Concepts of Optimal Utterance in Dialogue: Selection and Complexity

Paul E. Dunne and Peter McBurney

Dept. of Computer Science
University of Liverpool
Liverpool L69 7ZF, UK
{ped,p.j.mcburney}@csc.liv.ac.uk

Abstract. Dialogue protocols have been the subject of considerable attention with respect to their potential applications in multiagent system environments. Formalisations of such protocols define classes of dialogue *locutions*, concepts of a dialogue *state*, and *rules* under which a dialogue proceeds. One important consideration in implementing a protocol concerns the criteria an agent should apply in choosing which utterance will constitute its next contribution: ideally, an agent should select a locution that (by some measure) *optimises* the outcome. The precise interpretation of optimise may vary greatly depending on the nature and intent of a dialogue area. One option is to choose the locution that results in a *minimal length* debate. We present a formal setting for considering the problem of deciding if a particular utterance is optimal in this sense and show that this decision problem is *both* NP–hard *and* CO-NP–hard.

KEYWORDS: Agent Communication Languages, Argumentation and Persuasion, Computational Complexity, Dialogue Protocols, Locution Selection.

1 Introduction

Methods for modeling discussion and dialogue processes have proved to be of great importance in describing multiagent interactions. The study of *dialogue protocols* ranges from perspectives such as argumentation theory, e.g., [26, 30], taxonomies of types of dialogue such as [30, 33], and formalisms for describing and reasoning about protocols, e.g. [17, 23, 25]. Among the many applications that have been considered are bargaining and negotiation processes, e.g. [18, 26, 21]; legal reasoning, e.g. [15, 20, 2, 3, 29], persuasion in argumentation and other systems, e.g. [32, 12, 4, 9], and inquiry and information-discovery, e.g. [22, 24]. The collections of articles presented in [8, 16] give an overview of various perspectives relating to multiagent discourse.

While we present a general formal model for dialogue protocols below, informally we may view the core elements of such as comprising a description of the 'locution types' for the protocol ("what participants can *say*"); the topics of discussion ("what participants talk *about*"); and how discussions may start, evolve, and finish.

Despite the divers demands of protocols imposing special considerations of interest with particular applications, there are some properties that might be considered desirable irrespective of the protocol's specific domain, cf. [25]. Among such properties are *termination*; the capability to *validate* that a discussion is being conducted according to the protocol; and the ability for participants to determine "sensible" contributions. In [17] frameworks for uniform comparison of protocols are proposed that are defined independently of the application domain. In principle, if two distinct protocols can be shown 'equivalent' in the senses defined by [17], then termination and other properties need only be proved for one of them.

In this paper our concern is with the following problem: in realising a particular discussion protocol within a multiagent environment, one problem that must be addressed by each participant can, informally, be phrased as "*what do/should I say next*?" In other words, each agent must "be aware of" its permitted (under the protocol rules) *utterances* given the progress of the discussion so far, and following specific criteria, either choose to say nothing or contribute one of these. While the extent to which a protocol admits a 'reasonable' decision-making process is, of course, a property that is of domain-independent interest, one crucial feature distinguishing different types of discussion protocol is the criteria that apply when an agent makes its choice. More precisely, in making a contribution an agent may be seen as "optimising" the outcome. A clear distinction between protocol applications is that the sense of "optimality" in one protocol may be quite different from "optimality" in another. For example, in multiagent bidding and bargaining protocols, a widely-used concept of "optimal utterance" is based on the view that any utterance has the force of affording a particular "utility value" to the agent invoking it that may affect the utility enjoyed by other agents. In such settings, the *policy* (often modeled as a probability distribution) is "optimal" if no agent can improve its (expected) utility by unilaterally deviating. This – Nash equilibrium – has been the subject of intensive research and there is strong evidence of its computational intractability [5]. While valid as a criterion for utterances in multiagent bargaining protocols, such a model of "optimality" is less well-suited to fields such as persuasion, information-gathering, etc. We may treat a "persuasion protocol" as one in which an agent seeks to convince others of the validity of a given proposition, and interpreting such persuasion protocols as proof mechanisms – a view used in, among others, [32, 12] – we contend that a more appropriate sense of an utterance being "optimal", is that it *allows* the discussion to be *concluded* "as quickly as possible".[1] There are several reasons why such a measure is appropriate with respect to persuasion protocols. In practice, discussions in which one agent attempts to persuade another to carry out some action cannot (reasonably) be allowed to continue indefinitely; an agent may be unable to continue with other tasks which are time-constrained in some sense until other agents in the system have been persuaded through some reasoned discussion to accept particular propositions. It is, of course, the case that describing optimality in terms of length of discussion provides only one measure. We discuss alternative notions of optimality in the concluding sections.

Concentrating on persuasion protocols we formulate the "optimal utterance problem" and establish lower bounds on its complexity. In the next section we outline an abstract computational framework for dialogue protocols and introduce two variants of the optimal utterance decision problem. In Section 3 we present a setting in which this

[1] An alternative view is proposed in [11], where it is argued that utterances which *prolong* discussions can, in certain settings, be seen as "optimal".

problem is proved to be both NP–hard *and* CO-NP–hard. Conclusions and further work are presented in the final section.

2 Definitions

Definition 1 *Let $\mathcal{F}$ be the (infinite) set of all well-formed formulae (wff) in some propositional language (where we assume an enumerable set of propositional variables $x_1, x_2, \ldots$).*

A dialogue arena*, denoted $\mathcal{A}$, is a (typically infinite) set of* finite *subsets of $\mathcal{F}$. For a dialogue arena,*

$$\mathcal{A} = \{\Phi_1, \Phi_2, \ldots, \Phi_k, \ldots, \} \quad \Phi_i \subset \mathcal{F}$$

the set of wff in $\Phi_i = \{\psi_1, \psi_2, \ldots, \psi_q\}$ is called a dialogue context *from the dialogue arena $\mathcal{A}$.*

Definition 2 *A* dialogue schema *is a triple $\langle \mathcal{L}, \mathcal{D}, \Phi \rangle$, where $\mathcal{L} = \{L_j | 1 \leq j \leq l\}$ is a finite set of* locution types*, $\mathcal{D}$ is a* dialogue protocol *as defined below, and Φ is a dialogue context.*

We are interested in reasoning about properties of protocols operating in given dialogue arenas. In the following, $\mathcal{A} = \{\Phi_1, \Phi_2, \ldots, \}$ is a dialogue arena, with $\Phi = \{\psi_1, \ldots, \psi_q\}$ a (recall, finite) set of wff constituting a single dialogue context of this arena.

Definition 3 *Let $\mathcal{L} = \{L_j | 1 \leq j \leq l\}$ be a set of* locution types*. A* dialogue fragment *over the dialogue context Φ is a (finite) sequence,*

$$\mu_1 \cdot \mu_2 \cdots \mu_k$$

where $\mu_t = L_{j,t}(\theta_t)$ is the instantiated locution *or* utterance *(with $\theta_t \in \Phi$) at time t. The* commitment *represented by a dialogue fragment δ – denoted $\Sigma(\delta)$ – is a subset of the context Φ.*

The notation $M^*_{\mathcal{L},\Phi}$ is used to denote the set of all dialogue fragments involving instantiated locutions from $\mathcal{L}$; δ to denote an arbitrary member of this set, and $|\delta|$ to indicate the length (number of utterances) in δ.

In order to represent dialogues of interest we need to describe mechanisms by which dialogue fragments and their associated commitments evolve.

