Actions and Arguments:
Preliminaries and Examples

Guillemo R. Simari Alejandro J. García

grs@cs.uns.edu.ar agarcia@cs.uns.edu.ar
Laboratorio de Investigación y Desarrollo en Inteligencia Artificial
Departamento de Ciencias de la Computación
Universidad Nacional del Sur
Av. Alem 1253, (8000) Bahía Blanca, Argentina

Abstract

The goal of this work involves the developing of an argumentation-based formalism that an agent could use for constructing plans. In this formalism, agents will have certain knowledge about the world, and a set of actions that they will be capable of execution. The agent's knowledge will be represented by a knowledge base, using Defeasible Logic Programming. Actions will provide agents the ability of change their world, adding or removing facts from their knowledge base. We will develop here a formalism for representing actions in a defeasible argumentation environment. We will show how an agent may select a sequence of actions in order to satisfy some goal, and this action selection is more involved than expected.

Keywords: Artificial Intelligence, Planning, Defeasible Argumentation.

1 Introduction

The goal of this work involves the developing of an argumentation-based formalism that an agent could use for constructing plans. In this formalism, agents will have certain knowledge about the world, and a set of actions that they will be capable of execution. The agent's knowledge will be represented by a knowledge base, containing a consistent set of facts and a set of defeasible rules. Actions provide agents the ability to change their world. Once an action has been applied, the effect of the action will change the agent's knowledge base, adding or removing facts.

The agent knowledge base will be represented using *Defeasible Logic Programming* (DeLP), a logic programming paradigm based on a defeasible argumentation formalism. DeLP allows the representation of strict and defeasible knowledge and evaluates arguments and counter-arguments in order to *warrant* its conclusions. Actions will have preconditions and consequences. An action will be applicable only if there is a warrant for each precondition, that will be evaluated using the DeLP formalism. The effect of executing an action will be the revision of the knowledge base by the action consequences. Hence new facts may be added or removed.

The interaction between actions and argumentation is twofold. On one hand, defeasible argumentation will be used for testing preconditions through the warrant notion. On the other hand, actions may be used by an agent in order to change the world and

then have a warrant for a conclusion that was previously not warranted. In this work we will analyze different ways in which actions may change the knowledge base in order to warrant a conclusion. We will show how an agent may select a sequence of actions in order to satisfy some goal, and we will show also that action selection is more involved than expected.

2 Agent's knowledge

The agent's knowledge will be represented by a knowledge base $\mathcal{K} = (\Phi, \Delta)$, where Φ will be a consistent set of facts, and Δ a set of defeasible rules. This knowledge base is in fact a restricted *Defeasible Logic Program* [2]. The results already obtained for such argumentation-based extension of logic programming will be used freely here (see [2, 3, 4]). An example of a knowledge base follows.

Example 2.1 *Suppose that the knowledge of an agent h consist of the facts: h is a cast away, h is at the beach, h is at a palm tree, and it is raining. And suppose that the agent has the following defeasible rules as part of its knowledge: "having no water is a good reason for not surviving" and "if you are at the beach then usually you have a stone". Then the agent's knowledge base will consist of the following sets:*

$$\Phi = \{cast\_away(h), at(h, beach), at(h, palm\_tree), is(raining)\}, \text{ and}$$
$$\Delta = \left\{ \begin{array}{l} \sim survive(X) \prec \sim has(X, water) \\ has(X, stone) \prec at(X, beach) \end{array} \right\}$$

In DeLP a literal q is *warranted* if there exists a non-defeated *argument* $\mathcal{A}$ supporting q. An argument $\mathcal{A}$ for a literal q is a minimal and consistent set of defeasible rules that allows to infer q. In order to establish whether $\mathcal{A}$ is a non-defeated argument, *argument rebuttals* or *counter-arguments* that could be *defeaters* for $\mathcal{A}$ are considered, *i.e.*, counter-arguments that by some criterion, are preferred to $\mathcal{A}$. Since counter-arguments are arguments, there may exist defeaters for them, and so on. This prompts a complete *dialectical analysis*. An example of these concepts follows. The interested reader is referred to [2, 3, 4] for details about DeLP.

