Towards model checking argumentative dialogues with emotional reasoning

(Extended abstract)

Magdalena Kacprzak[1], Anna Sawicka[2], and Andrzej Zbrzezny[3]

[1] Bialystok University of Technology, Bialystok, Poland
[2] Polish-Japanese Academy of Information Technology, Warsaw, Poland
[3] Jan Długosz University, Czestochowa, Poland
`m.kacprzak@pb.edu.pl, asawicka@pja.edu.pl, a.zbrzezny@ajd.czest.pl`

Abstract. In the paper, we show a formal model for a dialogue game in which players can perform actions representing locutions like *claim*, *question*, *concede* as well as locutions which have a greater emotional charge like *scold* or *nod*. We define a protocol for dialogues in which participants have emotional skills and then give an interpreted system for them. Finally, we propose an extension of CTL logic with commitment, emotion and goal modalities. All of this is a formal basis, which we use to perform semantic verification of properties of dialogue systems with emotional reasoning.

1 Introduction

As members of society we have the need of collective work and protocols are an important part of our social skills. They help to facilitate structured conversations and are commonly used in everyday situations (sometimes unconsciously). Protocols should support dialogue, to help achieve the goal of the conversation. The first group of our interest are children who are known to have different cognitive abilities in comparison to adults. Usually, they are hard to convince and such a discourse is quite specific. For many people, it is hard to tell at first sight, from which argument we would benefit the most, and which one we should definitely avoid.

The aim of such an argumentation is a change in the emotional state of the interlocutor. The emotional state of our understanding consists of many factors, e.g. a sense of security, self-agency, self-satisfaction, self-confidence and so on. Some argument at the same time can increase one's sense of self-agency, but decrease one's sense of security ("if you find a job, you could move out, but you would have to rent your own flat"). We aim in designing an application which would be a support for people who have to convince somebody, but the more important factor is the emotional well-being of the interlocutor. We see such an application as a trainer of good practices in argumentation. We could consider possible reactions of potential interlocutor (e.g. a rebellious teenager, an expatriate) to specific arguments and monitor changes in the simplified representation of the emotional state. That is the reason we work on argumentative dialogue protocol, which is supposed to take into account change an emotional state of interlocutor in order to obtain the desired result (e.g. some kind of decision).

Usually, the aim of argumentation is figuring the agreement, the conviction of someone for their own reasons or even reaching a compromise [14, 32]. Persuasion dialogues are dialogues aimed at resolving conflicts of opinion between at least two participants. There are many types of such dialogues, e.g. *conflict resolution* dialogue begins with a conflict of opinion and ends when one of the participants convinces the other one. By the contrast, the argumentation under our consideration does not necessarily have to convince a child to do something, but it should help him become aware of his feelings. Certainly, we do not want to claim that there is an obvious argumentation that will convince everybody, but there are some argumentation strategies and mechanisms, which are quite known and considered as convincing ones.

Formal dialogue systems, which are growing field in the research on the process of communication, can be used as a schema for such dialogues, both between artificial agents or between the man and the machine. In our case, we need an argumentative dialogue model designed for human-computer communication, which applies the mentioned specific types of dialogues. There are other approaches which are focused rather on the agent to agent communication [4].

In this paper, we present a continuation of our work on the mathematical model of dialogue inspired by dialogue games [16]. We would like to use this model as a semantic structure in verification of properties of dialogue protocols and enable automated analysis of dialogues. The model we base our current research on is founded on the tradition of argumentative dialogue games by Prakken and others [25].

There are many approaches that assume very strict rules of communications. Our model, focusing on machine-man communication, is also based on such strict rules. On one hand, it makes it a little trivial, but on the other hand we can extract and focus on most important features of the dialogues. We can perceive *dialogue games* [7, 14, 26, 31, 33] as examples of such strict dialogues. In dialogue games, a dialogue is treated as some kind of a game played between two parties. Rules of such a game define policies for the communication between parties in order to meet some assumptions, for example in Hamblin system [12, 17] we have rules preventing argumentative mistakes, in Lorenzen system [18, 20] we have rules enabling validation of formulas [15, 34].

Each dialogue game should have three basic categories of rules. *Locution rules* define a set of actions (speech acts, locutions) the player is allowed to perform during the game. These actions express communication intentions of players. For example, rules of the dialogue game can assume that player can claim something, argue, justify, ask for justification, concede something etc. The second category of rules is responsible for the definition of possible answers for specific moves. For example, after one interlocutor claims something, the other one can concede it by performing *concede* or he can ask for justification by performing *why*. These rules are called *structural rules*. The third group of rules defines effects of actions. Due to performing some action (e.g. confirming or rejecting) a set of public declarations (commitments) of the interlocutor is changed. The result of an action is a change in the commitments set of the player, i.e. addition of some new statement to this set. These rules are called *effect rules*. We are specifying above rules which determine available moves for each player at every moment of the dialogue.