Definition 4 *A* dialogue protocol *for the discussion of the context Φ using locution set $\mathcal{L}$ – is a pair $\mathcal{D} = \langle \Pi, \Sigma \rangle$ defined by:*

a. *A* possible dialogue continuation *function –*

$$\Pi : M^*_{\mathcal{L},\Phi} \rightarrow \wp(\mathcal{L} \times \Phi) \cup \{\perp\}$$

The subset of dialogue fragments δ in $M^*_{\mathcal{L},\Phi}$ having $\Pi(\delta) \neq \perp$ is called the set of legal dialogues over $\langle \mathcal{L}, \Phi \rangle$ in the protocol $\mathcal{D}$*, this subset being denoted $T_{\mathcal{D}}$. It is* *required that the* empty dialogue fragment, ϵ *containing no locutions is a legal dialogue, i.e.* $\Pi(\epsilon) \neq \bot$, *and we call the set* $\Pi(\epsilon)$ *the* legal commencement locutions.[2] *We further require that* Π *satisfies the following condition:*

$$\forall \delta \in M^*_{\mathcal{L},\Phi} \ (\Pi(\delta) = \bot) \Rightarrow (\forall \, \mu = \mathcal{L}_j(\theta) \ \Pi(\delta \cdot \mu) = \bot)$$

i.e. if δ *is not a legal dialogue then no dialogue fragment starting with* δ *is a legal dialogue.*

b. A commitment function – $\Sigma : T_{\mathcal{D}} \rightarrow \wp(\Phi)$ *associating each legal dialogue with a subset of the dialogue context* Φ.

This definition abstracts away ideas concerning commencement, combination and termination rules into the pair $\langle \Pi, \Sigma \rangle$ through which the possible dialogues of a protocol and the associated states (subsets of Φ) are defined. Informally, given a legal dialogue, δ, $\Pi(\delta)$ delineates all of the utterances that may be used to continue the discussion.

A dialogue, δ, is *terminated* if $\Pi(\delta) = \emptyset$ and *partial* if $\Pi(\delta) \neq \emptyset$.

We now describe mechanisms for assessing dialogue protocols in terms of the *length* of a dialogue. The following notation is used.

$$\Delta = \{\Delta_k\} = \{\langle \mathcal{L}, \mathcal{D} = \langle \Pi, \Sigma \rangle, \Phi_k \rangle\}$$

is a (sequence of) dialogue schemata for an arena

$$\mathcal{A} = \{\Phi_1, \ldots, \Phi_k, \ldots\}$$

Although one can introduce concepts of dialogue length predicated on the number of utterances needed to attain a particular state Θ, the decision problem we consider will focus on the concept of "minimal length terminated *continuation* of a dialogue fragment δ". Formally

Definition 5 *Let* $\langle \mathcal{L}, \mathcal{D} = \langle \Pi, \Sigma \rangle, \Phi_k \rangle$ *be a dialogue schema* Δ_k *instantiated with the context* Φ_k *of* $\mathcal{A}$. *Let* $\delta \in M^*_{\langle \mathcal{L}, \Phi_k \rangle}$ *be a dialogue fragment. The* completion length *of* δ *under* $\mathcal{D}$ *for the context* Φ_k, *denoted* $\chi(\delta, \mathcal{D}, \Phi_k)$, *is,*

$$\min\{|\eta| : \eta \in T_{\mathcal{D}}, \, \eta = \delta \cdot \zeta, \, \Pi(\eta) = \emptyset\}$$

if *such a dialogue fragment exists, and undefined otherwise.*

Thus the completion length of a (legal) dialogue, δ, is the least number of utterances in a terminated dialogue that *starts* with δ. We note that if δ is *not* a legal dialogue then $\chi(\delta, \mathcal{D}, \Phi_k)$ is always undefined.

The decision problem whose properties we are concerned with is called the *Generic Optimal Utterance Problem.*

[2] Note that we allow $\Pi(\epsilon) = \emptyset$, although the dialogues that result from this case are unlikely to be of significant interest.

Definition 6 *An instance of the* Generic Optimal Utterance Problem *(*GOUP*) comprises,*

$$\mathcal{U} = \langle \Delta, \delta, \mu \rangle$$

where $\Delta = \langle \mathcal{L}, \mathcal{D}, \Phi \rangle$ is a dialogue schema with locution set $\mathcal{L}$, protocol $\mathcal{D} = \langle \Pi, \Sigma \rangle$, and dialogue context Φ; $\delta \in M^*_{\langle \mathcal{L}, \Phi \rangle}$ is a dialogue fragment, and $\mu \in \mathcal{L} \times \Phi$ is an utterance.

An instance $\mathcal{U}$ is accepted if there exists a dialogue fragment $\eta \in M^*_{\langle \mathcal{L}, \Phi \rangle}$ for which all *of the following hold*

1. $\eta = \delta \cdot \mu \cdot \zeta \in T_{\mathcal{D}}$.
2. $\Pi(\eta) = \emptyset$.
3. $|\eta| = \chi(\delta, \mathcal{D}, \Phi)$.

If any of these fail to hold, the instance is rejected.

Thus, given representations of a dialogue schema together with a *partial* dialogue, δ and utterance μ, an instance is accepted if there is a terminated dialogue (η) which commences with the dialogue fragment $\delta \cdot \mu$ and whose length is the completion length of δ under $\mathcal{D}$ for the context Φ. In other words, the utterance μ is such that it *is* a legal continuation of δ leading to a shortest length terminated dialogue.

Our formulation of GOUP, as given in Definition 6, raises a number of questions. The most immediate of these concerns how the schema Δ is to be represented, specifically the protocol $\langle \Pi, \Sigma \rangle$. Noting that we have (so far) viewed $\langle \Pi, \Sigma \rangle$ in abstract terms as mappings from dialogue fragments to sets of utterances (subsets of the context), one potential difficulty is that in "most" cases these will not be *computable*.[3] We can go some way to addressing this problem by representing $\langle \Pi, \Sigma \rangle$ through (encodings of) Turing machine programs $\langle M_\Pi, M_\Sigma \rangle$ with the following characteristics: M_Π takes as its input a *pair* $\langle \delta, \mu \rangle$, where $\delta \in M^*_{\langle \mathcal{L}, \Phi \rangle}$ and $\mu \in \mathcal{L} \times \Phi$, accepting if $\delta \cdot \mu$ is a legal dialogue and rejecting otherwise; similarly M_Σ takes as its input a pair $\langle \delta, \Psi \rangle$ with $\Psi \in \Phi$ accepting if δ is a legal dialogue having $\Psi \in \Sigma(\delta)$, rejecting otherwise. There remain, however, problems with this approach: it is *not* possible, in general, to validate that a given input *is* an instance of GOUP, cf. Rice's Theorem for Recursive Index Sets in e.g., [10, Chapter 5, pp. 58–61]; secondly, even in those cases where one can interpret the encoding of $\langle \Pi, \Sigma \rangle$ "appropriately" the definition places no time-bound on how long the computation of these programs need take. There are two methods we can use to overcome these difficulties: one is to employ 'clocked' Turing machine programs, so that, for example, if no decision has been reached for an instance $\langle \delta, \mu \rangle$ on M_Π after, say $|\delta \cdot \mu|$ steps, then the instance is rejected. The second is to consider *specific* instantiations of GOUP with protocols that can be established "independently" to have desirable efficient decision procedures. More formally,

Definition 7 *Instances of the* Optimal Utterance Problem in Δ *–* OUP$^{(\Delta)}$ *– where $\{\Delta\} = \{\langle \mathcal{L}, \mathcal{D} = \langle \Pi, \Sigma \rangle, \Phi_k \rangle\}$ is a sequence of dialogue schema over the arena $\mathcal{A} = \{\Phi_k : k \geq 1\}$, comprise*

$$\mathcal{U} = \langle \Phi_k, \delta, \mu \rangle$$

[3] For example, it is easy to show that the set of distinct protocols that could be defined using only *two* locutions and a *single element* context is not enumerable.

where $\delta \in M^*_{\langle \mathcal{L}, \Phi_k \rangle}$ is a dialogue fragment, and $\mu \in \mathcal{L} \times \Phi_k$ is an utterance.

An instance $\mathcal{U}$ is accepted if there exists a dialogue fragment $\eta \in M^*_{\langle \mathcal{L}, \Phi_k \rangle}$ for which all *of the following hold*

1. $\eta = \delta \cdot \mu \cdot \zeta \in T_{\mathcal{D}}$.
2. $\Pi(\eta) = \emptyset$.
3. $|\eta| = \chi(\delta, \mathcal{D}, \Phi_k)$.

