

# Towards Verification of Dialogue Protocols: A Mathematical Model

Magdalena Kacprzak<sup>1(✉)</sup>, Anna Sawicka<sup>2</sup>, and Andrzej Zbrzezny<sup>3</sup>

<sup>1</sup> Bialystok University of Technology, Bialystok, Poland  
m.kacprzak@pb.edu.pl

<sup>2</sup> Polish-Japanese Academy of Information Technology, Warsaw, Poland  
asawicka@pja.edu.pl

<sup>3</sup> Jan Długosz University, Czestochowa, Poland  
a.zbrzezny@ajd.czyst.pl

**Abstract.** Formal dialogue systems are an important trend in current research on the process of communication. They can be used as the schema of the dialogue conducted between artificial entities or as a simplified form of human dialogue with a machine or a human being with a man. In this work we introduce a mathematical model of dialogue, which is inspired by dialogue games. This model will be used as a semantic structure in verification of properties of dialogue protocols. For this purpose, the semantics of the dialogue games has been translated into interpreted systems that are commonly used in the model checking approach. The newly created model will be applied to develop methods and techniques for automated analysis of dialogues.

**Keywords:** Formal dialogue system · Protocol · Verification

## 1 Introduction

Fast progress in modern technology has resulted in the need for dialogues carried out between two machines or a machine and a man. The development of a protocol to be followed during a conversation between the machine and the man is complex and challenging. It requires natural language processing and analysis. However, it appears that in some applications, the dialogue can be executed according to very strict rules. It is then a little trivial, but still retains these elements and features that are necessary for the analysis. As examples of such limited dialogues we can consider dialogue games [4, 11, 23, 25, 27]. In this approach, a dialogue is treated as some kind of a game played between two parties. Rules of this game define principles for the exchange of messages between parties in order to meet some assumptions. For example, in Hamblin system [8, 14] these rules prevent making argumentative mistakes. In contrast, Lorenzen system [15, 17] is intended to validate formulas of some logic [12, 28]. Each game should have three basic categories of rules. The first one, called *locution rules*, defines a set of actions (speech acts) the player is allowed to use.

These actions express communication intentions of players. The second type of rules defines possible answers to legal moves. For example, if a player claims some sentence  $T$ , the opponent can agree with him by performing *concede  $T$*  or challenge  $T$  by performing *why  $T$* . These rules are called *structural rules*. The third group of rules applies to effects of actions. In dialogue games all actions relate to verbal expression: confirming, rejecting, questioning or arguing them. Therefore, only public declarations (commitments) of players are changed. As a result, a set containing publicly uttered statements is assigned to each player. The result of an action is a change in this set. These rules are called *effect rules*.

The dialogues, which we intend to analyze, are simple argumentation dialogues that human can use to communicate with the software agents. The purpose of these dialogues is to train and support psychologists who work with people with reduced cognitive skills. One of the main tasks of our research is to design a protocol for such communication. This protocol specifies the restrictions and rules which determine what players can say and when. One of the basic assumptions of our approach is that dialogue participants speak the same language. On the one hand, the protocol must meet specific requirements. On the other hand, designed, specified protocol has its own character and we need to verify what properties have a dialogue in accordance with this protocol. For this purpose, we use the method of model checking applied in verification of multi-agent systems (MAS). Main approaches in this field are based on combining bounded model checking (BMC) with symbolic verification using translations to either ordered binary decision diagrams (BDDs) [10] or propositional logic (SAT) [21]. Properties of MAS are expressed in logics which are combinations of the epistemic logic with either branching [24] or linear time temporal logic [9]. These logics are interpreted over interleaved interpreted systems (IIS) [16] or interpreted systems themselves [7]. IIS are systems in which only one action at a time is performed in a global transition. It is the ideal semantics also to interpret the properties of dialogue games.