Example 2.2 *Consider the following knowledge base* $\mathcal{K} = (\Phi, \Delta)$

$\Phi = \{a, b, c, \sim d\}$

$\Delta = \{(p \prec b), (q \prec r), (r \prec d), (\sim r \prec s), (s \prec b), (\sim s \prec a, b), (w \prec b), (\sim w \prec b, c)\}$

Here, the literal p has the argument $\mathcal{A} = \{p \prec b\}$ *supporting it,* $\mathcal{A}$ *is undefeated because there is no counter-argument for it. Hence, p is warranted.*

The literal q has the argument $\mathcal{A}_1 = \{ (q \prec r), (r \prec d) \}$, *but* $\mathcal{A}_1$ *is defeated by* $\mathcal{A}_2 = \{ (\sim r \prec s), (s \prec b) \}$, *that attacks r, an inner point in* $\mathcal{A}$. *The argument* $\mathcal{A}_2$ *is in turn defeated by* $\mathcal{A}_3 = \{ (\sim s \prec a, b) \}$. *Therefore, q is warranted because its supporting argument* $\mathcal{A}_1$ *has only one defeater* $\mathcal{A}_2$ *that it is defeated by* $\mathcal{A}_3$, *and* $\mathcal{A}_3$ *has no defeaters.*

Observe that there is no warrant for $\sim r$ *because* $\mathcal{A}_2$ *is defeated by* $\mathcal{A}_3$. *The literals t and* $\sim t$ *have no argument, so neither of them is warranted. Every fact of* Φ *is trivially warranted, because no counter-argument can defeat a fact.*

3 Defeasible actions

Besides its knowledge base $\mathcal{K}$, an agent will have a set of actions Γ that it may use to change its world. Once an action has been applied, the effect of the action will change the set $\mathcal{K}$. The formal definitions follows.

Definition 3.1 (Action) *An action A is an ordered triple $\langle \mathsf{P}, \mathsf{X}, \mathsf{C} \rangle$, where P is a set of literals representing preconditions for A, X is a consistent set of literals representing consequences of executing A, and C is a set of constraints of the form not L, where L is a literal. We will denote actions as follows:*

$$\{X_1, \ldots, X_n\} \xleftarrow{A} \{P_1, \ldots, P_m\}, not\ \{C_1, \ldots, C_k\}$$

Notice that the notation not $\{C_1, \ldots, C_k\}$ represents $\{not\ C_1, \ldots, not\ C_k\}$.

Definition 3.2 (Applicable Action) *Let $\mathcal{K} = (\Phi, \Delta)$ be an agent's knowledge base. Let Γ be the set of actions available to that agent. An action A in Γ, defined as before, is applicable if every precondition P_i in P has a warrant built from (Φ, Δ) and every constraint C_i in C fails to be warranted.*

Definition 3.3 (Action Effect) *Let $\mathcal{K} = (\Phi, \Delta)$ be an agent's knowledge base. Let Γ be the set of actions available to that agent. Let A be an applicable action in Γ defined by:*

$$\{X_1, \ldots, X_n\} \xleftarrow{A} \{P_1, \ldots, P_m\}, not\ \{C_1, \ldots, C_k\}$$

The effect of executing A is the revision of Φ by X, i.e. $\Phi^{*\mathsf{X}} = \Phi^{*\{X_1,\ldots,X_n\}}$. Revision will consist of removing any literal in Φ that is complementary of any literal in X and then adding X to the resulting set. Formally:

$$\Phi^{*\mathsf{X}} = \Phi^{*\{X_1,\ldots,X_n\}} = (\Phi - \overline{\mathsf{X}}) \cup \mathsf{X}$$

where $\overline{\mathsf{X}}$ represents the set of complements of members of X.