If any of these fail to hold, the instance is rejected.

The crucial difference between the problems GOUP and OUP$^{(\Delta)}$ is that we can consider the latter in the context of *specific* protocols without being concerned about *how* these are represented – the protocol description does *not* form part of an instance of OUP$^{(\Delta)}$ (only the specific context Φ_k). In particular, should we wish to consider some 'sense of complexity' for a given schema, we could use the device of employing an 'oracle' Turing machine, M_Δ, to report (at unit-cost) whether properties (1–2) hold of any given η. With such an approach, should Δ be such that the set of legal dialogues for a specific context is *finite*, then the decision problem OUP$^{(\Delta)}$ is *decidable* (relative to the oracle machine M_Δ). A further advantage is that any *lower* bound that can be demonstrated for a specific incarnation of OUP$^{(\Delta)}$ gives a lower bound on the "computable fragment" of GOUP. In the next section, we describe a (sequence of) dialogue schemata, $\{\Delta_k^{DPLL}\}$ for which the following computational properties are provable.

1. The set of legal dialogues for Δ_k^{DPLL} is finite: thus every continuation of any legal partial dialogue will result in a legal terminated dialogue.
2. Given $\langle \delta, \mu, \Phi_k \rangle$ with δ a dialogue fragment, μ an utterance, and Φ_k the dialogue context for Δ_k^{DPLL}, there is a deterministic algorithm that decides if $\delta \cdot \mu$ is a legal dialogue using time linear in the number of bits needed to encode the instance.
3. Given $\langle \delta, \Psi, \Phi_k \rangle$ with δ a legal dialogue and Ψ an element of the context Φ_k, there is a deterministic algorithm deciding if $\Psi \in \Sigma(\delta)$ using time linear in the number of bits needed to encode the instance.

We will show that the Optimal Utterance Problem for Δ_k^{DPLL} is both NP–hard and CO-NP–hard.

3 The Optimal Utterance Problem

Prior to defining the schema used as the basis of our results, we introduce the dialogue arena, $\mathcal{A}_{CNF}$ upon which it operates.

Let $\Theta(n)$ ($n \geq 1$) denote the set of all CNF formulae formed from propositional variables $\{x_1, \ldots, x_n\}$ (so that $|\Theta(n)| = 2^{3^n}$). For $\Psi \in \Theta(n)$ with

$$\Psi = \bigwedge_{i=1}^{m} \bigvee_{j=1}^{i_r} y_{i,j} \quad y_{i,j} \in \{x_k, \neg x_k \ : \ 1 \leq k \leq n\}$$

we use C_i to denote the clause $\vee_{j=1}^{i_r} y_{i,j}$. Let Ψ_{rep} be the set of wff given by,

$$\Psi_{rep} = \{\Psi, C_1, \ldots, C_m, x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$$

The *dialogue arena of formulae in* CNF is

$$\mathcal{A}_{\mathrm{CNF}} = \bigcup_{n=1}^{\infty} \bigcup_{\Psi \in \Theta(n)} \{\{\Psi_{rep}\}\}$$

Thus, each different CNF, Ψ gives rise to the dialogue context whose elements are defined by Ψ_{rep}.

We note that $\Phi \in \mathcal{A}_{CNF}$ may be encoded as a word, $\beta(\Phi)$, over alphabet $\{-1, 0, 1\}$

$$\beta(\Phi) = 1^n 0 \alpha \quad \text{with } \alpha \in \{-1, 0, 1\}^{nm}$$

where the i'th clause is described by the sub-word

$$\alpha_{(i-1)*n+1} \ldots \alpha_{i*n}$$

so that

$$\begin{array}{ll} \alpha_{(i-1)n+k} = -1 & \text{if } \neg x_k \in C_i \\ \alpha_{(i-1)n+k} = 1 & \text{if } x_k \in C_i \\ \alpha_{(i-1)n+k} = 0 & \text{if } \neg x_k \notin C_i \text{ and } x_k \notin C_i \end{array}$$

It is thus immediate that given any word $w \in \{-1, 0, 1\}^*$ there is an algorithm that accepts w if and only if $w = \beta(\Phi)$ for some CNF Φ and this algorithm runs in $O(|w|)$ steps.

The basis for the dialogue schema we now define is found in the classic DPLL procedure for determining whether a well-formed propositional formula is satisfiable or not [6, 7]. Our protocol – the DPLL-dialogue protocol – is derived from the realisation of the DPLL-procedure on CNF formulae.

In describing this we assume some ordering

$$\langle \Phi_1, \Phi_2, \ldots, \Phi_k, \ldots \rangle$$

of the contexts in the arena $\mathcal{A}_{CNF}$.

DPLL-Dialogue Schema The sequence of DPLL-*Dialogue Schema* – $\Delta_{DPLL} = \{\Delta_k^{DPLL}\}$ – is defined with contexts from the arena $\mathcal{A}_{CNF}$ as

$$\Delta_k^{DPLL} = \langle \mathcal{L}_{DPLL}, \mathcal{D}_{DPLL} = \langle \Pi_{DPLL}, \Sigma_{DPLL} \rangle, \Phi_k \rangle$$

where

$$\mathcal{L}_{DPLL} = \{\text{ASSERT,REBUT,PROPOSE,DENY,MONO,UNIT}\}$$

and the set Φ_k from $\mathcal{A}_{CNF}$ is,

$$\{\bigwedge_{i=1}^{m} C_i, C_1, \ldots, C_m, x_1, \ldots, x_n, \neg x_1, \ldots, \neg x_n\}$$

Recall that Ψ_k denotes the formula $\bigwedge_{i=1}^{m} C_i$, and C_i is the clause $\vee_{j=1}^{i_r} y_{i,j}$ from the context Φ_k. It will be convenient to regard a clause C both as a disjunction of literals and as a *set* of literals, so that we write $y \in C$ when C has the form $y \vee B$.

The protocol $\langle \Pi_{DPLL}, \Sigma_{DPLL} \rangle$ is defined through the following cases.

At any stage the commitment state – $\Sigma_{DPLL}(\delta)$ consists of a (possibly empty) subset of the clauses of Ψ_k and a (possibly empty) subset of the literals, subject to the condition that y and $\neg y$ are never simultaneously members of $\Sigma_{DPLL}(\delta)$. With the exception of $\{$ASSERT,REBUT$\}$ the instantiated form of any locution involves a literal y.

Case 1: $\delta = \epsilon$ the empty dialogue fragment.

$$\begin{array}{ll} \Pi_{DPLL}(\epsilon) & = \{\text{ASSERT}(\Psi_k)\} \\ \Sigma_{DPLL}(\epsilon) & = \emptyset \\ \Sigma_{DPLL}(\text{ASSERT}(\Psi_k)) & = \{C_i : 1 \leq i \leq m\} \end{array}$$

In the subsequent development, y is a literal and

$$\begin{array}{ll} Open(\delta) & = \{C_i : C_i \in \Sigma_{DPLL}(\delta)\} \\ Lits(\delta) & = \{y : y \in \Sigma_{DPLL}(\delta)\} \\ Single(\delta) & = \{y : \neg y \notin Lits(\delta) \textit{ and } \exists C \in Open(\delta) \textit{ s.t.} \\ & \quad y \in C \textit{ and } \forall z \in C/\{y\}\ \neg z \in Lits(\delta)\} \\ Unary(\delta) & = \{y : \neg y \notin Lits(\delta) \textit{ and} \\ & \quad \forall C \in Open(\delta)\ \neg y \notin C \textit{ and} \\ & \quad \exists C \in Open(\delta) \textit{ with } y \in C\} \\ Bad(\delta) & = \{C_i : C_i \in Open(\delta) \textit{ and} \\ & \quad \forall y \in C\ \neg y \in Lits(\delta)\} \end{array}$$

Informally, $Open(\delta)$ indicates clauses of Ψ_k that have yet to be satisfied and $Lits(\delta)$ the set of literals that have been committed to in trying to construct a satisfying assignment to Ψ_k. Over the progress of a dialogue the literals in $Lits(\delta)$ may, if instantiated to **true**, result in some clauses being reduced to a *single* literal – $Single(\delta)$ is the set of such literals. Similarly, either initially or following an instantantiation of the literals in $Lits(\delta)$ to **true**, the set of clauses in $Open(\delta)$ may be such that some variables occurs only positively among these clauses or only negated. The corresponding literals form the set $Unary(\delta)$. Finally, the course of committing to various literals may result in a set that contradicts all of the literals in some clause: thus this set cannot constitute a satisfying instantiation: the set of clauses in $Bad(\delta)$ if non-empty indicate that this has occurred. Notice that the definition of $Single(\delta)$ admits the possibility of a literal y and its negation being in this set: a case which cannot lead to the set of literals in $Lits(\delta)$ being extended to a satisfying set. Thus we say that the literal set $Lits(\delta)$ is a *failing set* if either $Bad(\delta) \neq \emptyset$ or for some y, $\{y, \neg y\} \subseteq Single(\delta)$.