Even though every protocol must meet some general requirements, each one can be quite unique and we are interested in verifying characteristic properties of the dialogue defined by the specific protocol. In order to do that, we would like to use model checking method applied in verification of multi-agent systems (MAS). Main solutions in this matter combine bounded model checking (BMC) with symbolic verification using translations to either ordered binary decision diagrams (BDDs) [13] or propositional logic (SAT) [24]. Verified properties are expressed in logics which are combinations of the epistemic logic with branching [27] or linear time temporal logic [30]. Such logics can be interpreted either over interleaved interpreted systems (IIS) [19] or interpreted systems themselves [11]. To interpret the properties of dialogue games we chose IIS, in which only one action at a time is performed in a global transition.

The work presents a sketch of a formal system, which is a base for designing model checking techniques for verification dialogue games. We are concerned with argumentative dialogues, in which players can perform actions affecting commitments as well as their emotions. As a result, they change the emotional state, mood and attitude of the players. The proposed model will be used to show what mechanisms occur in human argumentative dialogues. In particular, we focus on argumentations where rational arguments are less effective (or not as effective) as the arguments referring to the emotions. On this basis, we will build a tool for learning managing emotions. Since emotions play a major role in persuading children, this tool can be used for personal development training for teachers or parents, which are often confused about children's feelings.

The study of emotions is part of various disciplines like Psychology, Economics, Cognitive Neuroscience, and, in recent years, also Artificial Intelligence and Computer Science. These studies aim to establish systems for emotional interaction. Nowadays, more and more artificial agents integrate emotional skills to achieve expressiveness, adaptability, and credibility. Such multiagent systems find application in the improvement of human-machine interaction, testing, refining and developing an emotional hypothesis or even the improvement of artificial intelligence techniques, once it optimizes decision-making mechanisms [28, 23].

2 Interpreted system

We start out by defining a mathematical model for argumentation dialogue games. This model uses the concept of interpreted systems and Kripke structures. In this model formulas of a modal logic adequate to express properties that allow prediction of players' behavior are interpreted. The obtained Kripke structure will be used to perform automatic verification of dialogue protocols via model checking techniques.

First, we assume that the set of players of a dialogue game consists of two players: White (W) and Black (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.

Every action from Act_p can influence participant's commitments. We assume that the set Act_p contains also the special empty (null) action ε. Every action (except null action) is synonymous with locution expressed by the specific player. Results of locutions are determined by *evolution function* and are specified afterwards.

Player's local state $l_p \in L_p$ consists of the player's *commitments*, *emotions*, and *goals*, $l_p = (C_p, E_p, GO_p)$. Player's commitments and goals are elements of a fixed topic language, which allows expressing the content of locutions. Thus, C_p and GO_p are sets of such expressions. These sets may be subject to change after a player's action. More specifically, the player can add or delete the selected expression. Emotions which we consider are fear, disgust, joy, sadness, and anger. Their strength (intensity) is represented by natural numbers from the set $\{1, 2, \ldots, 10\}$. Thus, E_p is a 5-tuple consisting of five values, which may also change after a certain action. It is worth highlighting here that a change in the intensity of the emotions is dependent on the type of locution and, perhaps even more, on its content.

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 cannot speak at the same time. Moreover, a player cannot reply to his own moves. Thus, the empty action is performed alternately by players W and B.

Also, we need to order performed global actions and indicate which actions correspond with which ones and therefore we define *double-numbered global actions* 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).

Furthermore, we define *numbered global actions* set $Num_1Act = \mathbb{N} \times Act$. Each element of this set is a pair (n, a) consisting of an action $a \in Act$ and the identifier of the action it refers to, $n \in \mathbb{N}$. 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 refers to the different earlier move. We define function $Denum : Num_2Act \to Num_1Act$, 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 $d_{1..n} = d_1, ..., d_n$, where $d_i \in Num_2Act$, $d_i = (i, j, a), j \in \mathbb{N}, j < i, a \in Act$.

A global state g is a triple consisting of dialogue history and players' local states corresponding to a snapshot of the system at a given time point $g = (d(g), l_W(g), l_B(g))$, $g \in G$ where G is the set of global states. Given a global state g, we denote by $d(g)$ a sequence of moves executed on a way to state g and by $l_p(g)$ - the local state of player p in g.

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 $\alpha, \beta, \varphi, \psi_1, .., \psi_n, \gamma_1, .., \gamma_n \in Form(PV)$, 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* φ, *concede* φ, *why* φ, *scold* φ, *nod* φ, φ *since* $\{\psi_1, \ldots, \psi_n\}$, *retract* φ, *question* $\varphi\ \}$.

