Chapter 4

Representing commonsense knowledge

In this chapter, we propose a methodology for commonsense representation in ASP. In addition, we show how the ideological conflict domain can be represented in ASP, following the same methodology. We choose this domain, because it is dynamic, it has common sense involved, and it has not been done before.

4.1 Motivation for a methodology

In a recent discussion among the members of the Texas Action Group (TAG), researchers in the area of Knowledge Representation mention the need for a methodology for knowledge representation. Dr. Michael Gelfond thinks that finding the right methodology of formalization, including building micro-theories, learning how to expand them and how to combine them in larger modules is one of the most interesting challenges we are facing now. A member from TAG proposes to divide a formalization into two parts [members, 2006]:

- a body of background knowledge
- a translation of given English sentences

In addition, Dr. Gelfond mention that the current goal is to build micro-theories and after that to select examples to test them. For instance, we might have theory of fighting for a prize, theory of ownership, etc. In Section 4.2, we propose a methodology, that can be combined with the previous ideas for a more complete result.

4.2 The methodology

The need of a methodology for representing commonsense knowledge is growing every day. The motivation for the construction of a methodology is that we need to unify all the representations around the world. In order to obtain a precise and universe representation, we need to follow the same steps. Here we propose the following stages:

- The first step in the knowledge representation should be the choice of a domain. It has to have specific characteristics that make it possible to represent. For instance, the domain should be dynamic, it should involve simple commonsense reasoning and not complicated reasoning, and it should be interesting.
- After the domain is chosen, we have to introduce the types. What are the fluents, what are the actions? Moreover, we have to identify what are the relations between the objects and their properties, that is, we have to realize what laws are going to apply (inertia, static laws, dynamic laws, etc.).
- The third step is to model the domain in an action description language, such as AL.
- Next we have to represent the domain in ASP. Because we want to reuse our knowledge bases in the future, it is important to separate the domain dependent from the domain independent part.
- The last step is to formalize the history in specific, that is to instantiate it.

In Section 4.3, we show how to apply the methodology to the ideological conflict problem.

4.3 Ideological conflict

In the following sections, we discuss the applicability of the knowledge representation language *A-Prolog* for the design and implementation of a commonsense knowledge base about ideologies. We also present a formalization of a simple motivating story, which involves an ideological conflict between countries. The emphasis is on the development and implementation of a commonsense knowledge base needed for the axiomatization of the domain **ideological conflict**. We use *A-Prolog* (a language of logic programs under the answer set semantics), to model ideologies and to detect conflicts between them. The notion of ideological conflict presented in this paper is a special case of a more general notion of war of ideologies, which is an important topic for the intelligence community. This example is also a contribution that we made.

4.4 Introduction and motivation

The current political situation around the world is rapidly evolving. The intelligence organizations of all the countries are trying to understand the global consequences of different actions of their nations. It is important to be able to create automated tools to help those organizations. These tools should contain large knowledge bases about political sciences and commonsense knowledge. Therefore, the representation of knowledge regarding the political situation is important for the global intelligence and security.

For instance, the Advanced Research and Development Activity (ARDA) is the US

intelligence community (IC) center for conducting advanced research and development related to information technology (IT) (see [IC, 2003]). ARDA is a support system for the intelligence analysts and its goal is to improve the reliability of the conclusions of decision makers. One of the projects, in which we are participating, is the Advanced Questions Answering for Intelligence Program (AQUAINT). It is pursuing advanced research for scenario-based, advanced question answering in which, multiple, inter-related questions are asked in a particular topic area by a skilled, professional information analyst, who is attempting to respond to larger, more complex information needs or requirements.

Part of our efforts consists of building models of relevant domains. There are already results in this direction. For instance, the formalization of the travel domain [Gelfond, 2006]. In that work, the author shows the axiomatization of a journey (a movement of a group of objects from one place to another). He outlines a language M for defining knowledge modules and for assembling them into a knowledge base.

In this work, we model a different domain, namely the ideological conflict. It is important for the better understanding of the effects of actions of the countries in the real world. The emphasis of our work is on the development and implementation of a general commonsense knowledge base needed for the axiomatization of the domain **ideological conflict**. For simplicity, we will be mainly interested in the relationship between groups of countries, based on their political ideology at different stages of their history (other type of ideology is the religious one).

Each alliance is formed by several countries, which share the same ideology. In this paper, we view an ideology as a set of ideas central to a society and an ideological conflict as a discord between collections of countries with different ideologies. The following example will be used to illustrate the proposed formalization of knowledge.