Recognising that $\Sigma_{DPLL}(\delta) = Open(\delta) \cup Lits(\delta)$ it suffices to describe changes to $\Sigma_{DPLL}(\delta)$ in terms of changes to $Open(\delta)$ and $Lits(\delta)$.

Case 2: $\delta \neq \epsilon$, $Open(\delta) = \emptyset$

$$\Pi(\delta) = \emptyset$$

Case 3: $\delta \neq \epsilon$, $Open(\delta) \neq \emptyset$, $Lits(\delta)$ is *not a failing set*.

There are a number of sub-cases depending on $\Sigma_{DPLL}(\delta)$

Case 3.1: $Single(\delta) \neq \emptyset$.

$$\begin{aligned} \Pi_{DPLL}(\delta) &= \{\text{UNIT}(y) : y \in Single(\delta)\} \\ Open(\delta \cdot \text{UNIT}(y)) &= Open(\delta)/\{C : y \in C\} \\ Lits(\delta \cdot \text{UNIT}(y)) &= Lits(\delta) \cup \{y\} \end{aligned}$$

Case 3.2: $Single(\delta) = \emptyset, Unary(\delta) \neq \emptyset$

$$\begin{aligned} \Pi_{DPLL}(\delta) &= \{\text{MONO}(y) : y \in Unary(\delta)\} \\ Open(\delta \cdot \text{MONO}(y)) &= Open(\delta)/\{C : y \in C\} \\ Lits(\delta \cdot \text{MONO}(y)) &= Lits(\delta) \cup \{y\} \end{aligned}$$

Case 3.3: $Single(\delta) = Unary(\delta) = \emptyset$
Since $Bad(\delta) = \emptyset$ and $Open(\delta) \neq \emptyset$, instantiating the literals in $Lits(\delta)$ will neither falsify nor satisfy Ψ_k. It follows that the set

$$\begin{aligned} Poss(\delta) = \{y : y \notin Lits(\delta),\ \neg y \notin Lits(\delta) \textit{ and} \\ \exists C \in Open(\delta) \textit{ with } y \in C\} \end{aligned}$$

is non-empty. We note that since $Unary(\delta) = \emptyset$, $y \in Poss(\delta)$ if and only if $\neg y \in Poss(\delta)$. This gives,

$$\begin{aligned} \Pi_{DPLL}(\delta) &= \{\text{PROPOSE}(y) : y \in Poss(\delta)\} \\ Open(\delta \cdot \text{PROPOSE}(y)) &= Open(\delta)/\{C : y \in C\} \\ Lits(\delta \cdot \text{PROPOSE}(y)) &= Lits(\delta) \cup \{y\} \end{aligned}$$

This completes the possibilities for Case 3. We are left with,
Case 4: $\delta \neq \epsilon$, $Lits(\delta)$ *is a failing set.*
Let $\delta = \text{ASSERT}(\Psi_k) \cdots \mu_t$

Given the cases above, there are only three utterances that μ_t could be:

$$\mu_t \in \{\text{ASSERT}(\Psi_k),\ \text{PROPOSE}(y),\ \text{DENY}(y)\}$$

Case 4.1: $\mu_t = \mu_1 = \ \text{ASSERT}(\Psi_k)$
Sinces $Lits(\text{ASSERT}(\Psi_k)) = \emptyset$, Ψ_k either contains an *empty clause* (one containing no literals), or for some x both (x) and $(\neg x)$ are clauses in Ψ_k.[4] In either case Ψ_k is "trivially" unsatisfiable, giving

$$\begin{aligned} \Pi_{DPLL}(\text{ASSERT}(\Psi_k)) &= \{\text{REBUT}(\Psi_k)\} \\ \Sigma_{DPLL}(\text{ASSERT}(\Psi_k) \cdot \text{REBUT}(\Psi_k)) &= \emptyset \\ \Pi_{DPLL}(\text{ASSERT}(\Psi_k) \cdot \text{REBUT}(\Psi_k)) &= \emptyset \end{aligned}$$

Case 4.2: $\mu_t = \text{PROPOSE}(y)$

$$\begin{aligned} \Pi_{DPLL}(\delta) &= \{\text{DENY}(y)\} \\ Open(\delta \cdot \text{DENY}(y)) &= Open(\mu_1 \cdots \mu_{t-1})/\{C : \neg y \in C\} \\ Lits(\delta \cdot \text{DENY}(y)) &= Lits(\mu_1 \cdots \mu_{t-1}) \cup \{\neg y\} \end{aligned}$$

[4] Note that we distinguish the wff y (a *literal* used in Ψ_k) and (y) (a *clause* containing the *single* literal y) within the context Φ_k.

Notice this corresponds to a 'back-tracking' move under which having failed to complete a satisfying set by employing the literal y, its negation $\neg y$ is tried instead.
Case 4.3: $\mu_t = \text{DENY}(y)$
Consider the sequence of utterances given by

$$\eta = \mu_2 \cdot \mu_3 \cdots \mu_{t-1} \cdot \mu_t = \text{DENY}(y)$$

We say that η is *unbalanced* if there is a position p such that $\mu_p = \text{PROPOSE}(z)$ with $\text{DENY}(z) \notin \mu_{p+1} \cdots \mu_t$ and *balanced* otherwise. If η is unbalanced let $index(\eta)$ be the *highest* such position for which this holds (so that $p < t$).

We now obtain the final cases in our description.
Case 4.3(a): η is unbalanced with $index(\eta)$ equal to p.

$$\begin{array}{ll} \Pi_{DPLL}(\delta) & = \{\text{DENY}(y) : \mu_p = \text{PROPOSE}(y)\} \\ Open(\delta \cdot \text{DENY}(y)) & = Open(\mu_1 \cdots \mu_{p-1}) / \{C : \neg y \in C\} \\ Lits(\delta \cdot \text{DENY}(y)) & = Lits(\mu_1 \cdots \mu_{p-1}) \cup \{\neg y\} \end{array}$$

Thus this case corresponds to a 'back-tracking' move continuing from the "most recent" position at which a literal $\neg y$ instead of y can be tested.

Finally,
Case 4.3(b): η is balanced.

$$\begin{array}{ll} \Pi_{DPLL}(\delta) & = \{\text{REBUT}(\Psi_k)\} \\ \Sigma_{DPLL}(\delta \cdot \text{REBUT}(\Psi_k)) & = \emptyset \\ \Pi_{DPLL}(\delta \cdot \text{REBUT}(\Psi_k)) & = \emptyset \end{array}$$

We state the following without proof.

