Using argumentation-based dialogues for distributed plan management

Yuqing Tang
Department of Computer Science
Graduate Center
City University of New York
New York, NY 10016
ytang@gc.cuny.edu

Simon Parsons
Department of Computer & Information Science
Brooklyn College
City University of New York
Brooklyn, NY 11210
parsons@sci.brooklyn.cuny.edu

Abstract

In (Tang & Parsons 2005) we introduced a form of argumentation-based dialogue that generates distributed plans. This paper recaps the results of that work, sketching the way a dialogue proceeds, and describing how the underlying framework may be used for plan management.

Argumentation and planning

We use the framework introduced in (Parsons, Wooldridge, & Amgoud 2003b), which assumes a set of agents $A_1, A_2, \ldots, A_n$. Each agent has a knowledge base Σ_i containing formulae in some logical language $\mathcal{L}$, and these formulae express all the facts the agent knows about the world and its knowledge about how to create plans. In addition to its knowledge base, each agent has a *commitment store*, CS_i, which contains all the formulae that the agent has asserted during a given dialogues. Commitment stores are so named (Hamblin 1970) because by placing a formula in its commitment store, an agent is committing itself to defend that formula if another agent challenges its validity. Such a challenge is met by the assertion of the reason, the argument, that led the asserting agent to believe the formula was worth asserting. Commitment stores are public, accessible to all agents in the dialogue.

We consider a dialogue to start with the agents having agreed upon a goal for which they need a plan. One agent, A_i, then formulates what it believes to be a relevant, achievable sub-plan that will help towards the overall goal. The analogy here is when a group of people sit around trying to come up with a plan to do X, one person may start by saying "I could do Y", or "Bob could do Z" which helps to focus planning by identifying some sequence of actions that the agents believe will help towards the overall goal, and can be used to focus the remainder of the planning process.

In accordance with the rest of the argumentation-based dialogue literature, we think of A_i constructing a plan p by inference on what the agent knows, so:

$$\Sigma_i \cup \bigcup_{j=i}^{j=n} CS_j \vdash_{\mathcal{L}} p$$

where $\vdash_{\mathcal{L}}$ is a proof mechanism for the language $\mathcal{L}$. In other words A_i can make use of anything in its knowledge base or any commitment store to formulate p (all commitments stores will initially be empty).

During the formulation of the plan, A_i keeps track of the formulae used to derive p. The minimal, consistent, set of formulae that allows A_i to derive p is the *support* of p, and together they form (S, p), an argument for p. A_i can then suggest p, adding p into its commitment store.

The next step in the dialogue is for other agents to vet p. In essence this process is as follows. Each A_j checks that it agrees with p. If one doesn't, p is challenged, and A_i puts forward the support S. Each agent then checks if it believes that S holds (each agent can use information in its Σ and all commitment stores to do this checking) and is allowed to put forward counter-arguments if it has them. Eventually (as we showed in (Parsons, Wooldridge, & Amgoud 2003b)) this process ends with agents either agreed that p works for them, or that it doesn't.

If p is accepted by all the agents, and is sufficient to achieve the overall goal from the initial state, then the dialogue terminates. Otherwise another agent will put forward a new (sub)plan p', and the process repeats, constructing an *argumentation structure* which consists of various subplans each connected to the arguments that support them (and, given the recursive nature of support (Parsons, McBurney, & Wooldridge 2004) to counter-arguments and counter-counter arguments.)

Part of a simple argumentation structure is given in Figure 1. Here the overall plan is constructed from sub-plans that include p, p' and p'' (though not in that order), each of which is a sequence of actions. The plan starts with sub-plan p, this is directly followed by p'' and, somewhat later, by p'. The supports for p, p'' and p' are S, S'' and S' respectively — the triangle denotes the set of formulae that support the plan. S'' includes the formula t. t is *attacked* by the argument $(S^*, \neg t)$ and is *defended* by $(S^{**}, \neg u)$. $(S^{**}, \neg u)$ achieves this defence by attacking $u \in S^*$ and not being attacked itself.

Since all assertions are placed in a commitment store, and all items in the argumentation structure have been asserted, this entire structure is known to all agents, and they all agree on the truth-value of all the assertions (Parsons, Wooldridge, & Amgoud 2003a).