Example 4.1. Consider the following story:

In our hypothetical world, depicted in Figure 4.1, there are only ten countries: *US*, *Poland*, *Bulgaria*, *Korea*, *Cuba*, *Russia*, *China*, *Iran*, *Iraq* and *Syria*. We divide them into three ideologically-driven groups, based on our knowledge about the current global political situation: Capitalism (called *freedom alliance*), Communism (or *equality alliance*) and Muslim Radicalism (named *morality alliance*). Assume this unreal situation: at the beginning of the story *US*, *Poland* and *Bulgaria* belong to the first group. On the other hand, *Korea* and *Cuba* belong to the second group. The last one is composed by *Iran*, *Iraq* and *Syria*. *Russia* and *China* do not belong to any of those coalitions.

For simplicity assume that normally, there is an ideological conflict between the alliances *freedom* and *equality*. Imagine that later, after a political change, Bulgaria becomes a communist country; Russia also joins the *equality* alliance; and the US remains in the *freedom* alliance. Consider that an intelligence analyst would like to answer the following simple questions about three of the countries:

1. What is the alliance to which Bulgaria belonged before the change?
2. What is the alliance to which Bulgaria belongs after the change?
3. Is there an ideological conflict at the end of the story between US and Russia?

The expected answers to the first and second questions are *freedom* and *equality*, respectively. Since the new ideology of *Russia* is different from the ideology of *US*, the answer of the third question is *yes*: *Russia* and *US* are in an ideological conflict at the end of the story (see figure 4.2).

To automate this reasoning, we need a language capable of representing the above story as well as expressing defaults, causal relations and other types of commonsense knowledge. Therefore, we use *A-Prolog* (ASP) - a language of logic programs with

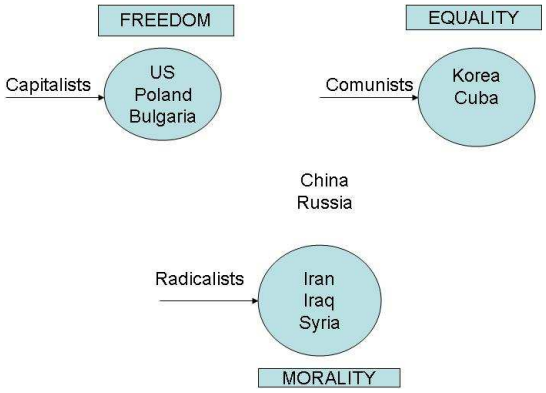

Figure 4.1: At the beginning of the story: Countries divided based on their ideologies

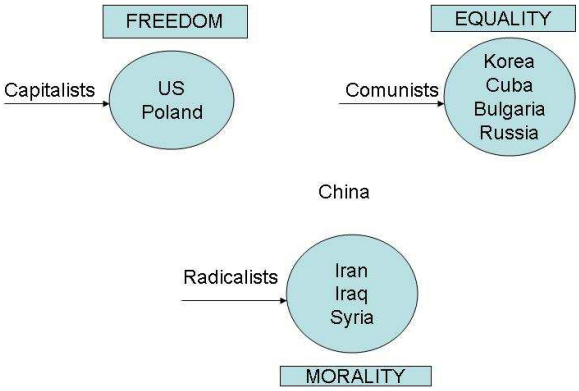

Figure 4.2: At the end of the story: Countries divided based on their ideologies

two negations and disjunction under the answer set semantics [Gelfond and Lifschitz, 1991]. Among the important properties of ASP are its simplicity, expressiveness and the ability to reason with incomplete information. ASP has also a theoretical support and already developed reasoning systems (see for instance [Simons, 2000]). First, we are going to represent our knowledge about ideological conflicts, using action language AL. In Section 4.5, we describe the basic notions of AL.

4.5 Action language AL

The action language AL is divided in three parts: *action description language*, *history description language*, and *query language*. Let review the action description part AL_d first, as described in [Gelfond and Baral, 2000].

The signature, Σ, of AL_d consists of the set F of fluents (statements whose truth depends on time) and the set A of *elementary actions*. A set $\{a_1, \ldots, a_n\}$ of elementary actions is called a *compound* action (where the elementary actions are performed simultaneously). An action description of $AL_d(\Sigma)$ is a collection of propositions of the form:

1. $causes(a_e, l_0, [l_1, ..., l_n])$,
2. $caused(l_0, [l_1, ..., l_n]), and$
3. $impossible\_if(a, [l_1, ..., l_n])$,

where a_e and a are elementary and arbitrary actions respectively and $l_0, \ldots, l_n$ are fluent literals (fluents and their negations) from the signature Σ. An action description A of AL_d defines a transition diagram describing effects of actions on the possible states of the domain.