Theorem 1 *In the following, δ is a dialogue fragment from $M^*_{\langle \mathcal{L}_{DPLL}, \Phi_k \rangle}$; Φ_k is a context from $\mathcal{A}_{CNF}$, and $N(\delta, \Phi_k)$ is the number of bits used to encode δ and Φ_k under some reasonable encoding scheme.*

1. *The problem of determining whether δ is a legal dialogue for the protocol $\mathcal{D}_{DPLL}$ in context Φ_k can be decided in $O(N(\delta, \Phi_k))$ steps.*
2. *The problem of determining whether δ is a* terminated *legal dialogue for the protocol $\mathcal{D}_{DPLL}$ in context Φ_k is decidable in $O(N(\delta, \Phi_k))$ steps.*
3. *For any $\psi \in \Phi_k$, the problem of determining whether $\psi \in \Sigma_{DPLL}(\delta)$ is decidable in $O(N(\delta, \Phi_k))$ steps.*
4. *For all contexts $\Phi_k \in \mathcal{A}_{CNF}$, the set of legal dialogues over Φ_k in the protocol $\mathcal{D}_{DPLL}$ is* finite.
5. *If δ is a terminated dialogue of $\mathcal{D}_{DPLL}$ in context Φ_k then $\Sigma_{DPLL}(\delta) \neq \emptyset$ if and only if Ψ_k is satisfiable. Furthermore, instantiating the set of literals in $Lits(\delta)$ to* **true***, yields a satisfying assignment for Ψ_k.*

Before analysing this protocol we review how it derives from the basic DPLL-procedure. Consider the description of this below.

```
DPLL-Procedure

Input: Set of clauses $C$
       Set of Literals $L$

if $C = \emptyset$ return true. (SAT)
if any clause of $C$ is empty
  or $C$ contains clauses $(y)$ and $(\neg y)$ (for some literal $y$)
        return false. (UNSAT)
if $C$ contains a clause containing a single literal $y$
        return DPLL$(C^{|y}, L \cup \{y\})$            (U)
if there is a literal $y$ such that $\neg y$ does not occur in any
  clause (and $y$ occurs in some clause)
        return DPLL$(C^{|y}, L \cup \{y\})$            (M)
choose a literal $y$.                                  (B)
if DPLL$(C^{|y}, L \cup \{y\})$
then    return true
else    return DPLL$(C^{|\neg y}, L \cup \{\neg y\})$       (FAIL).
```

For a set of clauses and literal, y, the set of clauses $C^{|y}$ is formed by removing all clauses, C_i for which $y \in C_i$ and deleting the literal $\neg y$ from all clauses C_j having $\neg y \in C_j$.

To test if $\Psi = \wedge_{i=1}^{m} C_i$ is satisfiable, the procedure is called with input $C = \{C_1, \ldots, C_m\}$ and $L = \emptyset$.

Lines (U) and (M) are the "unit-clause" and "monotone literal" rules which improve the run-time of the procedure: these are simulated by the UNIT and MONO locutions. Otherwise a literal is selected – at line (B) – to "branch" on: the PROPOSE locution; should the choice of branching literal FAIL to lead to a satisfying assignment, its negation is tested – the DENY locution. Each time a literal is set to **true**, clauses containing it can be deleted from the current set – the $Open(\delta)$ of the protocol; clauses containing its negation contain one fewer literal. Either all clauses will be eliminated (C is satisfiable) or an empty clause will result (the current set of literals chosen is not a satisfying assignment). When all choices have been exhausted the method will conclude that C is unsatisfiable.

The motivation for the form of the dialogue protocol Δ_k^{DPLL} is the connection between terminated dialogues in T_{DPLL} and *search trees* in the DPLL-procedure above.

Definition 8 *Given a set of clauses C, a* DPLL–search tree *for C is a binary tree, S, recursively defined as follows: if $C = \emptyset$ or C conforms to the condition specified by* UNSAT *in the* DPLL*-procedure, then S is the empty tree, i.e. S contains no nodes. If y is a monotone literal or defines a unit-clause in C, then S comprises a root labelled y whose sole child is a* DPLL*-search tree for the set $C^{|y}$. If none of these four cases apply, S consists of a root labelled with the branching literal y chosen in line (*B*) with at most two children – one comprising a* DPLL*-search tree for the set $C^{|y}$, the other child – if the case (*FAIL*) arises – a* DPLL*-search tree for the set $C^{|\neg y}$.*

A DPLL*-search tree is* full *if no further expansion of it can take place (under the procedure above).*

The size *of a* DPLL*-search tree, S – $\nu(S)$ – is the total number of* edges[5] *in S. A full* DPLL*-search tree, S, is* minimum *for the set of clauses C, if given any full* DPLL*-search tree, R for C, $\nu(S) \leq \nu(R)$. Finally, a literal y is an* optimal branching literal *for a clause set C, if there is a minimum* DPLL*-search tree for C whose root is labelled y.*

We say a set of clauses, C, is *non-trivial* if $C \neq \emptyset$. Without loss of generality we consider only CNF-formulae, Ψ, whose clause set is non-trivial. Of course, during the evolution of the DPLL-procedure and the dialogue protocol $\mathcal{D}_{DPLL}$ sets of clauses which are trivial may result (this will certainly be the case is Ψ is satisfiable): our assumption refers *only* to the initial instance set.

Theorem 2 *Let $\Psi = \wedge_{i=1}^{m} C_i$ be a* CNF*-formula over propositional variables $\langle x_1, \ldots, x_n \rangle$. Let $C(\Psi)$ and Φ_k be respectively the set of clauses in Ψ and the dialogue context from the arena $\mathcal{A}_{CNF}$ corresponding to Ψ, i.e. the set Ψ_{rep} above.*

1. *Given any full* DPLL*-search tree, S, for $C(\Psi)$ there is a legal terminated dialogue, $\delta_S \in T_{DPLL}$ for which,*

$$\delta_S \;=\; \text{ASSERT}(\Psi_k) \cdot \eta_S \cdot \mu$$

and $|\eta_S| = \nu(S)$, with μ being one of the locution types in

$$\{\text{PROPOSE,UNIT,MONO,REBUT}\}.$$

2. *Given any legal terminated dialogue $\delta \;=\; \text{ASSERT}(\Psi_k) \cdot \eta \cdot \mu$, with*

$$\mu \in \{\text{REBUT}(\Psi_k), \text{PROPOSE}(y), \text{MONO}(y), \text{UNIT}(y)\}$$

there is a full DPLL*-search tree, S_δ having $\nu(S_\delta) = |\eta|$.*

Proof. Let Ψ, $C(\Psi)$, and Φ_k be as in the Theorem statement. For Part 1, let S be any full DPLL-search tree for the clause set $C(\Psi)$. We obtain the result by induction on $\nu(S) \geq 0$.

For the inductive base, $\nu(S) = 0$, either S is the empty tree or S contains a single node labelled y. In the former instance, since Ψ is non-trivial it must be the case that Ψ is unsatisfiable (by reason of containing an empty clause or opposite polarity unit clauses). Choosing

$$\delta_S \;=\; \text{ASSERT}(\Psi_k) \cdot \eta_S \cdot \text{REBUT}(\Psi_k)$$

with $\eta_S = \epsilon$ is a legal terminated dialogue (Case 4.1) and $|\eta_S| = 0 = \nu(S)$.

When S contains a single node, so that $\nu(S) = 0$, let y be the literal labelling this. It must be the case that $C(\Psi)$ is *satisfiable* – it cannot hold that $C(\Psi)^{|y}$ *and* $C(\Psi)^{|\neg y}$ *both* yield empty search trees, since this would imply the presence of unit-clauses (y) and $(\neg y)$ in $C(\Psi)$.[6] Thus the literal y occurs in every clause of $C(\Psi)$. If y is a unit-clause, the dialogue fragment,

$$\delta_S \;=\; \text{ASSERT}(\Psi_k) \cdot \text{UNIT}(y)$$

[5] The usual definition of *size* is as the number of *nodes* in S, however, since S is a tree this value is exactly $\nu(S) + 1$.