In argumentation dialogues, a player can *claim* some facts, *concede* with the opponent or change his mind performing action *retract*. To challenge the opponent's statement, he may ask *why*, or ask whether the opponent commits to something, i.e., perform action *question*. For defense he can use the action *since*. It is the kind of reasoning and argumentation. Actions *scold* and *nod* express reprimand and approval, respectively. Note that all of these locutions refer to commitments, i.e., public announcements. We are not talking here about beliefs or knowledge, which may differ from the commitments.

Now we define *legal answer function* $F_{LA} : Num_2Act \to 2^{Num_1Act}$, which maps a double-numbered action to the set of possible numbered actions. This function is symmetrical for both players and determines for every action a set of legal actions which can be performed next.

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

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

(R1) For $\iota \in I\ Pr(\iota) = \{(1,0,(claim\ \varphi,\ \varepsilon)), (1,0,(question\ \varphi,\varepsilon)), (1,0,(\varphi\ since\ \{\psi_1,\ldots,\psi_n\},\varepsilon))\}$.

(R2) $Pr((d_{1..k-1},(k,l,(\varepsilon,\varepsilon)),l_W(g),l_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k,l,(\varepsilon,\varepsilon))$.

(R3) $Pr((d_{1..k-1},(k,l,(a,\varepsilon)),l_W(g),l_B(g))) = \{(k+1, numact) : numact \in F_{LA}(k,l,(a,\varepsilon))\}$, for $a \in \{\varepsilon,\ claim\ \varphi, scold\ \varphi,\ why\ \varphi,\ \varphi\ since\ \{\psi_1,\ldots,\psi_n\}\}$.

(R4) $Pr((d_{1..k-1},(k,l,(a,\varepsilon)),l_W(g),l_B(g))) = \{(k+1, numact) : numact \in ((\bigcup_{i<=k} F_{LA}(d_i) \cap \{(n,(\varepsilon,\alpha)) : n < k, \alpha \in Act_B\}) \setminus \{Denum(d_i) : i = 1,..,k\})\}$, for $a \in \{concede\ \varphi,\ nod\ \varphi,\ question\ \varphi\}$.
After opponent's locutions *concede*, *nod* or *question* the player can use one from possible answers for all previous opponent's moves, excluding these ones which he has already used.

(R5) $Pr((d_{1..k-1},(k,l,(retract\ \varphi,\varepsilon)),l_W(g),l_B(g))) = \{(k+1, numact) : numact \in ((\bigcup_{i<=k} F_{LA}(d_i) \cap \{(n,(\varepsilon,\alpha))\ :\ n < k, \alpha \in Act_B\}) \setminus \{Denum(d_i)\ :\ i = 1,..,k\})\} \cup \{(k+1,x,(\varepsilon, why\ \beta)) : \exists_{x<k}\ d_x = (x,y,(\beta \text{ since } \varphi,\varepsilon))\}$ for some $\varphi,\beta \in Form(PV)$.
After opponent's locution *retract* φ 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 the reason for β if φ was previously used to justify β.

These rules for player B are analogous.

The protocol is a crucial element of the model since it gives strict rules which determine the behaviour of players. In other words, it formally describes who, when and which action can perform. Rules (R1) and (R2) refer to the beginning and end of the dialogue, respectively. Rule (R3) states that after locutions *claim*, *scold*, *why*, and *since*, only actions determined by the legal answer function can be used. According to rules (R4) and (R5), actions *concede*, *nod* and *retract* end one of the threads of dialogue. Therefore, the next action can start a new thread or return to one of the unfinished. Actions *nod* and *scold* act similarly to actions *concede* and *claim*, but what distinguishes these actions is their emotional charge.

To show how locutions and their contents affect players' emotions and goals we define two functions. The first one determines the change of intensity of emotions: $EMOT_p : Act_w \times Emotion_p \to Emotion_p$ where $p \in Pl$ and $Emotion_p$ is a set of all possible 5-tuples for emotions, i.e., $Emotion_p = \{(n_1, \dots, n_5) : n_i \in \{1, \dots, 10\} \wedge i \in \{1, \dots, 5\}\}$. The second one determines the change of goals: $GOAL_p : Act_w \times Goal_p \to Goal_p$ where $p \in Pl$ and $Goal_p$ is a set of possible goals represented by expressions from the topic language, i.e. $Goal_p \subset Form(PV)$.