Example 3.4 *Let $\mathcal{K}= (\Phi, \Delta)$ be an agent's knowledge base as defined in Example 2.2*
$\Phi = \{a, b, c, \sim d\}$
$\Delta = \{(p -\!\prec b), (q -\!\prec r), (r -\!\prec d), (\sim r -\!\prec s), (s -\!\prec b), (\sim s -\!\prec a, b), (w -\!\prec b), (\sim w -\!\prec b, c)\}$
And Γ the set of available actions containing only:

$$\{\sim a, d, x\} \xleftarrow{A} \{a, p, q\}, not\ \{t, \sim t, w\}$$

That action is applicable because every literal in the precondition set has a warrant, and no constraints in $\{t, \sim t, w\}$ are warranted (see Example 2.2). If the action is executed the set of facts becomes:

$$\Phi' = \{b, c, \sim a, d, x\}$$

Sometimes some of the preconditions have to be 'consumed' by the action. For example the action of eating an apple will require the apple as a precondition but the apple will disappear after the action has been executed. This can be easily denoted by adding

the complement of consumed preconditions as consequences of the action. In our apple example we can denote:

$$\{\sim apple\} \overset{eats-apple}{\longleftarrow} \{apple\}, not\ \{\}$$

In a more general example suppose that an action A with a consequence q should be applicable when p, r and s are warranted but when A is executed then the preconditions p and r should disappear and s should remain. Then the action should be:

$$\{q, \sim p, \sim r\} \overset{A}{\longleftarrow} \{p, r, s\}, not\ \{\}.$$

As the reader may notice, the interaction between actions and the defeasible argumentation formalism is twofold. On one hand, as stated by Definition 3.2, defeasible argumentation is used for testing preconditions and constraints through the warrant notion. On the other hand, actions may be used by agents in order to change the world (actually the set Φ) and then have a warrant for a literal h that has no warrant from the current knowledge base (Φ, Δ). We will analyze this last situation in the next section.

4 Argumentation through actions

Suppose that for some reason, an agent needs a warrant for a literal h, but from its current knowledge base (Φ, Δ) it is not possible to have a warrant for h. Since the agent has the possibility of changing the world by executing actions, then the agent may select and execute some applicable action that changes the set Φ in such a way that a warrant for h may be obtained. There are several ways of changing Φ for having a warrant for a literal: new literals can be added to produce new arguments that support h, or some literals can be removed from Φ in order to disable defeaters. Some examples follows.

Example 4.1 (case 1: building an argument) *Let $\mathcal{K} = (\Phi, \Delta)$ be an agent's knowledge base:*

$\Phi = \{p, r\}$

$\Delta = \{(h -\!\prec p, q)\}$

And Γ the set of available actions containing only: $\{q\} \overset{A}{\longleftarrow} \{r\}, not\ \{\}$
Here it is not possible to build an argument for h from $\mathcal{K}$. However, after executing action A the set of facts becomes

$$\Phi' = \{p, q, r\}$$

and from (Φ', Δ) we obtain the warrant for h.

In the last example the action was executed in order to add a literal necessary for obtaining a supporting argument for h. In the case of the example below there is a different situation: there exists an argument $\mathcal{A}_1$ for h, but there is one defeater blocking $\mathcal{A}$. The warrant will be obtained by executing an action that disables the defeater.

Example 4.2 (case 2: disable a defeater) *Let $\mathcal{K} = (\Phi, \Delta)$ be an agent's knowledge base:*

$\Phi = \{p, r, q\}$

$\Delta = \{(h \prec q), (\sim h \prec r)\}$

And Γ *the set of available actions containing only:* $\{\sim r\} \xleftarrow{A} \{p, r\}, not\ \{\}$
Here there exists an argument $\mathcal{A}_1$=$\{h \prec p, q\}$ *for* h *from* $\mathcal{K}$*, but there is no warrant for* h *because* $\mathcal{A}_2$=$\{(\sim h \prec r)\}$ *defeats* $\mathcal{A}_1$*. However, if action* A *is executed, the literal* r *is removed from* Φ *and then the set of facts becomes*

$$\Phi' = \{p, q, \sim r\}$$

Now, from (Φ', Δ) *we obtain the warrant for* h *because the argument* $\mathcal{A}_2$ *cannot be constructed, and then* $\mathcal{A}_1$ *has no defeaters.*

In the following example an action is executed in order to reinstate the argument that supports h

Example 4.3 (case 3: defeating a defeater) *Let* $\mathcal{K}$= (Φ, Δ) *be an agent's knowledge base:*
$\Phi = \{p, s, q\}$
$\Delta = \{(h \prec q), (\sim h \prec r), (r \prec s), (\sim r \prec s, t)\}$