[6] It should be remembered that *at most one* of $\{y, \neg y\}$ occurs in any clause.

is legal (Case 3.1) and terminated (Case 2). Fixing $\eta_S = \epsilon$ and $\mu = \textsc{unit}(y)$ gives $|\eta_S| = 0 = \nu(S)$ and $\delta = \textsc{assert}(\Psi_k) \cdot \eta_S \cdot \mu$ a legal terminated dialogue. If y is not a unit clause, we obtain an identical conclusion using $\eta_S = \epsilon$ and $\mu = \textsc{mono}(y)$ via Case 3.2 and Case 2.

Now, inductively assume, for some M, that if S_M is a DPLL-search tree for a set of clauses $C(\Psi)$, with $\nu(S_M) < M$ then there is a legal terminated dialogue, δ_{S_M}, over the corresponding context, Φ, with $\delta_{S_M} = \textsc{assert}(\Psi) \cdot \eta_{S_M} \cdot \mu$ and $|\eta_{S_M}| = \nu(S_M)$.

Let S be a DPLL-search tree for $C(\Psi)$ with $\nu(S) = M \geq 1$. Consider the literal, y, labelling the root of S. Since $\nu(S) \geq 1$, the set $C(\Psi)^{|y}$ is non-empty. If $C(\Psi_k)$ contains a unit-clause, then (y) must be one such, thus S comprises the root labelled y and a single child, $S^{|y}$ forming a full DPLL-search tree for the (non-empty) clause set $C(\Psi)^{|y}$. It is obvious that $\nu(S^{|y}) < \nu(S) \leq M$, so from the Inductive Hypothesis, there is a legal terminated dialogue, $\delta^{|y}$ in the context formed by the CNF $\Psi_k^{|y}$. Hence,

$$\delta^{|y} = \textsc{assert}(\Psi_k^{|y}) \cdot \eta^{|y} \cdot \mu$$

and $|\eta^{|y}| = \nu(S^{|y})$. From Case(3.1), the dialogue fragment

$$\delta_S = \textsc{assert}(\Psi_k) \cdot \textsc{unit}(y) \cdot \eta^{|y} \cdot \mu$$

is legal and is terminated. Setting $\eta_S = \textsc{unit}(y) \cdot \eta^{|y}$, we obtain

$$|\eta_S| \; = \; 1 + |\eta^{|y}| \; = \; 1 + \nu(S^{|y}) \; = \; \nu(S)$$

A similar construction applies in those cases where y is a monotone literal – substituting the utterance $\textsc{mono}(y)$ for UNIT(y) – and when y is a branching literal with exactly one child $S^{|y}$ – in this case, substituting the utterance $\textsc{propose}(y)$ for $\textsc{unit}(y)$.

The remaining case is when S comprises a root node labelled y with *two* children – $S^{|y}$ and $S^{|\neg y}$ – the former a full DPLL-search tree for the clause set $C(\Psi)^{|y}$, the latter a full DPLL-search tree for the set $C(\Psi)^{|\neg y}$. We use $\Phi^{|y}$ and $\Phi^{|\neg y}$ to denote the contexts in $\mathcal{A}_{CNF}$ corresponding to these CNF-formulae. As in the previous case, $\nu(S^{|y}) < \nu(S) = M$ and $\nu(S^{|\neg y}) < \nu(S) = M$. Invoking the Inductive Hypothesis, we identify legal terminated dialogues, over the respective contexts $\Phi^{|y}$ and $\Phi^{|\neg y}$

$$\begin{aligned} \delta^{|y} &= \textsc{assert}(\Psi^{|y}) \cdot \eta^{|y} \cdot \mu^{|y} \\ \delta^{|\neg y} &= \textsc{assert}(\Psi^{|\neg y}) \cdot \eta^{|\neg y} \cdot \mu^{|\neg y} \end{aligned}$$

with $|\eta^{|y}| = \nu(S^{|y})$ and $|\eta^{|\neg y}| = \nu(S^{|\neg y})$.

We first note that the set $C(\Psi)^{|y}$ cannot be satisfiable – if it were the search-tree $S^{|\neg y}$ would not occur. We can thus deduce that $\mu^{|y} = \textsc{rebut}(\Psi^{|y})$. Now consider the dialogue fragment, δ_S, from the context Φ_k

$$\textsc{assert}(\Psi_k) \cdot \textsc{propose}(y) \cdot \eta^{|y} \cdot \textsc{deny}(y) \cdot \eta^{|\neg y} \cdot \mu^{|\neg y}$$

Certainly this is a legal terminated dialogue via the Inductive hypothesis and Cases 4.2, 4.3(a–b). In addition, with

$$\eta_S = \textsc{propose}(y) \cdot \eta^{|y} \cdot \textsc{deny}(y) \cdot \eta^{|\neg y}$$

we have

$$|\eta_S| \;=\; 2+|\eta^{|y}|+|\eta^{|\neg y}| \;=\; 2+\nu(S^{|y})+\nu(S^{|\neg y}) \;=\; \nu(S)$$

so completing the Inductive proof of Part 1.

For Part 2 we use an inductive argument on $|\eta| \geq 0$. Let $\delta = \text{ASSERT}(\Psi_k)\cdot\eta\cdot\mu$ be a legal terminated dialogue in T_{DPLL} as above, with

$$\mu \in \{\text{REBUT}(\Psi_k), \text{PROPOSE}(y), \text{MONO}(y), \text{UNIT}(y)\}$$

For the inductive base, we have $|\eta| = 0$, in which event it must hold that

$$\delta \in \{\text{ASSERT}(\Psi_k)\cdot\text{REBUT}(\Psi_k),\ \text{ASSERT}(\Psi_k)\cdot\text{MONO}(y),\ \text{ASSERT}(\Psi_k)\cdot\text{UNIT}(y)\}$$

In the first of these, via Case 4.1, Ψ_k is unsatisfiable by virtue of it containing an empty clause or having both (x) and $(\neg x)$ as clauses. Thus, choosing S as the *empty tree* gives $\nu(S) = |\eta| = 0$. In the remaining two possibilities, Ψ_k must be satisfied by the instantiation that sets the literal y to **true** and now choosing S as the tree consisting of a single node labelled y gives a full DPLL search tree for Ψ_k with $\nu(S) = |\eta| = 0$.

For the inductive step, assume that given any

$$\delta' = \text{ASSERT}(\Psi')\cdot\eta'\cdot\mu$$

a legal terminated dialogue in T_{DPLL} in which $|\eta'| < r+1$ for some $r \geq 0$, there is a full DPLL search tree S' for Ψ' with $\nu(S') = |\eta'|$. We show that if

$$\delta = \text{ASSERT}(\Psi)\cdot\eta\cdot\mu$$

is a legal terminated dialogue in T_{DPLL} in which $|\eta| = r+1$ then we can construct a full DPLL search tree, S for Ψ with $\nu(S) = r+1$. Noting that $|\eta| \geq 1$, let μ_1 be the first locution occuring in η, so that

$$\delta = \text{ASSERT}(\Psi)\cdot\mu_1\cdot\eta'\cdot\mu$$

It must be the case that

$$\mu_1 \in \{\text{MONO}(y),\ \text{UNIT}(y),\ \text{PROPOSE}(y)\}$$

For the first two,

$$\delta' = \text{ASSERT}(\Psi^{|y})\cdot\eta'\cdot\mu$$

is a legal terminated dialogue for the set of clauses $C(\Psi)^{|y}$, thus by the inductive hypothesis there is a full DPLL search tree $S^{|y}$ for this set with $\nu(S^{|y}) = |\eta'| = r$. Defining the tree S by taking a single node labelled y whose only child is the root of $S^{|y}$ provides a full DPLL search tree for Ψ whose size is exactly $|\eta| = r+1$.

The remaining possibility is $\mu_1 = \text{PROPOSE}(y)$. First suppose that the locution $\text{DENY}(y)$ does not occur in η': then, exactly as our previous two cases

$$\delta' = \text{ASSERT}(\Psi^{|y})\cdot\eta'\cdot\mu$$

is a legal terminated dialogue for the set of clauses $C(\Psi)^{|y}$ and we form a full DPLL search tree S for Ψ from $S^{|y}$ – a full DPLL search tree for the set of clauses $C(\Psi)^{|y}$ which has size $|\eta'|$ via the inductive hypothesis – by adding a single node labelled y whose sole child is the root of $S^{|y}$. The resulting tree has size $|\eta|$ as required.