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

- $t(g,(m+1, j,(claim\ \varphi, \varepsilon))) = g'$ iff $\varphi \notin C_W(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\}$ $\wedge\, E_W(g') = EMOT_W(claim\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(claim\ \varphi, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(claim\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(concede\ \varphi, \varepsilon))) = g'$ iff $\varphi \in C_B(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\} \wedge E_W(g')$ $= EMOT_W(concede\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(concede\ \varphi, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(concede\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(why\ \varphi, \varepsilon))) = g'$ iff $C_W(g') = C_W(g)$ $\wedge\, E_W(g') = EMOT_W(why\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(why\ \varphi, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(why\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon))) = g'$ iff $C_W(g') = C_W(g) \cup \{\varphi, \psi_1, .., \psi_n\}$ $\wedge\, E_W(g') = EMOT_W(\varphi\ since\ \{\psi_1, \dots, \psi_n\}, E_W(g)) \wedge GO_W(g') =$ $GOAL_W(\varphi\ since\ \{\psi_1, \dots, \psi_n\}, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(\varphi\ since\ \{\psi_1, \dots, \psi_n\}, \varepsilon)))$,
- $t(g,(m+1, j,(retract\ \varphi, \varepsilon))) = g'$ iff $C_W(g') = C_W(g) \setminus \{\varphi\}$ $\wedge\, E_W(g') = EMOT_W(retract\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(retract\ \varphi, GO_W(g))$ $\wedge\, d(g') = (d(g)_{1,\dots,m}, (m+1, j,(retract\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(question\ \varphi, \varepsilon))) = g'$ iff $C_W(g') = C_W(g)$ $\wedge\, E_W(g') = EMOT_W(question\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(question\ \varphi, GO_W(g))$ $\wedge\, d(g') = (d(g)_{1,\dots,m}, (m+1, j,(question\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(scold\ \varphi, \varepsilon))) = g'$ iff $\varphi \notin C_W(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\}$ $\wedge\, E_W(g') = EMOT_W(scold\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(scold\ \varphi, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(claim\ \varphi, \varepsilon)))$,
- $t(g,(m+1, j,(nod\ \varphi, \varepsilon))) = g'$ iff $\varphi \in C_B(g) \wedge C_W(g') = C_W(g) \cup \{\varphi\}$ $\wedge\, E_W(g') = EMOT_W(nod\ \varphi, E_W(g)) \wedge GO_W(g') = GOAL_W(nod\ \varphi, GO_W(g)) \wedge$ $d(g') = (d(g)_{1,\dots,m}, (m+1, j,(concede\ \varphi, \varepsilon)))$,

Global evolution function defines results of actions. In particular, actions *claim*, *concede*, *scold*, *nod* and *since* add an expression to the commitments set while action *retract* deletes it. Actions *why* and *question* do not modify this set.

3 Kripke model and model checking

The mathematical model for argumentative dialogue games 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 [2, 8, 9, 22] 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.

The prerequisite inputs to model checking are a *model* of the system under consideration and a formal characterisation of the *property* to be checked. 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 $M = (G, Act, T, I)$ 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. Relation T is defined as follows $(g, a, g') \in T$ *iff* $g' \in t(g, a)$. By T^* we will denote the reflexive and transitive closure of T.

To formulate properties of dialogue protocols suitable propositional temporal logics are applied. The most commonly used, in general, 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 the 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) [5, 6, 1, 3, 29]. It uses a reduction of the problem of truth of a temporal formula in a Kripke structure to the problem of satisfiability. In SAT-based BMC the aforementioned reduction is achieved by a translation of the transition relation and a translation of a given property to formulae of classical propositional calculus, whereas in SMT-based BMC to quantifier-free first order formulae.

The standard BMC algorithm, starting with $k = 0$, creates for a given Kripke structure M and a given formula φ, a formula $[M, \varphi]_k$. Then the formula $[M, \varphi]_k$ is forwarded to either a SAT-solver or a SMT-solver. Note, that in the case of SAT-base BMC the propositional formula is converted to a satisfiability equivalent propositional formula in conjunctive normal form before forwarding it 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 φ is not true in the Kripke structure M. On the other hand, satisfiability of $[M, \varphi]_k$, for some k means that the formula φ is true in M.

4 Computation Tree Logic of Commitment and Action with Past

Interpreted systems are traditionally used to give a semantics to an epistemic language enriched with temporal connectives based on linear time [11]. Here we use CTL by Emerson and Clarke [10] as our basic temporal language and add commitment, emotion, goal, dynamic and past components to it. We call the resulting logic *Computation Tree Logic of Commitment and Action with Past*.

