presented by Ian Frank (1996) in *"Search and Planning under Incomplete Information. A Study using Bridge Card Play"*. Frank constructs also new bounds for computational complexity of declarer play in Bridge and — most importantly — investigates the concept of tactics (inspired by machine theorem proving techniques) as a tool for reducing the game search space. The result — a PROLOG system called FINESSE — is reported to do quite well in a declarer play of No Trump contracts.

2 Constructing Model for the Game.

Bridge is an intelligent game — together with more "noble" chess, checkers and go. Well, the other games seem to be even more complex and sophisticated than Bridge. The number of positions of the game tree is $2 * 10^{33}$ but in actual game (with cards already dealt) — at most $2.3 * 10^{13}$ while the complexity of chess, checkers and go is about 10^{50}, 10^{20} and 10^{170} respectively [estimation according to L.V. Allis, H.J. van der Herik, I.S. Herschberg - see Frank (1996), p. 64]. So why chess and checkers systems play for a World Championship actually while existing Bridge programs cannot reach even intelligent novice level?

The first reason is that Bridge is a game with incomplete information. A Bridge player sees only a part of his/her situation (one fourth of all the cards when he/she bids; a half during proper play phase) so he/she may identify a lot of possible variants of reality. Thus, the standard tree model, so successfully used for chess and checkers (nodes represent states of the play, with the root for beginning state of the play and leaves for end states; arcs represent single moves), doesn't suffice for Bridge in many ways. The player knows that his actual position is represented somewhere in one of the possible trees but he doesn't know where. Moreover, the trees should be assigned different weights because their probability may vary. Further, the weights can be calculated (if only) from previously accomplished moves — so the analysis of actual situation involves analysis of the branch beyond actual node of the tree (which never has place in games with perfect information). That shows the existence of information level in the game of Bridge: the player must extract and use information from the earlier part of the play.

Second, Bridge involves both cooperation and competition so the information flow in the game is not trivial: it includes positive information as well as a disinformation (bluffs etc.). Positive informing puts the game near the domain of language activities; disinformation demands for some elements of psychology.

2.1 General Model of the Game.

The considerations above imply such a model for the game of Bridge:

- actual state of a game (from a player's point of view) is described by a node of the game tree (branches of the tree represent all potential lines of bidding and play) and a set of possible, weighted *worlds* Ω (all the possible deals). According to decision theory, the player should maximize his/her outcome assuming that he/she plays against ideal enemy — i.e. choose the $decision_1$ for which the value:

$$\min_{decision_2} \{ \sum_{w \in \Omega} outcome_w(decision_1, decision_2) * weight(w) \}$$

  is maximal.
- there has to be an *information processing system* to prepare input data for the decision layer. A bid (in the bidding phase) or leading of a card (in the play phase) are proper means of information in the game. The system must involve both signal analysis to update the weights that determine player's knowledge of actual situation, and generating signals passing some information to the other players. Of course, an advanced model of Bridge should include psychological moves, especially in the generating unit.

The main aim of this experiment is to create a framework for the information flow and information processing in the Bridge bidding phase.

Complexity of the Game. The number of all the possible lines of bidding can be calculated as follows: let *Pos* be the set of "positive" Bridge bids i.e. all bids except *pass*, *double* and *redouble*. We have 7 levels of bidding and 5 possible colours so the size of *Pos* is: $7 * 5 = 35$. Every line of bidding — except (pass, pass, pass, pass) — may be unambiguously constructed as an ordered nonempty sequence of "positive" bids with such "nonpositive" sequences added: (A) at the beginning: () or (pass) or (pass, pass) etc. — 4 possibilities; (B) between every two of positive bids and after the last of them: (), (pass), (dbl) etc. — 21 possibilities; (C) at the end — always 1 possibility: (pass, pass, pass). Thus, the total number of lines of bidding is:

$$lbs = 1 + \sum_{k=1}^{35} 4 * 21^k * \frac{35!}{k!(35-k)!} \approx 3.86 * 10^{47}$$

On the other hand - for a given, fixed world $w \in \Omega$ - the number of lines of play ($lps(w)$) is bound between $13! \approx 6.2 * 10^9$ and $(13!)^4 \approx 1.5 * 10^{39}$, and the number of possible worlds (wds) is $\binom{39}{13} * \binom{26}{13} * \binom{13}{13} \approx 8.5 * 10^{16}$. Thus the complexity of the game ($lbs * \sum_{w \in \Omega} lps(w)$) should be placed between $\mathbf{2 * 10^{74}}$ and $\mathbf{4.9 * 10^{103}}$. (For detailed analysis of possible Bridge lines of play number - see Frank (1996), pp. 53-69. In fact, following an argument analogous to the one presented by Frank, we can prove that $2.29 * 10^{33}$ is a lower bound of number of possible lines of play when one hand is known.)