Finally, we have the case in which $\text{DENY}(y)$ does occur in η'. For such,

$$\delta = \text{ASSERT}(\Psi) \cdot \text{PROPOSE}(y) \cdot \eta_1 \cdot \text{DENY}(y) \cdot \eta_2 \cdot \mu$$

Consider the two dialogues

$$\begin{array}{lcl} \delta_y & = & \text{ASSERT}(\Psi^{|y}) \cdot \eta_1 \text{REBUT}(\Psi^{|y}) \\ \delta_{\neg y} & = & \text{ASSERT}(\Psi^{|\neg y}) \cdot \eta_2 \cdot \mu \end{array}$$

Clearly δ_y is a legal terminated dialogue for the set of clauses $C(\Psi)^{|y}$ and, similarly, $\delta_{\neg y}$ one for the set of clauses $C(\Psi)^{|\neg y}$. Hence, by the inductive hypothesis we find full DPLL search trees – $S^{|y}$ and $S^{|\neg y}$ – of sizes $|\eta_1|$ and $|\eta_2|$ respectively for these clause sets. Consider the DPLL search tree, S, formed by adding a single node labelled y whose left child is the root of $S^{|y}$ and whose right child that of $S^{|\neg y}$. Then

$$\nu(S) = \nu(S^{|y}) \ + \ \nu(S^{|\neg y}) \ + \ 2 \ \ = \ \ |\eta_1| + |\eta_2| + 2 = |\eta|$$

Thus completing the inductive argument.

Corollary 1. *An instance,*

$$U = \langle \Phi_k, \text{ASSERT}(\Psi_k), \text{PROPOSE}(y) \rangle$$

of the Optimal Utterance Problem for Δ_{DPLL} is accepted if and only if y is neither a unit-clause nor a monotone literal and y is an optimal branching literal for the clause set $C(\Psi_k)$.

Proof. If y defines a unit-clause or monotone literal in Ψ_k then $\text{PROPOSE}(y)$ is not a legal continuation of $\text{ASSERT}(\Psi_k)$. The corollary is now an easy consequence of Theorem 2: suppose that

$$\delta \ = \ \text{ASSERT}(\Psi_k) \cdot \text{PROPOSE}(y) \cdot \eta \cdot \mu_y$$

is a minimum length completion of $\text{ASSERT}(\Psi_k)$, then Part 2 of Theorem 2 yields a full DPLL-search tree, R, for $C(\Psi_k)$ of size $1 + |\eta|$ whose root is labelled y. If R is not minimum then there is smaller full DPLL-search tree, S. From Part 1 of Theorem 2 this yields a legal terminated dialogue

$$\text{ASSERT}(\Psi_k) \cdot \mu_S \cdot \eta_S \cdot \mu$$

with

$$\nu(S) \ = \ |\mu_S \cdot \eta_S \cdot \mu| - 1 \ < \ |\text{PROPOSE}(y) \cdot \eta \cdot \mu_y| - 1 \ = \ \nu(R)$$

which contradicts the assumption that δ is a minimum length completion.

We now obtain a lower bound on the complexity of $\textsc{oup}^{(\Delta)}$ via the following result of Liberatore [19].

Fact 1 Liberatore ([19]) *Given an instance $\langle C, y\rangle$ where C is a set of clauses and y a literal in these, the problem of deciding whether y is an optimal branching literal for the set C is* NP*–hard* and CO-NP*–hard.*

Theorem 3 *The Optimal Utterance in Δ Problem is* NP*–hard and* CO-NP*–hard.*

Proof. Choose Δ as the sequence of schema $\{\Delta_k^{DPLL}\}$. From Corollary 1 an instance $\langle \Phi_k, \textsc{assert}(\Psi_k), \textsc{propose}(y)\rangle$ is accepted in $\textsc{oup}^{(\Delta)}$ if and only if y does not form a unit-clause of Ψ_k, is not a monotone literal, and is an optimal branching literal for the clause set $C(\Psi_k)$. We may assume, (since these are easily tested) that the first two conditions do not hold, whence it follows that decision methods for such instances of $\textsc{oup}^{(\Delta)}$ yield decision methods for determining if y is an optimal branching literal for $C(\Psi_k)$. The complexity lower bounds now follow directly from Liberatore's results stated in Fact 1.

4 Conclusion

The principal contentions of this paper are three-fold: firstly, in order for a dialogue protocol to be realised effectively in a multiagent setting, each agent must have the capability to determine what contribution(s) it must or should or can make to the discussion as it develops; secondly, in deciding which (if any) utterance to make, an agent should (ideally) take cognisance of the extent to which its utterance is 'optimal'; and, finally, the criteria by which an utterance is judged to be 'optimal' are *application dependent*. In effect, the factors that contributors take into consideration when participating in one style of dialogue, e.g. bargaining protocols, are *not* necessarily those that would be relevant in another style, e.g. persuasion protocols.

We have proposed one possible interpretation of "optimal utterance in persuasion protocols": that which leads to the debate terminating 'as quickly as possible'. There are, however, a number of "length-related" alternatives that may merit further study. We have already mentioned in passing the view explored in [11]. One drawback to the concept of "optimal utterance" as we have considered it, is that it presumes the protocol is "well-behaved" in a rather special sense: taking the aim of an agent in a persuasion process as "to convince others that a particular proposition is valid", the extent to which an agent is successful may depend on the 'final' commitment state attained. In the DPLL-protocol this final state is either *always* empty (if Ψ_k is not satisfiable) or *always* non-empty: the protocol is "sound" in the sense that conflicting interpretations of the final state are not possible. Suppose we consider persuasion protocols where there is an 'external' interpretation of final state, e.g. using a method of defining some (sequence) of mappings $\tau : \wp(\Phi_k) \rightarrow \{\mathbf{true}, \mathbf{false}, \perp\}$, so that a terminated dialogue, δ, with $\tau(\Sigma(\delta)) = \mathbf{true}$ indicates that the persuading agent has successful demonstrated its desired hypothesis; $\tau(\Sigma(\delta)) = \mathbf{false}$ indicates that its hypothesis is *not* valid; $\tau(\Sigma(\delta)) = \perp$ indicates that no conclusion can be drawn.[7] There are good reasons why

[7] For example, game theorists in economics have considered the situation where two advocates try to convince an impartial judge of the truth or otherwise of some claim, e.g. [14, 31].

we may wish to implement 'seemingly contradictory' protocols, i.e. in which the persuasion process for a given context Φ can terminate in any (or all) of **true**, **false** or $\perp$ states, e.g. to model concepts of cautious, credulous, and sceptical agent belief, cf. [27]. In such cases defining "optimal utterance" as that which can lead to a shortest terminated dialogue may not be ideal: the persuading agent's view of "optimal" is not simply to terminate discussion but to terminate in a **true** state; in contrast, "sceptical" agents may seek utterances that (at worst) terminate in the inconclusive $\perp$ state. We note that, in such settings, there is potentially an "asymmetry" in the objectives of individual agents – we conjecture that in suitably defined protocols and contexts with appropriately defined concepts of "optimal utterance" the decision problems arising are likely to prove at least as intractable as those for the basic variant we consider in Theorem 3.