Definition 1 (Syntax). *Let $Pl = \{W, B\}$ be a set of players. The set of formulas is defined inductively as follows:*

- *$true$ is a formula,*
- *if $\varphi \in Form(PV)$ and $p \in Pl$ then $COM_p(\varphi)$ and $G_p(\varphi)$ are formulas,*
- *$E_p(e)$ is a formula for $p \in Pl$ and $e \in \{$fear, disgust, joy, sadness, anger$\}$,*
- *if α and β are formulas, then so are $\neg\alpha$, $\alpha \wedge \beta$ and $\alpha \vee \beta$,*
- *if $\bar{a} \in Act_W$ and α is a formula, then so are $\mathrm{AX}_{(W,\bar{a})}\alpha$ and $\mathrm{AY}_{(W,\bar{a})}\alpha$,*
- *if $\bar{a} \in Act_B$ and α is a formula, then so are $\mathrm{AX}_{(\bar{a},B)}\alpha$ and $\mathrm{AY}_{(\bar{a},B)}\alpha$,*
- *if α and β are formulas, then so are $\mathrm{AX}\alpha$, $\mathrm{AG}\alpha$ and $\mathrm{A}(\alpha \mathrm{U} \beta)$,*
- *if α is a formula, then so are $\mathrm{AY}\alpha$ and $\mathrm{AH}\alpha$.*

The remaining basic modalities are defined by derivation: $\mathrm{EF}\alpha \stackrel{def}{=} \neg\mathrm{AG}\neg\alpha$, $\mathrm{EP}\alpha \stackrel{def}{=} \neg\mathrm{AH}\neg\alpha$, $\mathrm{EZ}\alpha \stackrel{def}{=} \neg\mathrm{AZ}\neg\alpha$, $\mathrm{EZ}_{(W,\bar{a})}\alpha \stackrel{def}{=} \neg\mathrm{AZ}_{(W,\bar{a})}\neg\alpha$, $\mathrm{EZ}_{(\bar{a},B)}\alpha \stackrel{def}{=} \neg\mathrm{AZ}_{(\bar{a},B)}\neg\alpha$, for $\mathrm{Z} \in \{\mathrm{X}, \mathrm{Y}\}$, Moreover, $\alpha \Rightarrow \beta \stackrel{def}{=} \neg\alpha \vee \beta$, $\alpha \Leftrightarrow \beta \stackrel{def}{=} (\alpha \Rightarrow \beta) \wedge (\beta \Rightarrow \alpha)$, and $false \stackrel{def}{=} \neg\, true$.

The formula *true* is used for technical reasons and helps to express that some action is possible to execute, i.e., an action can lead to a state in which *true* holds. Of course, *true* is satisfied in every state.

Formula $COM_p(\varphi)$ describes the actual set of commitments of player p, more precisely, it expresses that φ is in this set. We should emphasize that φ is not a formula of the language defined herein, but a part of a separate structure in which it is possible to express the spoken sentences. In dialogue system, all actions are aimed at influencing the players' commitments. Therefore, the modality *COM* is very important and often used in the protocol specification. Modalities E_p and G_p allow for expressing properties concerning emotions and goals of player p.

The temporal modalities X, G stand for "at the next step", and "forever in the future", respectively. Y, H are their past counterparts "at the previous step", and "forever in the past". The modality A is the universal quantifier - "for all". Thus, AX means "for all next states" while AG means "for all states on all paths".

We also introduce modality $\mathrm{AX}_{(W,\bar{a})}$. It encodes an additional fact calling the action that led to the next state. Since we are talking about the implementation of a specific action, we must also indicate its executor. Hence, the subscript $(W, \bar{a})$, expressing that the performer is a player White, is added. A similar modality is defined for Black: $\mathrm{AX}_{(\bar{a},B)}$. As a result, the formula $\mathrm{AX}_{(\bar{a},B)}\alpha$ intuitively expresses that "at all next states reached after execution of action $\bar{a}$ by Black, α is true".

The operator U stands for *Until*; the formula $\alpha \mathrm{U} \beta$, expresses the fact that β eventually occurs and that α holds continuously until then.

As customary, the negation $\neg$A can be replaced by the existential quantifier E using the de Morgan's laws. So, $\neg \mathrm{AX}\alpha$ is equivalent with $\mathrm{EX}\neg\alpha$ - there exists a next state at which α holds. The interpretation of the other existential formulas is similar.

First, in order to give the semantics for the above formulas, we need to give a formal definition of a computation. A *computation* in a Kripke structure $M = (G, Act, T, I)$ is a possibly infinite sequence of states $\pi = (g_0, g_1, \ldots)$ such that there exists an action a_m for which $(g_m, a_m, g_{m+1}) \in T$ for each $m \in \mathbb{N}$, i.e., g_{m+1} is the result of applying the transition relation T to the global state g_m, and the action a_m.