And Γ *the set of available actions containing only:* $\{t\} \xleftarrow{A} \{p\}, not\ \{\}$
From $\mathcal{K}$ *it is possible to build the argument* $\mathcal{A}_1$=$\{h \prec p, q\}$ *for* h*, but there is no warrant for* h *because* $\mathcal{A}_2$=$\{(\sim h \prec r), (r \prec s)\}$ *defeats* $\mathcal{A}_1$*. Here it is not possible to disable the defeater* $\mathcal{A}_2$*. However, if action* A *is executed, then* t *can be added to* Φ *and then the set of facts becomes*

$$\Phi' = \{p, s, q, t\}$$

Now, from (Φ', Δ) *a new argument* $\mathcal{A}_3$= $\{(\sim r \prec s, t)\}$ *can be obtained.* $\mathcal{A}_3$ *defeats* $\mathcal{A}_2$ *and therefore reinstates* $\mathcal{A}_1$*. Since* $\mathcal{A}_3$ *has no defeaters, then* h *is warranted.*

Sometimes, executing one single action is not enough, and a sequence of actions is needed. Here follows one example.

Example 4.4 (sequence of actions) *Let* $\mathcal{K}$= (Φ, Δ) *be an agent's knowledge base:*
$\Phi = \{p, s\}$
$\Delta = \{(h \prec p, q)\}$
And Γ *the set of available actions containing:*
$\{q\} \xleftarrow{A} \{r\}, not\ \{\}$
$\{r\} \xleftarrow{B} \{s\}, not\ \{\}$
From $\mathcal{K}$ *no argument for* h *can be built. Action* A *could add the literal* q *necessary for having a warrant for* h*, but action* A *is not applicable because there is no warrant for its precondition* r*. However, action* B *is applicable and by executing* B *then* Φ=$\{p, s, r\}$*. With this new set* Φ *action* A *is now applicable and the literal* q *can be added to the set of facts. After executing* A *the set of facts becomes* $\{p, q, r, w\}$ *and now it is possible to build a warrant for* h*.*

4.1 Unexpected side effects

In the last examples we have shown that actions can be executed in order to change the world (actually Φ) and then have a warrant for a literal that was previously not warranted. However, when an action is executed for changing Φ, it could also produce some unexpected side effects if more than one literal is added or deleted by the action. The following example shows that literals necessary for the construction of arguments must be protected from the effects of the executed actions.

Example 4.5 *Let* $\mathcal{K} = (\Phi, \Delta)$ *be an agent's knowledge base:*
$\Phi = \{p, r\}$
$\Delta = \{(h -\!\prec p, q)\}$
And Γ *the set of available actions containing:*

$$\{q, \sim p\} \xleftarrow{A} \{r\}, not\ \{\}$$

Suppose that the agent wants to have a warrant for h*. From* $\mathcal{K}$ *no argument for* h *could be built because* q *is not derivable. Using the action* A *the literal* q *becomes a fact but, the literal* p *is deleted by* A *from* Φ*. After executing action* A *the set of facts becomes*

$$\Phi' = \{\sim p, q, r\}$$

Now the literal q *is present, but the literal* p *that is also needed for building an argument for* h *is now absent. Therefore,* h *cannot be warranted.*

In order to avoid problems like the ones shown in example 4.5 a set R of protected facts will be built during argumentation, and there will be an extra condition for using an action with set of consequences X:

$$\overline{\mathsf{X}} \cap\ \mathsf{R} = \emptyset$$

Definition 4.6 (Applicable Action for constructing and argument)
Let $\mathcal{K} = (\Phi, \Delta)$ *be an agent's knowledge base, and let* Γ *be the set of actions available to that agent. Let* $\mathcal{A}_1$ *be a set of rules that is being used for building an argument, and* R *be the literals from* Φ *needed for the derivation of the rules in* $\mathcal{A}_1$*. An action* $A = \langle \mathsf{P}, \mathsf{X}, \mathsf{C} \rangle \in \Gamma$ *is applicable for building an argument with* $\mathcal{A}_1$ *if every precondition* P_i *in* P *has a warrant built from* (Φ, Δ)*, every constraint* C_i *in* C *fails to be warranted, and* $\overline{\mathsf{X}} \cap\ \mathsf{R} = \emptyset$*.*

5 Planning

In order to satisfy some goal g, an agent could first verify whether g is warranted. If g is warranted then there is nothing else to do. However, if g is not warranted then the agent may look for a sequence of actions (plan) that change the world in order to warrant g. Searching for a plan involves the interaction of argumentation and actions in two ways: actions may be needed for obtaining warrants, and warrants may be needed for action preconditions or action constraints. We will start by showing a meaningful example.