Interpretation. The model allows us to describe precisely the process of changing weights connected with every specific world. To achieve this, we assign a weight modifier — function $m_a : (\Omega \to [0,1]) \to (\Omega \to [0,1])$ — to every arc a, and determine a sequence of start weights. Then the modifier functions determine how each move in the game changes our knowledge of the actual situation. Unfortunately, it is rather difficult to take this advantage due to high complexity of the game tree. Assigning separate modifier values to each arc is completely impossible, even if we knew what values they should be.

However, it's the complexity of bidding phase subtree that is most surprising. One would expect that the bidding should be less complex than the proper play. In fact, any human player would ignore most of the $3.86 * 10^{47}$ lines of bidding as "nonsense" or "idiotic". On the other hand, we haven't any strict criterion for pruning those "idiotic" sequences without tracing them.

In this paper, the bidding is presented in a different perspective — as a special conversation of 4 players who exchange information about cards they posses and decisions they take. It reflects the way humans play Bridge: every bid is generated and analysed according to a language of signals, called *bidding system*. Of course, the processes demand for some analysis of actual situation, but as a result we usually need some approximation of the worlds' weigt function (for that purpose we will use the concept of prototype-extensional semantics). Similar worlds are grouped by logical constraints and "general weight modifiers" are encoded in information processing rules. The perpective gives also an opportunity to extract some extra linguistic information from players' moves. Information of that kind often allows to reduce the scope of search or even makes the search needless. Thus, searching through possible lines of bidding and play appears only when absolutely necessary — and it limits to the lines *a priori* accepted to be "sensible". Despite the disadvantages, such approach reduces computational complexity of the problem to accessible values.

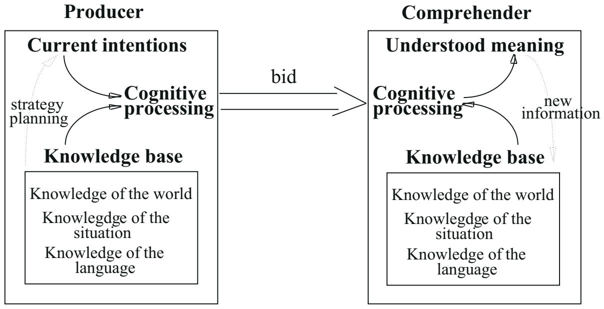
2.2 A New Perspective: Language.

Developing the perspective, we will use techniques of natural linguistics and natural language processing. In the domain of so-called "computational linguistics" language activity is described as a special process of communication between intelligent active agents. The agents are often modelled as processes with knowledge base. One of the agents, *producer* of an utterance, generates a message (in our case: a bid) that fits best his/her current goals as a result of knowledge-based cognitive processing. The other agent analyses the messages through cognitive processing to extract pure meaning of the message [see Winograd (1983)]. Figure 1 contains also two elements not included in original Winograd's model, i.e. *planning process* that prepares actual goals (in games often called a strategy or a decision) and process of *updating the knowledge* with the information from recently interpreted message.

The *message* is understandable if it refers to some *context*, shared by agents who use the same *code* (in our case: a bidding system) to create and maintain the *contact*. Each of these elements relates with one of the six utterance functions, proposed by Jakobson in (1960). Various speech acts differ in functions that may be recognized as more or less dominant.

Informative function refers to information about context (i.e. the knowledge of world or situation), contained by the utterance. *Expressive function* refers to producer's emotions and states expressed with the message. These aspects of message which may cause

Fig. 1. Model of cooperative communication.



the comprehender act, feel or think in a specific way, form the *impressive function. Fatic function* is centred around starting and maintaining communication ("Halo!"), *metalanguage function* refers to the language itself ("What is finesse?", "I don't understand what you say."), and *poetic function refers* to the form of the message: its style, rhythm, sound etc.