Below we abstract from the transition relation, the actions, and the protocols, and simply use T, but it should be clear that this is uniquely determined by the interpreted system under consideration. In interpreted systems terminology, a computation is a part of a run. A k-computation is a computation of length k. For a computation $\pi = (g_0, g_1, \ldots)$, let $\pi(k) = g_k$, and $\pi_k = (g_0, \ldots, g_k)$, for each $k \in \mathbb{N}$. By $\Pi(g)$ we denote the set of all the infinite computations starting at g in M, whereas by $\Pi_k(g)$ the set of all the k-computations starting at g.

Definition 2 (Semantics – Interpretation). *Let M be a model (Kripke structure), $g \in G$ be a state, π be a computation, and α, β be formulas. $M, g \models \alpha$ denotes that α is true at the state g in the model M. M is omitted, if it is implicitly understood. The relation $\models$ is defined inductively as follows:*

$g \models true$ *for all* $g \in G$,
$g \models COM_p(\varphi)$ *iff* $\varphi \in C_p(g)$,
$g \models E_p(e)$ *iff* $n_i > 5$ *in* $E_p(g) = (n_1, .., n_5)$, *where e is fear, disgust, joy, sadness, anger and* $i = 1, 2, 3, 4, 5$, *respectively,*
$g \models G_p(\varphi)$ *iff* $\varphi \in GO_p(g)$,
$g \models \neg\alpha$ *iff* $g \not\models \alpha$,
$g \models \alpha \wedge \beta$ *iff* $g \models \alpha$ *and* $g \models \beta$,
$g \models \mathrm{AX}_{(W,\bar{a})}\alpha$ *iff* $\forall a = (i, j, (\bar{a}, \varepsilon)) \in Num_2Act$ *and* $\forall g' \in G$ (*if* $(g, \bar{a}, g') \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AX}_{(\bar{a},B)}\alpha$ *iff* $\forall a = (i, j, (\varepsilon, \bar{a})) \in Num_2Act$ *and* $\forall g' \in G$ (*if* $(g, \bar{a}, g') \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AX}\alpha$ *iff* $\forall g' \in G\, \forall a \in Num_2Act$ (*if* $(g, a, g') \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AG}\alpha$ *iff* $\forall \pi \in \Pi(g)\ (\forall_{m \geq 0}\, \pi(m) \models \alpha)$,
$g \models \mathrm{A}(\alpha \mathrm{U} \beta)$ *iff* $\forall \pi \in \Pi(g)\ (\exists_{m \geq 0}\, [\pi(m) \models \beta$ *and* $\forall_{j<m}\, \pi(j) \models \alpha])$,
$g \models \mathrm{AY}_{(W,\bar{a})}\alpha$ *iff* $\forall a = (i, j, (\bar{a}, \varepsilon)) \in Num_2Act$ *and* $\forall g' \in G$ (*if* $(g', a, g) \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AY}_{(\bar{a},B)}\alpha$ *iff* $\forall a = (i, j, (\varepsilon, \bar{a})) \in Num_2Act$ *and* $\forall g' \in G$ (*if* $(g', a, g) \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AY}\alpha$ *iff* $\forall g' \in G\, \forall a \in Num_2Act$ (*if* $(g', a, g) \in T$, *then* $g' \models \alpha$),
$g \models \mathrm{AH}\alpha$ *iff* $\forall g' \in G$ (*if* $(g', g) \in T^*$, $g' \models \alpha$).

The description of the semantics is finished by giving the definition of the validity in the model.

Definition 3. (Validity) *A formula φ is valid in M (denoted $M \models \varphi$) iff $M, \iota \models \varphi$, i.e., φ is true at the initial state of the model M.*

5 Properties of dialogue protocols

The formal language introduced in the previous section is used for giving the specification for dialogue protocols as well as for describing properties of these protocols. The properties can be divided into several classes [21]. Some of them are studied below.

Safety. Safety property usually expresses that something bad does not happen. However, it can also express that something good is always true. The best illustration here is the specification of locutions used in dialogues:

$$\mathrm{AG}(\mathrm{AX}_{(W, claim\ \alpha)}\, COM_W\, \alpha).$$

This formula states that after locution *claim* α, the formula α is in the set of commitments of the performer.

The next formula expresses a similar property, i.e., before the execution of the locution *retract* α, the formula α must be in the commitments set of the player:

$$\mathrm{AG}(\mathrm{AY}_{(W, retract\ \alpha)}\, COM_W\, \alpha).$$

Nontermination. One of the most important safety properties is nontermination. It expresses that every legal dialogue, i.e., dialogue in accordance with rules of a dialogue game does not have a termination state:

$$\mathrm{AG}(\mathrm{EX}true).$$

This formula states that in every state of every computation there is an action which can be performed and after execution of this action a formula *true* is satisfied. As a consequence, every dialogue is infinite.