In this work we introduce a mathematical, general model for dialogue systems. This model refers to the tradition of model checking. Therefore, the first step of the research is to translate locution, structural and effect rules into the concept of interpreted systems. This is not a trivial task, because we need to find a bridge between two different structures. The difficulty of the translation lies in the fact that we should postpone dialogue system rules to notions such as a protocol function, a transition relation, and an evolution function. The initial, much poorer version of this model was described in [13]. In this article, we define among others legal answer function and we formalize all rules of the protocol by means of this new function. We also introduce a concept of numbered and double numbered actions. Subsequently, we will offer new modal language to express properties and adjust the appropriate model checking techniques to verify them. The property, which is within the range of our interests is primarily the reachability, i.e. whether after a concrete dialogue it is possible to achieve a state that satisfies a condition  $\alpha$ . For example, if one of the participants can convince his opponent to a statement  $\varphi$ . Another interesting property to verify

is whether there is a dialogue compatible with the protocol, after which some condition  $\alpha$  is true. Such a condition is determined by the outcome rule and depends on the type of dialogue that is analyzed.

## 2 General Framework for Argumentation Dialogues

There are many different dialogue systems [4, 12, 18, 20, 26]. The general specification for argumentation dialogues, which we use in our research, is described in [22]. In accordance with this specification, dialogues are assumed to be for two parties arguing about a single dialogue *topic*  $T$  - the *proponent*  $P$  who defends  $T$  and the *opponent*  $O$  who challenges  $T$ . Both are equipped with a set of commitments that are understood as publicly incurred standpoints.

**Definition 1.** A dialogue system for argumentation is a pair  $(\mathcal{L}, \mathcal{D})$ , where  $\mathcal{L}$  is a logic for argumentation and  $\mathcal{D}$  is a dialogue system proper.

The elements of the above top-level definition are defined as follows.

**Definition 2.** A logic for argumentation  $\mathcal{L}$  is a tuple  $(L_t, R, \text{Args}, \rightarrow)$ , where  $L_t$  is a logical language called the topic language,  $R$  is a set of inference rules over  $L_t$ ,  $\text{Args}$  is a set of arguments, and  $\rightarrow$  is a binary relation of defeat defined on  $\text{Args}$ .

**Definition 3.** A dialogue system proper is a triple  $\mathcal{D} = (L_c, Pr, C)$  where  $L_c$  is a communication language (the set of locution rules),  $Pr$  is a protocol (the set of structural rules) for  $L_c$ , and  $C$  is a set of effect rules of locutions in  $L_c$ .

The protocol for  $L_c$  is defined in terms of the notion of a dialogue, which in turn is defined using the notion of a move. The set  $M$  of moves is defined as  $\mathbb{N} \times \{P, O\} \times L_c \times \mathbb{N}$ , where the four elements of a move  $m$  are denoted as follows:  $id(m)$  - the *identifier* of the move,  $pl(m)$  - the *player* of the move,  $s(m)$  - the *speech act* (locution) performed in  $m$ , and  $t(m)$  - the *target* of  $m$ . The set of dialogues, denoted by  $M^{\leq \infty}$ , is the set of all sequences  $m_1, \dots, m_i, \dots$  from  $M$  such that 1) for every  $i$ ,  $id(m_i) = i$  (the identifier of move  $m_i$  is  $i$ ); 2)  $t(m_1) = 0$  (the target of move  $m_1$  is assumed to be 0); and 3) for every  $i > 1$   $t(m_i) = j$  for some  $j < i$  (the target of move  $m_i$  is one of the preceding moves). The set of *finite dialogues*, denoted by  $M^{< \infty}$ , is the set of all finite sequences that satisfy these conditions. When  $d$  is a dialogue and  $m$  a move, then  $(d, m)$  will denote the continuation of  $d$  with  $m$ .

The key notion for the dialogue system is the protocol. A *protocol* on the set of moves  $M$  is a set  $Pr \subseteq M^{< \infty}$  satisfying the condition that whenever  $d$  is in  $Pr$ , all initial sequences beginning with  $d$  are also in  $Pr$ . A partial function  $\overline{Pr} : M^{< \infty} \rightarrow 2^M$  is derived from  $Pr$  as follows:  $\overline{Pr}(d) = \text{undefined}$  whenever  $d \notin Pr$ ; otherwise  $\overline{Pr}(d) = \{m : (d, m) \in Pr\}$ . The elements of the domain  $dom(\overline{Pr})$  are called the *legal finite dialogues*.  $\overline{Pr}(d)$  is the set of moves permitted after  $d$ . If  $d$  is a legal dialogue and  $\overline{Pr}(d) = \emptyset$ , then  $d$  is said to be a *terminated* dialogue.