Let $\mathcal{K}$= (Φ, Δ) be an agent's knowledge base, where: Φ= { cast_away(h), at(h,beach), at(h,palm_tree), is(raining) }, expressing that h is a cast away, h is at the beach, h is at the palm tree, and it is raining. And

$$\Delta = \left\{ \begin{array}{l} \sim survive(X) -\!\prec \sim has(X, water) \\ \sim has(X, water) -\!\prec cast\_away(X) \\ survive(X) -\!\prec not\ \sim survive(X) \\ has(X, water) -\!\prec cast\_away(X), collected(X, water) \\ has(X, container) -\!\prec has(X, cup) \\ has(X, container) -\!\prec has(X, coconut\_shell) \\ has(X, opening\_tool) -\!\prec has(X, axe) \\ has(X, opening\_tool) -\!\prec has(X, sharp\_stone) \\ has(X, stone) -\!\prec at(X, beach) \end{array} \right\}$$

be the set of defeasible rules. The set Γ of available actions will be:

$\{collected(X, water)\} \overset{collect\_rain}{\longleftarrow} \{has(X, container), is(raining)\}, not\ \{asleep(X)\}$
$\{\sim has(X, coconut), has(X, coconut\_shell)\} \overset{open\_coconut}{\longleftarrow} \{has(X, coconut), has(X, opening\_tool)\}, not\ \{\}$
$\{has(X, coconut)\} \overset{get\_coconut}{\longleftarrow} \{at(X, palm\_tree)\}, not\ \{\}$
$\{has(X, sharp\_stone)\} \overset{make\_sharp\_stone}{\longleftarrow} \{has(X, stone)\}, not\ \{\}$

From its knowledge base (Φ, Δ) the agent has a warrant for "$\sim$survive(h)",

$$\mathcal{A}_1 = \left\{ \begin{array}{l} \sim survive(h -\!\prec \sim has(h, water) \\ \sim has(h, water) -\!\prec cast\_away(h) \end{array} \right\}$$

i.e., in this situation the agent h will not survive. However, it could change the situation executing some actions. Observe that the argument $\mathcal{A}_2$ for $has(h, water)$ could be built if the literal $collected(h, water)$ would be present in Φ.

$$\mathcal{A}_2 \ = \left\{ \ has(h, water) -\!\prec cast\_away(h), collected(h, water) \ \right\}$$

If the argument $\mathcal{A}_2$ could be built, then $\mathcal{A}_2$ would defeat $\mathcal{A}_1$, and there would be no warrant for $\sim survive(h)$. Unfortunately, the literal $collected(h, water)$ necessary for building $\mathcal{A}_2$ is not in Φ. However, the agent may find a sequence of actions that add $collected(h, water)$ to its knowledge base.

To obtain $collected(h, water)$ the action *collect_rain* could be executed if it were applicable. It is not applicable because there is no warrant for $has(h, container)$. The action *open_coconut* would do the job, but there are two literals in the preconditions that are not warranted. The actions *get_coconut* and *make_sharp_stone* will provide the literals needed for constructing those warrants.

The sequence of actions that need to be executed to have $collected(h, water)$, and their effect changing Φ is shown in the following table:

Action	Φ
get_coconut	{ has(h, coconut), cast_away(h), at(h,beach), at(h,palm_tree), is(raining) }
make_sharp_stone	{ has(h, sharp_stone), has(h, coconut), cast_away(h), at(h,beach), at(h,palm_tree), is(raining) }
open_coconut	{ has(h, coconut_shell),has(h, sharp_stone), cast_away(h), at(h,beach), at(h,palm_tree), is(raining) }
collect_rain	{ collected(h, water), has(h, coconut_shell), has(h, sharp_stone), cast_away(h), at(h,beach), at(h,palm_tree), is(raining) }

Once this sequence of actions is executed from the resulting set of facts it is possible to build the argument:

$$\mathcal{A}_2 = \left\{ \; has(h, water) \prec cast\_away(h), collected(h, water) \; \right\}$$

which defeats the argument for $\sim survive(h)$ and therefore now the argument:

$$\mathcal{A} = \left\{ \; survive(h) \prec not \sim survive(h) \; \right\}$$

becomes a warrant for $survive(h)$.