Guarantee. One of the guarantee properties, i.e., properties that ensure that some event eventually happens, is *termination*. In dialogue systems, we often assume that the end of a dialogue means the fulfillment of a certain condition. This condition may express that one of the players, e.g. W, is happy:

$$\mathrm{E}(true\ \mathrm{U}\ E_W(joy)).$$

If any dialogue should end with the *termination* condition and this condition means that White does not feel fear, then we can express this fact as follows:

$$\mathrm{A}(true\ \mathrm{U}\ \neg E_W(fear)).$$

The formula claims that every computation contains a state at which the required condition holds.

Response. The response property expresses the fact that a property β is a guaranteed response to a condition α. An example of this is the formula

$$\mathrm{AG}(COM_p(\alpha) \Rightarrow \mathrm{E}(true\ \mathrm{U}\ \neg\, COM_p(\alpha)))$$

which states that if a player is committed to α, then during the dialogue he can change it. This property is very important since it states that it is possible to reject some commitment and at the same time it means the ability to change some opinion, what is crucial for argumentative dialogues. It makes no sense to provide and analyze arguments if the change of players' commitments is not possible at all.

6 Conclusion

The aim of our research is to design and implement a framework to provide a communication between a user and a machine which allows to better understand emotions that appear during human dialogues. We plan to create a tool that will support the personal development in this matter, i.e., the acquisition of skills of identifying and naming emotions. This is particularly important for training teachers, educators, psychologists, and parents. This process can take place between a human, which plays a role of a student, and a software agent, which plays a role of a teacher. The challenge is to design a suitable interface for such communication. However, the implementation should be preceded by constructing a mathematical model and proposing a new dialogue protocol. In our work, we also propose formal language for protocol specification and expressing its properties. On this basis, we plan to design and implement a multimedia tool for educational purposes. Psychological aspects of the project are consulted with a group of psychologists. Our research does not deal with linguistic analysis, but we want to explore dialogues with the fixed base so that the user can learn to recognize these places and elements of dialogue which relate to emotions.

Acknowledgment. The research by Kacprzak have been carried out within the framework of the work S/W/1/2014 and funded by Ministry of Science and Higher Education.

References

1. A. Armando, J. Mantovani, and L. Platania. Bounded model checking of software using SMT solvers instead of SAT solvers. *STTT*, 11(1):69–83, 2009.
2. C. Baier and J.-P. Katoen. *Principles of Model Checking*. MIT Press, 2008.
3. C. Barrett, R. Sebastiani, S. Seshia, and C. Tinelli. Satisfiability modulo theories. In A. Biere, M. J. H. Heule, H. van Maaren, and T. Walsh, editors, *Handbook of Satisfiability*, vol. 185 of *Frontiers in Artificial Intelligence and Applications*, chapter 26, pp. 825–885, 2009.
4. J. Bentahar, J.-J. C. Meyer, and W. Wan. *Specification and Verification of Multi-agent Systems*, chapter Model Checking Agent Communication, pages 67–102. Springer US, 2010.
5. A. Biere, A. Cimatti, E. M. Clarke, M. Fujita, and Y. Zhu. Symbolic model checking using SAT procedures instead of BDDs. In *Proc. of DAC'99*, pages 317–320, 1999.
6. A. Biere, A. Cimatti, E. M. Clarke, O. Strichman, and Y. Zhu. Bounded model checking. *Advances in Computers*, 58:117–148, 2003.
7. K. Budzynska, M. Kacprzak, A. Sawicka, and O. Yaskorska. *Dialogue Dynamics: Formal Approach*. IFS PAS, 2015.
8. E. M. Clarke, E. A. Emerson, and A. P. Sistla. Automatic verification of finite state concurrent systems using temporal logic specifications: A practical approach. In J. R. Wright, L. Landweber, A. J. Demers, and T. Teitelbaum, editors, *Conf. Rec. of the Tenth Annual ACM Symposium on Principles of Programming Languages,* pages 117–126, ACM Press, 1983.
9. E. M. Clarke, O. Grumberg, and D. A. Peled. *Model checking*. MIT Press, 2001.
10. E. A. Emerson and E. Clarke. Using branching-time temporal logic to synthesize synchronization skeletons. *Science of Computer Programming*, 2(3):241–266, 1982.
11. R. Fagin, J.Halpern, Y. Moses, and M.Vardi. *Reasoning about Knowledge*. MIT Press, 1995.
12. C. Hamblin. *Fallacies*. Methuen, London, 1970.