Every utterance from  $L_c$  can influence participants' commitments. The results of utterances are determined by commitment rules, which are specified as a commitment function. A *commitment function* is a function:  $C : M^{<\infty} \times \{P, O\} \rightarrow 2^{L_t}$ , such that  $C(\emptyset, i) = \emptyset$  for  $i \in \{P, O\}$ .  $C(d, i)$ , for a participant  $i \in \{P, O\}$  and a dialogue  $d \in M^{<\infty}$ , denotes a player  $i$ 's commitments following the execution of  $d$ .

### 3 Interpreted System for Dialogue Game

In this section we define a new model for argumentation dialogue games as presented in Sect. 2. This model uses the concept of interpreted systems and Kripke structures. In our further research we will interpret in this model formulas of a modal logic adequate to express properties describing the dynamics of dialogue systems and properties that allow prediction of players' behavior. The obtained Kripke structure will be used to perform model checking for dialogue protocols.

The set of players of a dialogue game consists of two players:  $W$  and  $B$ ,  $Pl = \{W, B\}$ . To each player  $p \in Pl$ , we assign a set of actions  $Act_p$  and a set of possible local states  $L_p$ . By  $\bar{p}$  we denote the opponent of  $p$ .

Every action from  $Act_p$  can influence participant's commitments. We assume that the set  $Act_p$  contains also the special empty (null) action  $\varepsilon$ . Players' commitments are elements of topic language and are represented by *commitment sets*. Every action (except null action) is synonymous with locution expressed by specific player. The results of locutions are determined by evolution function and are specified afterwards. As we describe player's local state as a set of commitments,  $L_p$  denotes possible commitment sets of player  $p$ . Next,  $Act$  denotes the Cartesian product of the players actions, i.e.  $Act = Act_W \times Act_B$ . The global action  $a \in Act$  is a pair of actions  $a = (a_W, a_B)$ , where  $a_W \in Act_W$ ,  $a_B \in Act_B$  and at least one of these actions is the empty action. This means that players can not speak at the same time and player cannot reply to his own moves.

Also, we need to order performed global actions and indicate which actions correspond with which ones and therefore we define double-numbered global action set  $Num_2Act = \mathbb{N} \times \mathbb{N} \times Act$ . During the dialogue, we assign to each performed global action two numbers: the first one (ascending) indicates order (starting from the value 1). The second one points out to which earlier action this action is referring (0 at the beginning of the dialogue means that we are not referring to any move). Therefore, we define numbered global action set  $Num_1Act = \mathbb{N} \times Act$ . The element of this set has additional information about action it refers to. If we want to find out whether we can use some global action one more time, we should check if the possible move containing the same global action refer to the different earlier move. We define function  $Denum : Num_2Act \rightarrow Num_1Act$ , which maps double-numbered global action to the numbered global action. We understand dialogue  $d$  as a sequence of moves and in particular we denote  $d_{1..n} = d_1, \dots, d_n$ , where  $d_i \in Num_2Act$ ,  $d_i = (i, j, act)$ ,  $j \in \mathbb{N}$ ,  $j < i$ ,  $act \in Act$ .

A global state  $g$  is a triple consisting of dialogue and players' commitment sets corresponding to a snapshot of the system at a given time

$g = (d(g), C_W(g), C_B(g))$ ,  $g \in G$  where  $G$  is the set of global states. Given a global state  $g = (d(g), C_W(g), C_B(g))$ , we denote by  $d(g)$  a sequence of moves executed on a way to state  $g$  and by  $C_p(g)$  - the commitment set of player  $p$  (by  $C_{\bar{p}}(g)$  we denote the commitment set of the  $p$ 's opponent).