CONTEXT – *informative f.*
CONTACT – *fatic f.*
PRODUCER ⟶ COMPREHENDER
expressive f. *impressive f.*
CODE – *metalanguage f.*
MESSAGE – *poetic f.*

Jacobson's scheme structures potential interactions between an utterance and reality. The scheme may be used to describe the producer's intention as well as the meaning extracted from message by the comprehender. This provides us a homogenous semantic representation for all the processes involved in communication.

2.3 Bridge Bidding Model.

In this point we shall apply concepts presented above to Bridge bidding. The player actually bidding would be the producer, the other players — comprehenders; of course, the roles change as the bidding continues. Every moment the conversation consists of three dialogues: between the producer and each of the comprehenders respectively. It should be noted that for every player the most important is the dialogue with his/her partner. It means that, almost always, positive informing is much more important than disinformation.

Applying Jakobson's Scheme to Bridge Bids. In language communication the scheme may be useful for classifying both the producer's intention and meaning of the utterance itself (which the comprehender attempts to catch). The most dominant in bridge bidding are: informative and impressive function, and fatic function is also of some importance. Besides, we have to introduce "causal function", absent in the original scheme but necessary to describe *active* interaction between a bid and its context. The following examples use classical Culbertson's bidding system to show how the domination may vary:

1. *informative function* dominates bids such as **1**♡ opening, which *informs* us that the declarer has 12-18 Point Counts (PC) and at least 5 Hearts;
2. *impressive function* dominates every bid aimed to make declarer's partner act in a particular way: questions, suggestions etc. For instance, **4♣** in most systems should make the partner describe the number of Aces he posses in his card;
3. *fatic function* refers to the existence of contact. Bids, forcing declarer's partner not to let the actual bidding close (opening **2♣**, **2**♢, etc.), are good examples of its domination;
4. *causal function*. It is assumed in Jakobson's model that an utterance may refer to the context in a passive way only — i.e. the utterance can describe "objective reality", but cannot be used to shape it. Obviously, it's not true in the case of Bridge bidding, because every positive bid creates a new situation: when an opponent bode **3♠** none can bid below 3 No Trump level. So the opening **3♠** *causes* dramatic reduction of bidding space for all the players.

The four dimensions serve as a basis for representing semantic and intentional information in this experiment.

3 Constructing Detailed Solution.

Now it's time to specify the details of machine Bridge player's knowledge and information processing algorithms. This leads to an experimental rule-based bidding system in PROLOG programming language. The knowledge and specification of information-oriented processes, in the shape of Horn clauses set, form the core of the system. At the end of this chapter, a new method for evaluating performance of machine bidding programs will be proposed.

3.1 Knowledge of the Language.

Any Bridge player should know at least his/her own bidding language and the system actually used by the enemies (which may be a different one). All Bridge bidding languages prefer following phases of bidding: (1) bidding with Conventions, (2) "natural" bidding, (3) decision making phase. Bidding with Conventions is almost automatic; choosing bid for a given card and bidding history is often deterministic. Thus, it's easy to formalize that phase using rules:

if history conditions **and** card conditions **then** correct bid.

The bidding becomes more sophisticated when it comes to the "natural" phase. The rule here are a bit obscure and refers both to comprehender intuition and some higher-order notions:

if partner *invites* you to play 4♡ and you *think* that you have some *extra values* in your card **then** *show* them.

The rules are few here, but they are much more general. They refer not only to literal information about player's card, but to every part of the potential bid meaning.

Semantic Representation for a Bid. The meaning/intentional plan for a bid may be modelled as a frame, containing some slots to fill. According to previously presented concepts, the shape of the frame should be as follows:

informative function:	**impressive function:**	
fatic function	**causal function:**	*dominant function:* ....

Informative function is represented by a sequence of logical formulae (Horn clauses) defining cards of the declarer and/or his estimation of the actual situation. The clauses may refer to some predefined notions, such as *Point Counts*, *number of cards in a colour*, *extra values* etc. The other functions slots are filled with sequences of references to built-in notions like: *forcing*, *sign-off*, *invitation* etc. Note that:

- a player doesn't need to fill all the slots to generate his bid;
- some conventions are ambiguous in their informative, as well as the other layers, for instance opening **1♣** implies 12-18 PC with 5 Clubs or 12-15 PC without any 5-cards long colour. Sets of alternative variants are declared as: *variant(ConventionName, VariantName)*. Then we can describe a variant property:
  Property(Args...) :- variant(ConventionName, VariantName).