13. A. V. Jones and A. Lomuscio. Distributed BDD-based BMC for the verification of multi-agent systems. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, editors, *Proc. of AAMAS, Volume 1-3*, pages 675–682, 2010.
14. M. Kacprzak, M. Dziubinski, and K. Budzynska. Strategies in dialogues: A game-theoretic approach. In S. Parsons, N. Oren, C. Reed, and F. Cerutti, editors, *Proc. of COMMA*, volume 266 of *Frontiers in Artificial Intelligence and Applications*, pages 333–344. IOS Press, 2014.
15. M. Kacprzak and A. Sawicka. Identification of formal fallacies in a natural dialogue. *Fundam. Inform.*, 135(4):403–417, 2014.
16. M. Kacprzak, A. Sawicka, and A. Zbrzezny. Dialogue systems: Modeling and prediction of their dynamics. In A. Abraham, K. Wegrzyn-Wolska, E. A. Hassanien, V. Snasel, and M. A. Alimi, editors, *Proc. of AECIA*, pages 421–431. Springer International Publishing, 2016.
17. M. Kacprzak and O. Yaskorska. Dialogue protocols for formal fallacies. *Argumentation*, 28(3):349–369, 2014.
18. L. Keiff. Dialogical logic. In E. N. Zalta, editor, *The Stanford Enc. of Philosophy*. 2011.
19. A. Lomuscio, W. Penczek, and H. Qu. Partial order reductions for model checking temporal epistemic logics over interleaved multi-agent systems. In W. van der Hoek, G. A. Kaminka, Y. Lespérance, M. Luck, and S. Sen, editors, *IFAAMAS, Volume 1-3*, pages 659–666, 2010.
20. K. Lorenz and P. Lorenzen. Dialogische logik. *WBG. Darmstadt*, 1978.
21. Z. Manna and A. Pnueli. *The temporal logic of reactive and concurrent systems - specification*. Springer, 1992.
22. A. Meski, W. Penczek, M. Szreter, B. Wozna-Szczesniak, and A. Zbrzezny. BDD-versus SAT-based bounded model checking for the existential fragment of linear temporal logic with knowledge: algorithms and their performance. *AAMAS*, 28(4):558–604, 2014.
23. F. S. Nawwab, P. E. Dunne, and T. Bench-Capon. Exploring the role of emotions in rational decision making. In *Proc. of COMMA*, 2010.
24. W. Penczek and A. Lomuscio. Verifying epistemic properties of multi-agent systems via bounded model checking. *Fundam. Inform.*, 55(2):167–185, 2003.
25. H. Prakken. Models of persuasion dialogue. In *Argumentation in AI*, pages 281–300. Springer, 2009.
26. S. Rahman and T. Tulenheimo. From games to dialogues and back: towards a general frame for validity. In O. Majer, A. Pietarinen, and T. Tulenheimo, editors, *Games: Unifying Logic, Language, and Philosophy*, volume 15 of *LEUS*, pages 153–208. Dordrecht: Springer, 2006.
27. F. Raimondi and A. Lomuscio. Automatic verification of multi-agent systems by model checking via ordered binary decision diagrams. *Journal of Applied Logic*, 5(2):235 – 251, 2007. Logic-Based Agent Verification.
28. R. Silveira, G. K. da Silva Bitencourt, T. . Gelaim, J. Marchi, and F. de la Prieta. Towards a model of open and reliable cognitive multiagent systems: Dealing with trust and emotions. *ADCAIJ: Advances in Distributed Computing and Artificial Intelligence Journal*, 4(3), 2016.
29. C. Tinelli. SMT-based model checking. In *NASA Formal Methods - 4th International Symposium, NFM 2012, Norfolk, VA, USA, April 3-5, 2012. Proceedings*, page 1, 2012.
30. W. van der Hoek and M. Wooldridge. Model checking knowledge and time. In D. Bosnacki and S. Leue, editors, *Proc. of SPIN*, volume 2318 of *LNCS*, pages 95–111, 2002.
31. J. Visser, F. Bex, C. Reed, and B. Garssen. Correspondence between the pragma-dialectical discussion model and the argument interchange format. *Studies in Logic, Grammar and Rhetoric*, 23(36):189–224, 2011.
32. D. N. Walton and E. C. W. Krabbe. *Commitment in Dialogue: Basic Concepts of Interpersonal Reasoning*. State University of N.Y. Press, 1995.
33. S. Wells and C. A. Reed. A domain specific language for describing diverse systems of dialogue. *J. Applied Logic*, 10(4):309–329, 2012.
34. O. Yaskorska, K. Budzynska, and M. Kacprzak. Proving propositional tautologies in a natural dialogue. *Fundam. Inform.*, 128(1-2):239–253, 2013.