An *interpreted system* for a dialogue game is a tuple

$$IS = (I, \{L_p, Act_p\}_{p \in Pl})$$

where  $I \subseteq G$  is the set of initial global states. Let  $\alpha, \beta, \varphi, \psi_1, \dots, \psi_n, \gamma_1, \dots, \gamma_n \in Form(PV)$ , i.e., be formulas defined over the set  $PV$  which is a set of atomic propositions under which a content of speech acts is specified. Locutions used in players' actions are the same for both players:  $Act_W = Act_B = \{\varepsilon, claim\varphi, concede\varphi, why\varphi, \varphi\ since \{\psi_1, \dots, \psi_n\}, retract\varphi, question\varphi\}$ .

We define *legal answers function*  $F_{LA} : Num_2Act \rightarrow 2^{Num_1Act}$ , which maps double-numbered action to the set of possible numbered actions. This function is symmetrical for both players.

- $F_{LA}(i, j, (\varepsilon, \varepsilon)) = \emptyset$ .
- $F_{LA}(i, j, (claim\ \varphi, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, why\ \varphi), (\varepsilon, concede\ \varphi), (\varepsilon, claim\ \neg\varphi)\}\}$ ,
- $F_{LA}(i, j, (why\ \varphi, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, \varphi\ since\ \{\psi_1, \dots, \psi_n\}), (\varepsilon, retract\ \varphi)\}\}$ ,
- $F_{LA}(i, j, (\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, why\ \alpha), (\varepsilon, concede\ \beta), (\varepsilon, \neg\varphi\ since\ \{\gamma_1, \dots, \gamma_n\})\}\}$ , where  $\alpha \in \{\psi_1, \dots, \psi_n\}$  and  $\beta \in \{\varphi, \psi_1, \dots, \psi_n\}$ ,
- $F_{LA}(i, j, (concede\ \varphi, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, \varepsilon), (\varepsilon, claim\ \alpha), (\varepsilon, \alpha\ since\ \{\psi_1, \dots, \psi_n\})\}\}$ , for some  $\alpha, \psi_1, \dots, \psi_n \in Form(PV)$
- $F_{LA}(i, j, (retract\ \varphi, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, \varepsilon), (\varepsilon, claim\ \alpha), (\varepsilon, \alpha\ since\ \{\psi_1, \dots, \psi_n\})\}\}$ , for some  $\alpha, \psi_1, \dots, \psi_n \in Form(PV)$
- $F_{LA}(i, j, (question\ \varphi, \varepsilon)) = \{(i, act) : act \in \{(\varepsilon, retract\ \varphi), (\varepsilon, claim\ \varphi), (\varepsilon, claim\ \neg\varphi)\}\}$ ,

The actions executed by players are selected according to a *protocol function*  $Pr : G \rightarrow 2^{Num_2Act}$ , which maps a global state  $g$  to the set of possible double-numbered global actions. The function  $Pr$  satisfies the following rules.

- (R1)** For  $\iota \in I$   $Pr(\iota) = \{(1, 0, (claim\ \varphi, \varepsilon)), (1, 0, (question\ \varphi, \varepsilon)), (1, 0, (\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon))\}$
- (R2)**  $Pr((d_{1..k-1}, (k, l, (\varepsilon, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k, l, (\varepsilon, \varepsilon))\}$
- (R3)**  $Pr((d_{1..k-1}, (k, l, (claim\ \varphi, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k, l, (claim\ \varphi, \varepsilon))\}$
- (R4)**  $Pr((d_{1..k-1}, (k, l, (why\ \varphi, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k, l, (why\ \varphi, \varepsilon))\}$
- (R5)**  $Pr((d_{1..k-1}, (k, l, (\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k, l, (\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon))\}$

$$(R6) \ Pr((d_{1..k-1}, (k, l, (\text{concede } \varphi, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, \text{numact}) : \text{numact} \in ((\bigcup_{i <= k} F_{LA}(d_i) \cap \{(n, (\varepsilon, \alpha)) : n < k, \alpha \in Act_B\}) \setminus \{Denum(d_i) : i = 1, \dots, k\})\}$$

After opponent's locution *concede* the player can use one from possible answers for all previous opponent's moves, excluding these ones which he has already used.