Fillers for the Slots. The rule base includes definitions of some notions, often used by human Bridge players:

- impressive function: *invitation(ToWhat), strong_invitation(ToWhat), discouragement, proposition(Colour), settle(Colour), convention(ConventionName)*;
- fatic function: *forcing, forced_bid, sign_off*;
- causal function: *bidding_space_occupation(BiddingLevel), settle_contract(Contract)*;
- informative function: *pc(PointCount), extra_values, no_extras, length(Suit, Length), shorter(Suit1, Suit2), extra_values(Suit), good(Suit), variant(VariantName), impossible(VariantName), pc_on_2_hands(PointCount), length_on_2_hands(Suit, Length), possible(Contract), impossible(Contract), situation(Estimation).*

Of course, this set is arbitrary and it doesn't suffice to describe all the bidding systems. Notions such as *invitation, forcing, settle_contract* etc. are defined procedurally, i.e. through algorithms describing how to interpret them. Elements of the informative notions set are defined literally.

Constraints, Defaults and Nonmonotonic Reasoning. Kronenfeld, Armstrong & Wilmoth in Kronenfeld et al. (1985) describe the idea of *prototype-extensional semantics*. Meaning is defined by the *prototype* (the most typical object for given word, clause or notion) and *extension rules* which enable deriving less typical representatives from more typical ones. This concept may be useful in representing and processing information in the game of Bridge. For instance, we can model implications of every bid on three levels:

- the prototype (for instance, opening **1♠** suggests 13 PC with exactly 5 Spades and no suit shorter than 2 cards),

- typical cases set, defined by plain logical constraints (for opening **1♠**: 12-18 PC, 5 or more Spades),
- non-typical, but acceptable cases, defined by derivation rules (opening **1♠**: the declarer may have a little less points, but with stronger Spade suit).

Note that although every bidding language is defined in notions of the second level, it is the prototype level that's often used by human players to reason about the next move. Why? It reduces complexity of the problem.

Reasoning on the prototype level recalls the *default logic* in a manner. A player accepts *by default* that the actual card set of the declarer fulfils prototype conditions. Of course, new information may abolish or change his beliefs. Human players often reason in nonmonotonic way. Say, our partner signalled 5 or more Diamonds and Heart suit shorter than Diamond one. By default we usually assume that number of Diamonds in his cards is exactly 5 — thus has no more than 4 Hearts. In the next bid he signals 5 Hearts... The default judgement is now inconsistent so it has to be changed to nearest accessible value — in this case, 6 Diamonds and 5-card long Heart suit.

3.2 Knowledge of the World & the Situation.

Player's *knowledge of the world* consists of rules defining: cards properties, scoring, correct bids etc. (which is simple) as well as his/her necessary skills: situation classifying and decision making methods. In most situations the classification may be "rough" — based, for instance, on estimation of PC on the player's and his/her partner's hands. Results of such classification are then input data for no less "rough" decision-making process (an example of such "rough" decision rule was presented in section 3.1). More detailed analysis appears to be necessary only in more sophisticated states of a play, especially when the player has to decide on the final contract.

One obvious way to achieve this is simulation of all the possible continuations of the actual play (assuming ingeniousity and omniscience of the opponents) to compute the expected value for every possible decision. Unfortunately, the simulation is impossible due to its computational complexity. This implies necessity for use of some special techniques, highly reducing complexity of the search. Frank proposes a concept of *tactics*, inspired by *automated theorem proving* methodology. A tactic defines a set of elementary game moves sequences what flattens the game tree immediately. Moreover, the set of tactics is arbitrary and rather small (finesse, cashing a master card etc.), which prunes a lot of possible lines of play, presumed to be "non-sensible". Of course, this leads to suboptimal results, but Frank reports that his tactic-based system FINESSE presents quite good performance when playing No Trump contracts.

Generalized (or recursive) tactics should be investigated in the future.