5.1 Interaction among Actions

As stated above, actions can be used for changing the world in order to have a warranted for a literal that was previously unwarrant. However, action selection is not a trivial task. We will now introduce some examples that show that action selection is more involved than expected.

Example 5.1 *Let* $\mathcal{K} = (\Phi, \Delta)$ *be an agent's knowledge base:*
$\Phi = \{p, s\}$
$\Delta = \{(h \prec p, q), (t \prec w)\}$
And Γ *the set of available actions containing:*

$$\{q\} \xleftarrow{A} \{r\}, not\ \{t\}$$
$$\{w, r\} \xleftarrow{B} \{s\}, not\ \{\}$$

Suppose that the agent wants a warrant for h*. From* $\mathcal{K}$ *no argument for* h *could be built because* q *is not derivable. Through action* A *the literal* q *could be added to the set of facts, but the literal* r *is also needed as a precondition for executing* A*. Observe that action* B *would provide* r *and, even though it is not necessary will also add* w*. Unfortunately, if* w *is added the literal* t *becomes warranted. This will prevent the use of action* A*, because it activates a constraint for* A*.*

The example above shows one problem of selecting a sequence of actions backwards. Action A is applicable with respect to Φ but not applicable after the execution of B, because a new argument can be built. The following example shows a similar case, but here instead of having a new argument for a constraint, a literal is removed and a defeater that was preventing a warrant for the constraint disappear.

Example 5.2 *Let* $\mathcal{K}=(\Phi,\Delta)$ *be an agent's knowledge base:*
$\Phi=\{p,s,w,z\}$
$\Delta=\{(h-\prec p,q),(t-\prec w),(\sim t-\prec w,z)\}$
And Γ *the set of available actions containing:*

$$\{q\} \xleftarrow{A} \{r\}, not\ \{t\}$$
$$\{\sim z, r\} \xleftarrow{B} \{s\}, not\ \{\}$$

Again, from $\mathcal{K}$ *no argument for* h *could be built. Through action* A *the literal* q *could be added to the set of facts, but the precondition* r *is also needed. Observe that there is no warrant for the constraint* t *because the argument* $\{\ t-\prec w\}$ *for* t *is being defeated by* $\{\sim t-\prec w,z\}$*. Action* B *would provide* r *and, even though it is not necessary will also remove* z*. Unfortunately, when* z *is removed the literal* t *becomes warranted because the defeater that was stopping the warrant for* t *is no longer valid. This will prevent the use of action* A*.*

The following example shows that sometimes the side effect can be positive.

Example 5.3 *Let* $\mathcal{K}=(\Phi,\Delta)$ *be an agent's knowledge base:*
$\Phi=\{p,s,w\}$
$\Delta=\{(h-\prec p,q),(t-\prec w),(\sim t-\prec w,z)\}$
And Γ *the set of available actions containing:*

$$\{q\} \xleftarrow{A} \{r\}, not\ \{t\}$$
$$\{z, r\} \xleftarrow{B} \{s\}, not\ \{\}$$

Again, from $\mathcal{K}$ *no argument for* h *could be built. Through action* A *the literal* q *could be added to the set of facts. Note that literal* r *is needed and that also the constraint* t *is warranted. The effect of executing action* B *will be twofold. It will provide* r*, which is necessary as a precondition for action* A*, and also* z *which is needed to defeat the argument for the literal* t*. This will allow the use of action* A*.*