$$(R7) \ Pr((d_{1..k-1}, (k, l, (\text{retract } \varphi, \varepsilon)), C_W(g), C_B(g))) = \{(k+1, \text{numact}) : \text{numact} \in ((\bigcup_{i <= k} F_{LA}(d_i) \cap \{(n, (\varepsilon, \alpha)) : n < k, \alpha \in Act_B\}) \setminus \{Denum(d_i) : i = 1, \dots, k\}) \cup \{(k+1, x, (\varepsilon, \text{why } \beta)) : \exists x < k \ d_x = (x, y, (\beta \text{ since } \varphi, \varepsilon))\} \text{ for some } \varphi, \beta \in Form(PV)\}$$

Again, after opponent's locution *retract* $\varphi$  the player can use one from possible answers for all previous opponent's moves, excluding these ones which he has already used but also he can ask for reason for  $\beta$  if  $\varphi$  was previously used to justify  $\beta$ .

These rules for player *B* are analogous.

Locution *claim* can start a dialogue or can be introduced during the dialogue, but only if it can lead to the conflict resolution and termination of discussion. Such a *claim* should refer to previous actions in the dialogue and do not start a completely new topic, which has no connection with previous ones. Therefore *claim* cannot be introduced after locutions, which demand immediate answer (locutions *why* and *since*). Also, after player's *claim*  $\alpha$ , the opponent can utter (among others) the only possible subsequent *claim* – *claim*  $\neg\alpha$  because the answer to the first locution should reveal opponent's attitude to player's statement.

To avoid infinite dialogues, at the beginning of the game we limit the number of possible introduced atomic sentences and the number of uttered locutions *claim*. In our system, we consider only finite dialogues, that is why we need the above rules preventing infinite ones. Similar rules are constructed e.g. for locution *since*, which can also start a new thread of the discussion.

Finally, we define *global* (partial) *evolution function*  $t : G \times Num_2Act \rightarrow G$ , which determines results of actions. This function is symmetrical for both players. Let  $d(g) = d(g)_{1, \dots, m}$ , then:

$$\begin{aligned} - \ t(g, (m+1, j, (\text{claim } \varphi, \varepsilon))) = g' \text{ iff } \varphi \notin C_W(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\} \\ \wedge d(g') = (d(g)_{1, \dots, m}, (m+1, j, (\text{claim } \varphi, \varepsilon))), \\ - \ t(g, (m+1, j, (\text{concede } \varphi, \varepsilon))) = g' \text{ iff } \varphi \in C_B(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\} \\ \wedge d(g') = (d(g)_{1, \dots, m}, (m+1, j, (\text{concede } \varphi, \varepsilon))), \\ - \ t(g, (m+1, j, (\text{why } \varphi, \varepsilon))) = g' \text{ iff } C_W(g') = C_W(g) \wedge d(g') = \\ (d(g)_{1, \dots, m}, (m+1, j, (\text{why } \varphi, \varepsilon))), \\ - \ t(g, (m+1, j, (\varphi \text{ since } \{\psi_1, \dots, \psi_n\}, \varepsilon))) = g' \text{ iff } C_W(g') = C_W(g) \cup \\ \{\varphi, \psi_1, \dots, \psi_n\} \wedge d(g') = (d(g)_{1, \dots, m}, (m+1, j, (\varphi \text{ since } \{\psi_1, \dots, \psi_n\}, \varepsilon))), \\ - \ t(g, (m+1, j, (\text{retract } \varphi, \varepsilon))) = g' \text{ iff } C_W(g') = C_W(g) \setminus \{\varphi\} \wedge d(g') = \\ (d(g)_{1, \dots, m}, (m+1, j, (\text{retract } \varphi, \varepsilon))), \end{aligned}$$

$$- t(g, (m + 1, j, (question \ \varphi, \varepsilon))) = g' \text{ iff } C_W(g') = C_W(g) \wedge d(g') = (d(g)_{1, \dots, m}, (m + 1, j, (question \ \varphi, \varepsilon))).$$

## 4 Example of Formal Dialogue

The following example shows the dialogue, which is compatible with the protocol as described in the previous section. This dialogue is between two persons Alice and Bob:

- [1] A: Taxes should not be raised.
- [2] B: I rather think that they should be raised.
- [3] A: Why do you think they should be raised?
- [4] B: I think so because state needs more money for health care and education and that is the only way to get them.
- [5] A: Why do you think that is the only way to get them?
- [6] B: Probably you are right, that is not the only way.
- [7] A: So why do you think taxes should be raised?
- [8] B: It seems that maybe they should not.
- [9] A: I think they shouldn't be raised because tax increase would kill economic growth and higher taxes mean jobs loss.
- [10] B: You convinced me, they should not be raised.