The *knowledge of the situation* is to record temporary information about actual game, reasoned from bids and visible cards. The main problem is that bids often carry information about general situation evaluation by the bidding player. Thus we have to *know what the player knows* about other players' cards to extract some useful information from his/her bid. For instance, our partner's bid suggests that he/she estimates the sum of our and his/her power to about 25-28 PC. We know he/she believes that we have 16-18

PC. So, the partner must have 7-12 PC. A smart way for representing such facts was presented by Allen (1995). Every fact of that kind can be asserted in a form *BEL(Who, What)* where *BEL* is called *belief* modal *operator*. For example, one of the facts above may be defined as follows:

BEL(me, BEL(partner, pc(me, between(16,18)))).

3.3 A New Method for Performance Evaluation.

There has been no satisfying method of evaluating performance of machine bidding systems. The method being preferred is: "let's take several human experts and make them rate the program for up to 100 points..." First, the method fails in its objectivity: a particular expert's rating depends on his evaluation of different lines of play importance, for which there is no certain applicable criterion. Moreover, even experts sometimes make mistakes. Second, the method ignores completely the connection between the bidding phase and the proper play phase: the bidding is only a unit which output serves as input data for another unit! Thus, there is no objectively optimal contract: the same contract would be optimal for a good Bridge player, too high for a weak one, and too low for a genius... That's very important, especially when we note that no Bridge program has proved to be even moderately good in the play.

The method being proposed uses the idea of the Bridge match. Two teams (each consisting of two pairs of players) take part in the match. If the pair A-B plays (for instance) in the first room on NS position, then their team mates C-D play in the second room on WE position. The same cards are played in both rooms so A has the same cards as V, C — the same as X etc. The result of the play (for the ABCD team) is defined as *outcome(AB) — outcome(CD)*. Thus, the result does not depend on the cards that was dealt, but on the difference between the ABCD and XYVZ team skills.

To evaluate a machine bidding system, we play a Bridge match in which one particular person (say, A) is coupled with the program — i.e. in half of the match A bids by himself/herself, and in the other half his bids are generated by the program. Of course, in the proper play phase only original players take part. The system performance evaluation may be calculated as **the average difference between ABCD total scoring in the rounds in which the program was bidding, and the team scoring in the rounds in what A was bidding himself**.

This method is similar to the original **Turing test** ("imitation game") in some sense. In both cases the machine is evaluated as a unit of a greater structure by comparison with an arbitrary model. In the proposed method the structure consists of bidding and playing unit, and human bidding is the model. To evaluate any bidding unit, we measure *how replacing the model unit for the actual unit affects the whole structure performance.*

4 Experiment.

The conceptions presented above were used in the implementation of prototype bidding program REDOUBLE. REDOUBLE performs non-competitive bidding (i.e. the enemies must pass all the time) in the cases when it or its human partner may open with

1♣/♢/♡/♠. The aim of the experiment was to show that such an "imitation" of human players, following some deep analysis of human player performance and the game itself, may lead to quite good results in games with cooperation and incomplete information. However, as Allen (1995) put it, the more sophisticated framework, the more time we need to create even a moderately performing program. It is not surprising then that REDOUBLE bids (at best) at a novice level.

An example of the program performance is presented below. REDOUBLE play the role of N, and S is a human player (to be more specific — the author of this paper). N's cards are: ♠9863 ♡J753 ♢A ♣Q974. S's cards are: ♠KJ74 ♡AQ106 ♢K3 ♣J86.

1st bid (N): pass
[Intention] *inform:*pc(max(11)). *impres:* discouragement. *fatic,causal:* nothing. *domin:* impres.
[Meaning] *inform:* pc(max(11)). *impres:* discouragement. *fatic,causal:* nothing. *domin:* impres.
Next bid (S): 1♣
[Meaning] *inform:* pc(min(18)) :- variant(opening(1♣),strongClub). pc(between(12,15)) :- variant(opening(1♣),weakNoTrump). no_longer :- variant(opening(1♣),weakNoTrump). pc(min(16)) :- variant(opening(1♣),clubs). length(♣,min(5)) :- variant(opening(1♣),clubs). length(♢,max(4)) :- variant(opening(1♣),clubs). length(♡,max(4)) :- variant(opening(1♣),clubs). length(♠,max(4)) :- variant(opening(1♣),clubs). *impres:* convention(opening(1♣)). *fatic, causal:* nothing. *dominant:* impres.
Next bid (N): 1♡
[Intention] *inform:* pc(between(7,7)). length(♠,between(4,4)). length(♡,between(4,4)). length(♢,between(1,1)). length(♣,between(4,4)). *impres, fatic, causal:* ?. *dominant:* inform.
[Meaning] *inform:* pc(min(6)). length(♡,min(4)). shorter_or_equal(♠,♡). *impres:* proposition(♡). *fatic:* forcing1. *causal:* nothing. *dominant:* inform.
Next bid (S): 2♡
[Meaning] *inform:* variant(opening(1♣),weakNoTrump). length(♡,min(4)). *impres:* settle(♡). *fatic, causal:* nothing. *dominant:* impres.
Next bid (N): pass
[Intention] *inform:* length(♠,between(4,4)). *impres, fatic, causal, dominant:* ?.
[Meaning] *inform:* nothing. *impres:* discouragement. *fatic:* nothing. *causal:* settle_contract(2♡). *dominant:* causal.