We now review the language AL_h, which specifies the history of the domain. The past is described by the set Γ of axioms (referred to as *observations*):

1. $happened(a, k)$.

2. $observed(l, k)$,

where $happened(a, k)$ means that action a happened in moment k and $observed(l, k)$ denotes that the fluent l is observed in moment k. A set of axioms defines the collection of paths in a transition diagram. A pair $\langle A, \Gamma \rangle$, where A is an action description and Γ is a set of observations, is called a *domain description*.

Finally, the query language AL_q includes the following queries:

1. $holds\_at(l, t)$.

2. $currently(l)$.

3. $holds\_after(l, [a_n, \ldots, a_1], t)$,

where $hold\_at(l, t)$ means that the fluent l is true in moment t; $currently(l)$ means that the fluent l is currently true; and $holds\_after(l, [a_n, \ldots, a_1], t)$ denotes that l is true in moment t after the actions $a_1, \ldots, a_n$ were performed. There exists a close relationship between AL and logic programming under the answer set semantics, which allows reformulation of the knowledge in ASP [Balduccini and M.Gelfond, 2003].

In fact, the following theorem has been shown in [Gelfond, 2002]:

Theorem 4.1. *For any action a and any state σ, a state σ' is a successor state of a on σ iff there is an answer set S of:*

$$\phi(A) \cup \{h(l, 0) : l \in \sigma\} \cup \{occurs\_at(a_i, 0) : a_i \in a\}$$

such that, $\sigma' = \{l : h(l, 1) \in S\}$ and $\phi(A)$ is the direct encoding of the action description A into ASP.

4.6 Building the knowledge base

We start with the building of the knowledge base, which is used to describe the ideological conflict domain. This domain can be represented by a transition diagram, whose states are sets of fluents and whose arcs are labeled by actions.

Our domain has countries, alliances and ideologies. A country may perform an action of switching to a new ideology and an action of changing from its current alliance to another one. The first action is possible only if the country has different ideology than the one it desires to switch to. The second action is possible only if the country belongs to a different alliance than the one it desires to join.

An ideological conflict may occur between two alliances with different ideologies. Let us construct an action description A of the domain and use it as a starting point to model an ideological conflict in ASP. We start with a description of the signature.

The fluent $belong(C, A)$ means that a country C belongs to an alliance A. The value of this fluent may be changed with a dynamic causal law, which says that if a country changes to an alliance, it will belong to that new alliance. In addition, a static law guarantees that a country belongs to only one alliance at the same time. The fluent $has(A, I)$ stands for: an alliance A has an ideology I. It is defined by a static causal law, which says that an alliance can have only one ideology at the same time. The predicate $in\_conf(A1, A2)$ says that two alliances are in conflict. There is a similar law for the symmetry of this relationship. Finally, the fluent $conflict(C1, C2)$ means that country $C1$ and country $C2$ are in conflict. It is changed by the static causal law, which says that if two alliances are in conflict, then the countries that belong to these alliances, will be in conflict, respectively. Below is the action description.

Types:

```
country(C).

ideology(I).

alliance(A).
```

Fluents:

```
fluent(belong(C,A)).

fluent(has(A,I)).

fluent(conflict(C1,C2)).
```

Actions:

```
action(change(C,A)).
```

Causal Laws:

```
impossible change(C,A) if belong(C,A).


change(C,A) causes belong(C,A).


caused -belong(C,A1) if belong(C,A2),
                        A1 != A2.


caused -has(A,I1) if has(A,I2),
                     I1 != I2.


caused conflict(C1,C2) if in_conf(A1,A2),
                          belong(C1,A1),
                          belong(C2,A2),
                          A1 != A2.
```

```
caused in_conf(A1,A2) if in_conf(A2,A1),
                         A1 != A2.
```

Initially:

```
normally in_conf(equality,freedom).
```

In the following two sections, we represent the general knowledge about ideologies, effects of actions and situation in our hypothetical world (Section 4.7), as well as the particular story (Section 4.8).

4.7 ASP representation of the knowledge base

This is the first part of our knowledge base, namely, the representation of general knowledge of ideologies and conflicts.