We introduce some abbreviations to construct the game based on the above dialogue:  $\alpha$  – “taxes should be raised”,  $\beta$  – “state needs more money for health care and education”,  $\theta$  – “raising taxes is the only way to get money”,  $\delta$  – “tax increase would kill economic growth”,  $\lambda$  – “higher taxes mean jobs loss”.

Corresponding dialogue looks then as follows:

- A: (1, 0, (*claim*  $\neg\alpha$ ,  $\varepsilon$ ))  
 B: (2, 1, ( $\varepsilon$ , *claim*  $\alpha$ ))  
 A: (3, 2, (*why*  $\alpha$ ,  $\varepsilon$ ))  
 B: (4, 3, ( $\varepsilon$ ,  $\alpha$  *since*  $\{\beta, \theta\}$ ))  
 A: (5, 4, (*why*  $\theta$ ,  $\varepsilon$ ))  
 B: (6, 5, ( $\varepsilon$ , *retract*  $\theta$ ))  
 A: (7, 4, (*why*  $\alpha$ ,  $\varepsilon$ ))  
 B: (8, 7, ( $\varepsilon$ , *retract*  $\alpha$ ))  
 A: (9, 4, ( $\neg\alpha$  *since*  $\{\delta, \lambda\}$ ,  $\varepsilon$ ))  
 B: (10, 9, ( $\varepsilon$ , *concede*  $\neg\alpha$ ))

Alice starts a dialogue by stating that taxes should not be raised (move 1) and Bob claims the opposite (that they should) in the next move. When asked for justification (move 3), he gives two premises (move 4). Alice asks for reasons for the second premise (move 5), but because Tom hasn't got anything to support his claim, he retracts from this premise in move 6. Because the justification for rising taxes presented in move 4 is not valid any more, Alice ask one more time for tax increase justification (move 7). It is noteworthy that this move does not refer to the immediately preceding move (like all previous moves), but contains reference

to the move 4, where the first justification for tax increase took place. Move 7 can be executed according to the rule **(R7)**. In this rule we are excluding from possible moves set these ones, which have been already executed. Even though the global action (*why*  $\alpha, \varepsilon$ ) used in move 7 was already used in move 3, these two are different numbered global actions, because they refer to the different earlier moves. Apparently, Bob has no more arguments for his standpoint that raising taxes is a good idea and he retracts from it in move 8. Then, according to rules **(R7)** and **(R5)**, Alice in move 9 can reply also to one of previous moves of Bob, and she decides to reply directly to move 4 and justify the opposite thesis (that we should not raise taxes) by giving two premises. (We could also interpret this reply as a reference to move 8.) Bob accepts these arguments and concedes that taxes should not be raised (move 10), what was originally proposed by Alice.

## 5 Notes on Verification of Dialogue Protocols

Mathematical model for argumentative dialogue games presented in Sect. 3 provides a basis for applying the methods of model checking to verify the correctness of dialogue protocols relative to the properties that the protocols should satisfy. Model checking [1, 5, 6, 19] is an automatic verifying technique for concurrent systems such as: digital systems, distributed systems, real time systems, multi-agent systems, communication protocols, cryptographic protocols, concurrent programs, dialogue systems, and many others. To be able to check automatically whether the system satisfies a given property, one must first create a *model* of the system, and then describe in a formal language both the created model and the property.

Therefore, we associate with the given interpreted system a *Kripke structure*, that is the basis for the application of model checking. A Kripke structure is defined as a tuple

$$M = (G, Act, T, I, AP, V)$$

consisting of a set of global states  $G$ , a set of actions  $Act$  (in our approach  $Num_2Act$ ), a set of initial states  $I \subseteq G$ , a transition relation  $T \subseteq G \times Act \times G$  such that  $T$  is left-total, a set of atomic propositions  $AP$ , and a valuation function  $V : G \rightarrow 2^{AP}$  that assigns to each state a set of atomic propositions that are assumed to be true at that state.