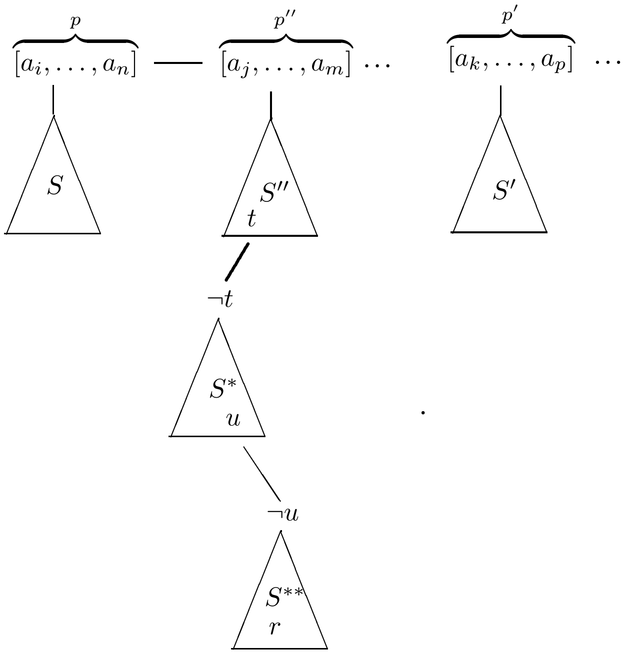

Figure 1: Part of an argumentation structure — see text for denotation.

Plan management

We consider *plan management* to be the activity that, given a plan P (itself a sequence of subplans $p, p'', p', \ldots$) to achieve a goal g, ensures that there is always a plan P' to achieve g until g is achieved or deemed unachievable. In other words, while planning is a one-shot process to construct a plan, plan management is an ongoing process that is concerned with making sure that a given outcome remains attainable.

From this perspective, the initial plan, P, is almost incidental. Typically, we don't care if the plan remains the same — though there may be costs involved if the plan is changed arbitrarily — so long as we have a plan that does what we need it to. Thus one, extreme, form of plan management is to simply re-plan if, for example, the environment changes so that P becomes impractical. This approach seems extreme, because it can be the case that a perfectly good new plan, P', can be found by making a minor change to P, and establishing what such changes are and carrying them out will be much less costly than replanning. Establishing these kinds of modification is particularly straightforward with a plan that is the result of an argumentation process precisely because of the dependencies established during plan formation. Indeed, the dependency information is useful even before embarking on this part of plan management — the dependencies also help us in determining that there is a problem with the initial plan P.

To see why this is the case, let's consider what happens when one of the agents involved in constructing P, agent A_i, learns something new about its environment. On perceiving a new fact q about the environment, A_i adds it to its knowledge base Σ_i. If q is something that has a detrimental effect on the ability of the group of agents to carry out P, then it will be possible for A_i to detect the threat to the plan. In particular, if q contradicts one of the facts r in the argumentation structure, then

$$\Sigma_i \cup \bigcup_{j=i}^{j=n} CS_j \not\vdash_{\mathcal{L}} \neg r$$

(if this were not the case then A_i would not have been able to agree to the plan in the first place) but

$$\Sigma_i \cup \bigcup_{j=i}^{j=n} CS_j \cup \{q\} \vdash_{\mathcal{L}} \neg r$$

and so the structure can be used to check for problems.

In addition, the argumentation structure can be used to identify how the plan can be fixed in order to take account of q. To do this we need to circumvent r, the fact that q demolishes, and the argumentation structure tells us how to do this. If r is in the support of some argument at the periphery of the structure (as it is in Figure 1), then the first thing to try is to find an alternative support for the conclusion of the argument $\neg u$. If this fails then a whole new argument must be sought, and this may have a knock-on effect on whatever the argument connects to in the argumentation structure. For example, if it is not possible to find a new argument in support of $\neg u$, then it is necessary to either find an argument that attacks another formula in S^*, or find an alternative support for p'', or an alternative sub-plan that achieves the same transitions as p''. If this fails, it may be necessary to construct a new overall plan to replace P.

Whatever is required, the structure makes it clear to all agents which parts need to be replaced or rebuilt in order to rehabilitate the plan (and, in the worst case, make it clear to all agents that the entire plan must be replaced)

The reconstruction of the argumentation structure can, of course, be carried out using precisely the same procedure as was used to build the structure in the first place — the recursive nature of that process means that it works just as well for rebuilding a small section of the structure as it does for constructing the whole thing.

References

Hamblin, C. L. 1970. *Fallacies*. London, UK: Methuen and Co Ltd.

Parsons, S.; McBurney, P.; and Wooldridge, M. 2004. Some preliminary steps towards a meta-theory for formal inter-agent dialogues. In Rahwan, I., ed., *Proceedings of the 1st International Workshop on Argumentation in Multiagent Systems*.

Parsons, S.; Wooldridge, M.; and Amgoud, L. 2003a. On the outcomes of formal inter-agent dialogues. In *2nd International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press.

Parsons, S.; Wooldridge, M.; and Amgoud, L. 2003b. Properties and complexity of formal inter-agent dialogues. *Journal of Logic and Computation* 13(3):347–376.

Tang, Y., and Parsons, S. 2005. Argumentation-based dialogues for deliberation. In *Proceedings of the 4th International Conference on Autonomous Agents and Multi-Agent Systems*. ACM Press.