A natural objection to the use of length-related measures to assess persuasion processes is that these do not provide any sense of how convincing a given discourse might be, i.e. that an argument can be presented concisely does not necessarily render it effective in persuading those to whom it is addressed. One problem with trying formally to capture concepts of persuasiveness is that, unlike measures based on length, this is a subjective measure: a reasoning process felt to be extremely convincing by one party may fail to move another. One interesting problem in this respect concerns modeling the following scenario. Suppose we have a collection of agents with differing knowledge and 'prejudices' each of whom an external agent wishes to persuade to accept some proposition, e.g. election candidates seeking to persuade a cross-section of voters to vote in their favour. In such settings one might typically expect contributions by the persuading party to affect the degree of conviction felt by members of the audience in different ways. As such the concept of an 'optimal' utterance might be better assessed in terms of proportionate increase in acceptance that the individual audience members hold after the utterance is made. Recent work in multi-agent argumentation has considered dialogues between agents having different knowledge, different prejudices or different attitudes to the utterance and acceptance of uncertain claims, e.g. [1, 28].

We conclude by mentioning two open questions of interest within the context of persuasion protocols and the optimal utterance problem in these. In practical terms, one problem of interest is, informally, phrased as follows: can one define "non-trivial" persuasion protocols for a "broad" collection of dialogue contexts within which the optimal utterance problem is tractable? We note that, it is unlikely that dialogue arenas encompassing the totality of all propositional formulae will admit such protocols, however, for those subsets which have efficient decision procedures e.g. Horn clauses, 2-CNF formulae, appropriate methods may be available. A second issue is to consider complexity-bounds for other persuasion protocols: e.g. one may develop schema for the arena $\mathcal{A}_{CNF}$ defined via the TPI–dispute mechanism of [32], the complexity (lower and upper bounds) of the optimal utterance problem in this setting is open, although in view of our results concerning $\mathcal{D}_{DPLL}$ it is plausible to conjecture that the optimal utterance problem for $\mathcal{D}_{TPI}$ will also prove intractable.

Acknowledgement

A preliminary version of this article appeared in [13]. We thank the anonymous referees and the AAMAS 2003 audience for their comments.

References

1. L. Amgoud and S. Parsons. Agent dialogues with conflicting preferences. In J. J. Meyer and M. Tambe, editors, *Pre-Proc. Eighth Intern. Workshop on Agent Theories, Architectures, and Languages (ATAL 2001)*, pages 1–14, 2001.
2. T. J. M. Bench-Capon. Specification and implementation of Toulmin dialogue game. In J. C. Hage *et al.*, editor, *Legal Knowledge Based Systems*, pages 5–20. GNI, 1998.
3. T. J. M. Bench-Capon, P. E. Dunne, and P. H. Leng. A dialogue game for dialectical interaction with expert systems. In *Proc. 12th Annual Conf. Expert Systems and their Applications*, pages 105–113, 1992.
4. C. Cayrol, S. Doutre, and J. Mengin. Dialectical proof theories for the credulous preferred semantics of argumentation frameworks. In *Sixth European Conference on Symbolic and Quantitative Approaches to Reasoning with Uncertainty (ECSQARU-2001)*, pages 668–679. Lecture Notes in A.I., 2143, Springer, 2001.
5. V. Conitzer and T. Sandholm. Complexity results about Nash equilibria. Technical Report CMU-CS-02-135, School of Computer Science, Carnegie-Mellon University, May 2002.
6. M. Davis, G. Logemann, and D. Loveland. A machine program for theorem proving. *Communications of the ACM*, 5:394–397, 1962.
7. M. Davis and H. Putnam. A computing procedure for quantification theory. *Journal of the ACM*, 7:201–215, 1960.
8. F Dignum and M. Greaves (editors). *Issues in Agent Communication*. Springer-Verlag, 2000.
9. S. Doutre and J. Mengin. Preferred extensions of argumentation frameworks: Query answering and computation. In *First Intern. Joint Conf. Automated Reasoning (IJCAR 2001)*, pages 272–288. Lecture Notes in A.I., 2083, Springer, June 2001.
10. P.E. Dunne. *Computability Theory - Concepts and Applications*. Ellis-Horwood, 1991.
11. P.E. Dunne. Prevarication in dispute protocols. *Proc. Ninth International Conference on A.I. and Law*, (ICAIL' 03), Edinburgh, June 2003, ACM Press, pages 12–21
12. P.E. Dunne and T.J.M. Bench-Capon. Two party immediate response disputes: Properties and efficiency. *Artificial Intelligence*, 149(2):221–250, 2003.
13. P.E. Dunne and P.J. McBurney Optimal Utterances in Dialogue Protocols *Proc. Second International joint Conference on Autonomous Agents and Multiagent Systems* (AAMAS'03), July 2003, ACM Press, pages 608-615
14. J. Glazer and A. Rubinstein. Debates and decisions: on a rationale of argumentation rules. *Games and Economic Behavior*, 36(2):158–173, 2001.
15. T. F. Gordon. *The Pleadings Game: An Artificial Intelligence Model of Procedural Justice*. Kluwer Academic, Dordrecht, 1995.
16. M. P. Huget, editor, *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies*. Lecture Notes in A.I., 2650, Springer, 2003.
17. M. W. Johnson, P. McBurney, and S. Parsons. When are two protocols the same? In M. P. Huget, editor, *Communication in Multiagent Systems: Agent Communication Languages and Conversation Policies*, pages 253–268. Lecture Notes in A.I., 2650, Springer, 2003.
18. S. Kraus. *Strategic negotiation in multiagent environments*. MIT Press, 2001.
19. P. Liberatore. On the complexity of choosing the branching literal in DPLL. *Artificial Intelligence*, 116:315–326, 2000.

20. A. R. Lodder. *Dialaw: On legal justification and Dialogue Games*. PhD thesis, Univ.of Maastricht, 1998.
21. P. McBurney, R. van Eijk, S. Parsons, and L. Amgoud. A dialogue-game protocol for agent purchase negotiations. *J. Autonomous Agents and Multiagent Systems*, 7(3): 235–273, 2003.
22. P. McBurney and S. Parsons. Representing epistemic uncertainty by means of dialectical argumentation. *Annals of Mathematics and AI*, 32(1–4):125–169, 2001.
23. P. McBurney and S. Parsons. Games that agents play: A formal framework for dialogues between autonomous agents. *J. Logic, Language and Information*, 11:315–334, 2002.
24. P. McBurney and S. Parsons. Chance Discovery using dialectical argumentation. In T. Terano *et al.*, editors, *New Frontiers in Artificial Intelligence*, pages 414–424. Lecture Notes in A.I., 2253, Springer, 2001.
25. P. McBurney, S. Parsons, and M. J. Wooldridge. Desiderata for agent argumentation protocols. In *Proc. First Intern. Joint Conf. on Autonomous Agents and Multiagent Systems*, pages 402–409. ACM Press, 2002.
26. S. Parsons, C. A. Sierra, and N. R. Jennings. Agents that reason and negotiate by arguing. *J. Logic and Computation*, 8(3):261–292, 1998.
27. S. Parsons, M. J. Wooldridge, and L. Amgoud. An analysis of formal inter-agent dialogues. In *Proc. First Intern. Joint Conf. Autonomous Agents and Multiagent Systems*, pages 394–401. ACM Press, 2002.
28. S. Parsons, M. J. Wooldridge, and L. Amgoud. Properties and complexity of some formal inter-agent dialogues. *J. Logic and Computation*, 13(3):347–376, 2003.
29. H. Prakken. *Logical Tools for Modelling Legal Argument*. Kluwer, Dordrecht, 1997.
30. C. Reed. Dialogue frames in agent communications. In Y. Demazeau, editor, *Proc. 3rd Intern. Conf. Multiagent Systems (ICMAS-98)*, pages 246–253. IEEE Press, 1998.
31. A. Rubinstein. Strategic considerations in pragmatics. In *Economics and Language: Five essays*, pages 37–52. Cambridge Univ. Press, 2000.
32. G. Vreeswijk and H. Prakken. Credulous and sceptical argument games for preferred semantics. In *Proc. JELIA'2000, The 7th European Workshop on Logic for Artificial Intelligence.*, pages 224–238, Berlin, 2000. Lecture Notes in A.I., 1919, Springer.
33. D. N. Walton and E. C. W. Krabbe. *Committment in Dialogue: Basic Concepts of Interpersonal Reasoning*. SUNY Press, Albany, 1995.