To formulate properties of dialogue protocols suitable propositional temporal logics are applied. The most commonly used are: linear temporal logic (LTL), computation temporal logic (CTL), a full branching time logic (CTL\*), the universal and existential fragments of these logics, and other logics which are their modifications and extensions. One of the most important practical problems in the model checking is the exponential growth of number of states of the Kripke structure. That is why, in future work we intend to focus on symbolic model checking of dialogue protocols. Symbolic model checking avoids building a state graph; instead, sets and relations are represented by Boolean formulae. One of the possible methods of symbolic model checking is bounded model checking (BMC) [2, 3]. It uses a reduction of the problem of truth of a temporal formula



in a Kripke structure to the problem of satisfiability of formulae of the classical propositional calculus. The reduction is achieved by a translation of the transition relation and a translation of a given property to formulae of classical propositional calculus.

The standard BMC algorithm, starting with  $k = 0$ , creates for a given Kripke structure  $M$  and a given formula  $\varphi$ , a propositional formula  $[M, \varphi]_k$ . Then the formula  $[M, \varphi]_k$  is converted to a satisfiability equivalent propositional formula in conjunctive normal form and forwarded to a SAT-solver. If the tested formula is unsatisfiable, then  $k$  is increased (usually by 1) and the process is repeated. The BMC algorithm terminates if either the formula  $[M, \varphi]_k$  turns out to be satisfiable for some  $k$ , or  $k$  becomes greater than a certain,  $M$ -dependent, threshold (e.g. the number of states of  $M$ ). Exceeding this threshold means that the formula  $\varphi$  is not true in the Kripke structure  $M$ . On the other hand, satisfiability of  $[M, \varphi]_k$ , for some  $k$  means that the formula  $\varphi$  is true in  $M$ .

## 6 Conclusions

Dialogue is the primary means of interpersonal communication. Structured dialogue represents a class of dialogue practices developed as a means of orienting the dialogic discourse toward problem understanding and consensual action. It can be used as a practice in e.g. education or business. In our research we analyze dialogues and communication processes. We decided to build on the tradition of formal dialogue systems to propose a concise model of a dialogue game. Natural dialogue may be simplified so as to keep only the essential elements. A simplified dialogue can be successfully used for scientific, educational and illustrative purposes. Its realization requires the elaboration of a protocol. A protocol is in simple terms a set of rules that guide the conversation and a powerful tool to help facilitate the structuring of conversation. Its form closely depends on what type of dialogue we want to carry out: persuasive, negotiating, argumentation, etc. For example, the aim of the dialogue may be to move towards consensus on difficult (disputed) issues. That is why there is a need to identify which protocol is best for a concrete application, what you can specify examining its properties. The aim of our research is dialogue protocol verification with the use of model checking. The first step was to establish a mathematical model, which will be a semantic structure for describing the properties of these protocols. Further work will focus on the development of adequate methods of verification.