The following is a representation of the domain description in AL. We use the syntax of the ASP's inference engine Smodels [Simons, 2000]. In order to save place, we use $h(F,T)$ instead of $holds(F,T)$ and $o(A,T)$ instead of $occurs(A,T)$. The first one is used to say that a fluent F holds (or is true) at a given point of time T. The second one means that an action A occurs at some moment T. We assume that the reader is familiar with Smodels. However, we want to explain that the code is documented. Before each rule, there is a comment.

```
#const n=1.
time(0..n).
#domain country(C;C1;C2).
#domain ideology(I;I1;I2).
#domain alliance(A;A1;A2).
```

```
#domain time(T).
#domain fluent(FL;FL1;FL2).

% country C belong to alliance A
fluent(belong(C,A)).

% alliance A has ideology I
fluent(has(A,I)).

% country C1 is in
% conflict with country C2
fluent(conflict(C1,C2)).

% country C changes to
% a new alliance A
action(change(C,A)).

% symmetry

in_conf(A1,A2):- in_conf(A2,A1),
                      A1!=A2.

-in_conf(A1,A2):- -in_conf(A2,A1),
                  A1!=A2.

% impossible to change to an alliance,
```

```
% if already belongs to it

:-o(change(C,A),T),h(belong(C,A),T).

% if a country changes to a
% new alliance,it will belong to it

h(belong(C,A),T+1):-o(change(C,A),T).

% a country can belong
% only to one alliance,
% at the same time

-h(belong(C,A1),T):-
            h(belong(C,A2),T),
            A1!=A2.

% an alliance can have
% only one ideology,
% at the same time

-h(has(A,I1),T):-
            h(has(A,I2),T),
            I1!=I2.

% two countries are in conflict,
```

```
% if they belong to alliances
% in conflict


h(conflict(C1,C2),T):-
                  in_conf(A1,A2),
                  h(belong(C1,A1),T),
                  h(belong(C2,A2),T),
                  C1!=C2,
                  A1!=A2.
% DEFAULT: Normally equality and freedom
% alliances are in conflict, if there
% is no evidence of the opposite


in_conf(equality,freedom) :-
       not -in_conf(equality,freedom).
```

Now we state the domain independent part, which may be used with other domains. We define the inertia rule (*normally actions do not affect fluents*) in this part.

```
% INERTIA RULE


h(FL,T+1) :- T<n, h(FL,T),
             not -h(FL,T+1).
-h(FL,T+1) :- T<n, -h(FL,T),
              not h(FL,T+1).
```

4.8 Formalizing the story

This is the second part of our knowledge base, i.e., the representation of the particular story. The general knowledge base, created previously, can be used in different stories containing information about ideological conflicts. To illustrate this, let us consider again our main story from Section 4.4 and present a logical representation of it.

```
country(us).
country(bulgaria).
country(russia).

alliance(freedom).
alliance(equality).
alliance(none).

ideology(capitalism).
ideology(communism).
ideology(other).

% HISTORY OF THE DOMAIN

h(belong(bulgaria,freedom),0).
h(belong(russia,none),0).
h(belong(us,freedom),0).

h(has(freedom,capitalism),0).
h(has(equality,communism),0).
```

```
h(has(none,other),0).


o(change(bulgaria,equality),0).
o(change(russia,equality),0).

% the equality and the freedom
% alliance are not in conflict
% with the none alliance of the
% non-allied countries


-in_conf(equality,none).
-in_conf(freedom,none).
```

The program is implemented in *Smodels* and it runs with the following command:

```
lparse --true-negation conflict.sm |
smodels 0| mkatoms
```

It is useful to download the output formatting program *mkatoms* from:

```
http://krlab.cs.ttu.edu/~marcy/mkatoms
```

As expected, in the output of this program, we have that, before the change, Bulgaria is in the $freedom$ alliance; after the change Bulgaria is in the *equality* alliance; and at the end of the story Russia and US are in conflict.

```
h(belong(bulgaria,freedom),0)
h(belong(bulgaria,equality),1)
h(conflict(russia,us),1)
```

We realize that *Smodels* is sufficient for our purpose. We did not experimented with a bigger example, but we believe that *Smodels* has the basics for representation of common sense. For instance, the choice rule can be replaced with a less sophisticated rule.

4.9 Closing remarks

In this chapter, we first proposed a methodology to follow when dealing with commonsense knowledge representation and reasoning. We automated a particular example by constructing a knowledge base consisting of general rules and rules of a particular story. This is a simple version of what is going to be a formalization of ideological conflicts. We illustrated our methodology of using ASP and *AL* for formalizing commonsense knowledge in a different domain from what has been done before. In the past, many researchers have worked with knowledge representation. However, we believe that an ideological conflict may lead to a war of ideology between countries, where one country is trying to impose a new ideology on the other country, which is a global situation that need to be solved. Currently, we are working on the expansion of the general knowledge base. Our future work is to extend our commonsense knowledge base with knowledge necessary to reason about more countries and more complex stories. The ideal goal is to have one general knowledge base, containing common senses and expert knowledge about various domains.