Consider now the knowledge base $\mathcal{K}=(\Phi,\Delta)$, $\Phi=\{p,s,z\}$, $\Delta=\{(h-\prec p,q)\}$, and Γ the set of available actions containing:

$$\{q\} \xleftarrow{A} \{r\}, not\ \{t\}$$
$$\{r, \sim p\} \xleftarrow{B_1} \{s\}, not\ \{\}$$
$$\{r, t\} \xleftarrow{B_2} \{s\}, not\ \{\}$$
$$\{r, x\} \xleftarrow{B_3} \{s\}, not\ \{\}$$
$$\{r\} \xleftarrow{B_4} \{s\}, not\ \{\}$$
$$\{r\} \xleftarrow{B_5} \{s,z\}, not\ \{\}$$
$$\{r\} \xleftarrow{B_6} \{s\}, not\ \{w,v\}$$
$$\{r\} \xleftarrow{B_7} \{y\}, not\ \{\}$$
$$\{y\} \xleftarrow{C} \{z\}, not\ \{\}$$

From $\mathcal{K}$ no argument for h could be built. However, through action A the literal q could be added to the set of facts, but the literal r is needed. Actions B_1, B_2, B_3, B_4, B_5 and B_6 can be chosen in order to have the literal r. However, B_1 and B_2 cannot be used because B_1 removes p, that it is needed, and B_2 add t (a constraint in A). For the rest of actions some selection criterion may be applied:

- minimize undesired change: select an action with minimal consequences
- minimize preconditions: select an action with less preconditions
- minimal constraints: select an action with less constraints

For example action B_4 has less consequences than B_3, and action B_4 has less preconditions than B_5, and also less constraints than B_6. Therefore action B_4 should be selected first.

Another criterion could be to prefer actions with preconditions that do not require executing other actions. For instance B_7 requires the execution of action C, whereas B_4 has a fact as its preconditions.

6 Conclusions and Future Developments

We have defined here a formalism for representing knowledge and actions. With this formalism an agent may use a DeLP program for representing facts about the world, and defeasible rules that allow the inference of other conclusions. The agent will warrant its conclusions using the underlying defeasible argumentation formalism of DeLP.

Actions provide agents the ability to change their world. We have shown how an agent can select and action in order to warrant a literal or performing some task. The next step will be to combine this formalism with existing planning techniques [10] in order to obtain an argumentative-based planning system.

Another research line will be the development of a collaborative environment for multiple agents. The formalism will be extended allowing several agents to collaborate in plan formation. Thus, an agent could request to other agents a plan for a task that it is not able to perform.

References

[1] John Fox and Simon Parsons. On using arguments for reasoning about action and values. In *Proceedings of the AAAI Spring Symposium on Qualitative*. Stanford, 1997.

[2] Alejandro J. García. *Defeasible Logic Programming: Definition, Operational Semantics and Parallelism*. PhD thesis, Computer Science Department, Universidad Nacional del Sur, Bahía Blanca, Argentina, December 2000.

[3] Alejandro J. García and Guillermo R. Simari. Parallel defeasible argumentation. *Journal of Computer Science and Technology Special Issue: Artificial Intelligence and Evolutive Computation. http://journal.info.unlp.edu.ar/*, 1(2):45–57, 1999.

[4] Alejandro J. García, Guillermo R. Simari, and Carlos I. Chesñevar. An argumentative framework for reasoning with inconsistent and incomplete information. In *Workshop on Practical Reasoning and Rationality*. 13th biennial European Conference on Artificial Intelligence (ECAI-98), August 1998.

[5] Pablo Noriega and Carles Sierra. Towards layered dialogical agents. In *Proc. of the ECAI'96 Workshop on Agents, Theories, Architectures and Languages (Budapest)*, pages 69–81, 1996.

[6] John Pollock. Implementing defeasible reasoing. *workshop on Computation Dialectics*, 1996.

[7] Jordi Sabater, Carles Sierra, Simon Parsons, and Nick Jennings. Engineering executable agents using multi-context systems. *Journal of Logic and Computation (In-press)*, 2001.

[8] Bart Verheij. *Rules, Reasons, Arguments: formal studies of argumentation and defeat*. PhD thesis, Maastricht University, Holland, December 1996.

[9] Gerard A.W. Vreeswijk. Abstract argumentation systems. *Artificial Intelligence*, 90:225–279, 1997.

[10] Daniel S. Weld. Recent advances in ai planning. *AI Magazine*, August 1999.