**Acknowledgments.** Kacprzak's contribution was supported by the Bialystok University of Technology grant S/W/1/2014.

## References

1. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press, Cambridge (2008)
2. Biere, A., Cimatti, A., Clarke, E., Fujita, M., Zhu, Y.: Symbolic model checking using SAT procedures instead of BDDs. In: Proceedings of the ACM/IEEE Design Automation Conference (DAC 1999), pp. 317–320 (1999)
3. Biere, A., Cimatti, A., Clarke, E.M., Strichman, O., Zhu, Y.: Bounded model checking. *Adv. Comput.* **58**, 117–148 (2003)
4. Budzynska, K., Kacprzak, M., Sawicka, A., Yaskorska, O.: Dialogue Dynamics: Formal Approach, IFS PAS (2015)
5. Clarke, E.M., Emerson, E.A., Sistla, A.P.: Automatic verification of finite state concurrent systems using temporal logic specifications: a practical approach. In: Conference Record of the Tenth Annual ACM Symposium on Principles of Programming Languages, Austin, pp. 117–126. ACM Press (1983)
6. Clarke, E.M., Grumberg, O., Peled, D.: Model Checking. MIT Press, Cambridge (2001)
7. Fagin, R., Halpern, J.Y., Moses, Y., Vardi, M.: Reasoning About Knowledge. MIT Press, Cambridge (1995)
8. Hamblin, C.: Fallacies. Methuen, London (1970)
9. van der Hoek, W., Wooldridge, M.J.: Model checking knowledge and time. In: Bošnački, D., Leue, S. (eds.) SPIN 2002. LNCS, vol. 2318, pp. 95–111. Springer, Heidelberg (2002)
10. Jones, A.V., Lomuscio, A.: Distributed BDD-based BMC for the verification of multi-agent systems. In: Proceedings of of AAMAS 2010, pp. 675–682 (2010)
11. Kacprzak, M., Dziubinski, M., Budzynska, K.: Strategies in dialogues: a game-Theoretic approach. In: Computational Models of Argument, Frontiers in Artificial Intelligence and Applications, vol. 266, pp. 333–344 (2014)
12. Kacprzak, M., Sawicka, A.: Identification of formal fallacies in a natural dialogue. *Fundamenta Informaticae* **135**(4), 403–417 (2014)
13. Kacprzak, M., Sawicka, A., Zbrzezny, A.: Dialogue systems: modeling and prediction of their dynamics. In: Proceedings of AECIA 2015. Advances in Intelligent Systems and Computing, pp. 421–431 (2016)
14. Kacprzak, M., Yaskorska, O.: Dialogue protocols for formal fallacies. *Argumentation* **28**(3), 349–369 (2014)
15. Keiff, L.: Dialogical logic. In: The Stanford Encyclopedia of Philosophy (2011)
16. Lomuscio, A., Penczek, W., Qu, H.: Partial order reduction for model checking interleaved multi-agent systems. In: Proceedings of AAMAS 2010, pp. 659–666 (2010)
17. Lorenz, K., Lorenzen, P.: Dialogische logik. WBG, Darmstadt (1978)
18. Mackenzie, J.D.: Question begging in non-cumulative systems. *J. Philos. Logic* **8**, 117–133 (1979)
19. Meski, A., Penczek, W., Szreter, M., Wozna-Szczesniak, B., Zbrzezny, A.: BDD-versus SAT-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance. *Auton. Agent Multi-Agent Syst.* **28**, 558–604 (2014)
20. Parsons, S., Wooldridge, M., Amgoud, L.: Properties and complexity of some formal inter-agent dialogues. *J. Logic Comp.* **13**, 347–376 (2003)
21. Penczek, W., Lomuscio, A.: Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundamenta Informaticae* **55**(2), 167–185 (2003)

22. Prakken, H.: Coherence and flexibility in dialogue games for argumentation. *J. Logic Comput.* **15**, 1009–1040 (2005)
23. Rahman, S., Tuluheimo, T.: From games to dialogues and back: towards a general frame for validity. In: Majer, O., Pietarinen, A.-V., Tuluheimo, T. (eds.) *Games: Unifying Logic, Language, and Philosophy*, pp. 153–208. Springer, The Netherlands (2009)
24. Raimondi, F., Lomuscio, A.: Automatic verification of multi-agent systems by model checking via OBDDs. *J. Appl. Logic* **5**(2), 235–251 (2007)
25. Visser, J., Bex, F., Reed, C., Garssen, B.: Correspondence between the pragma-dialectical discussion model and the argument interchange format. *Stud. Logic Grammar Rhetoric* **23**(36), 189–224 (2011)
26. Walton, D.N., Krabbe, E.C.W.: *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of N.Y. Press, Albany (1995)
27. Wells, S., Reed, C.: A domain specific language for describing diverse systems of dialogue. *J. Appl. Logic* **10**(4), 309–329 (2012)
28. Yaskorska, O., Budzynska, K., Kacprzak, M.: Proving propositional tautologies in a natural dialogue. *Fundamenta Informaticae* **128**(1–2), 239–253 (2013)