Rough comparision with several commercial Bridge playing programs produced the following output:

Program being compared	Better contracts by REDOUBLE	Identical contracts	Better contracts by the other program
Grand Slam	*11%*	*44%*	*44%*
Micro Bridge 8 and BridgeMate 3 (demos)	*50%*	*12%*	*38%*
BrigeBuff v.6.0 (demo)	*18%*	*64%*	*18%*

Anyway, the implementation led to some important observations:

– semantic structure based on Jakobson's scheme appeared to be suitable for models of games with language-like communication;

– two-step model of information processing (planning/generation, analysis/interpretation) proved very practical in the system design phase;
– Horn clauses language is rather inflexible and has too little expressive power for representing elementary facts in player's knowledge base. Thus, PROLOG inference engine may be insufficient in such an application. Further experiments should investigate First Order Predicate Calculus (or even Higher Order one) in that place;
– programming detailed information processing rules with Horn clauses results in horrible, completely illegible specifications. Construction of a "bidding system compiler" may be a good solution;
– two-level reasoning model (nonmonotonic reasoning on the default beliefs level, and classical one on the logical constraints level) proves adequate for reasoning under incomplete information and should be developed in the future.

5 Final Remarks.

In this paper a conception of a framework for modelling games with cooperation and competition under incomplete information was presented. In this approach a player is modelled as an independent, active agent that takes part in the communication process. The agent must "know" classical game theory and decision theory as well as reasoning of various kinds on various types of information sets. The thesis of this work is that such a model reflects better the real nature of something we call **intelligence**.

References

1. Allen James (1995), *Natural Language Understanding* (2nd ed.), The Benjamin/Cummings Publishing Company.
2. Frank Ian (1996), *Search and Planning under Incomplete Information. A Study using Bridge Card Play*, University of Edinburgh, Ph.D. Dissertation. Published in 1998 by Springer-Verlag.
3. Jakobson Roman (1960), Linguistic and Poetics, [in:] Sebeok T.A. (ed.), *Style in Language.*
4. Kofler Edward (1963), *Wstęp do teorii gier. Zarys popularny* [*An Introduction to Game Theory. Popular Approach*].
5. Kronenfeld David B., Armstrong James D., Wilmoth Stan (1985), Exploring the Internal Structure of Linguistic Categories: An Extensionist Semantic View, [in:] Dougherty J.W.D (ed.), *Directions in Cognitive Anthropology*, Urbana, University of Illinois Press, pp. 91-110.
6. Macieszczak Andrzej, Mikke Janusz (1980), *Bez impasu. Elementy logiki i psychologii w brydżu* [*No Finesse. Logic and Psychology in Bridge*], Młodzieżowa Agencja Wydawnicza 1987.
7. Malawski Marcin, Wieczorek Andrzej, Sosnowska Honorata (1997), *Konkurencja i kooperacja. Teoria gier w ekonomii i naukach społecznych* [*Competition and Cooperation. Game Theory in Economy and Social Science*], Wydawnictwo Naukowe PWN.
8. Turing Alan M. (1950), *Computing Machinery and Intelligence*, [in:] "Mind", pp. 433-460.
9. Winograd Terry (1983), *Language as a Cognitive Process. Syntax*, Addison-Wesley Publishing Company.
10. Winograd Terry, Flores Fernando (1986), *Understanding Computers And Cognition. A New Foundation for Design*, Addison-Wesley Publishing Company.