Chapter 3

Commonsense reasoning

For long time ago, researchers have been study the theoretical parts of how to represent commonsense knowledge. However, nobody has compared different opinions of what is in fact the commonsense knowledge. This research area is new and it needs our attention. The commonsense reasoning has been studied in different areas, such as philosophy, psychology, and computer science. As part of our research, we studies different views of what is commonsense knowledge and reasoning. At the end of this chapter, we suggest four important definitions for our research. We start with the notions of common sense as part of "folk psychology". According to *Wikipedia Online Dictionary*, "folk psychology" (sometimes called naive psychology or common sense psychology) is the set of background assumptions, socially-conditioned prejudices and convictions that are implicit in our everyday descriptions of others' behavior and in our ascriptions of their mental states. We also discuss how some of the knowledge representation researchers in different Computer Science fields describe commonsense reasoning, as well as default reasoning and non-monotonic reasoning.

3.1 Issues in commonsense reasoning

In her book, “Common sense, Reasoning, and Rationality”, Renée Elio describes common sense as found in “folk psychology” area: “sound and prudent but often unsophisticated judgment” and “the unreflective opinions of ordinary [persons].” Although from this definition common sense is characterized as *unsophisticated*, it is a good feature that most ordinary people possess [R.Elio, 2002]. Moreover, there are evidences that most of the people are very good at commonsense reasoning in an uncertain and complex everyday world. However, there is still no generally accepted formalism and computational method that enable us to build systems that achieve the same commonsense reasoning as humans do [Elio, 2002].

In addition, Renée Elio talks about default reasoning, which occurs very often during our commonsense reasoning. In [Elio and F.J.Pelletier, 1993], she states that default reasoning occurs when the available information does not deductively guarantee the truth of the conclusion and the conclusion is nonetheless correctly arrived at. A lot of researchers have characterized formally the default reasoning. In particular, Vladimir Lifschitz in his paper [V.Lifschitz, 1989] published a list of 25 ”Nonmonotonic Benchmark Problems” which gave the answers to many doubts of the researchers in the area. Lately, all formal accounts of nonmonotonic reasoning were supposed to be able to yield these answers. In his list of problems, Lifschitz divides the problems in different areas. For instance, *Basic Default Inference* problems and the *Inheritance Inference* problems. Elio decided to investigate what sorts of default reasoning ordinary people in fact employ. She took the benchmark problems of Lifschitz and observe the people’s performance. The conclusion is that people’s plausible conclusions about defaults and exceptions are influenced by differences in the amount of information available about the objects, by the specificity of information about the exception, and by the apparent similarity between objects that might be governed by the same rules. This result suggests that the non-monotonic community should study the issue of what kind of information is relevant to the application of a default rule [Elio and F.J.Pelletier, 1993].

Reneée Elio and her research group, investigate the relation of commonsense and philosophy. In the early 18^{th} century, psychologism emerged within the field of philosophy. As defined in *Wikipedia Online Dictionary*, psychologism is a generic type of position in philosophy according to which psychology plays a central role in grounding or explaining some other, non-psychological type of fact or law. In our case, there is a belief, that human psychology is useful for the developing of computational models with commonsense features [Elio, 2002]. Let go back to the characteristics of commonsense in order to understand better how we can achieve our goal (to make the computers use commonsense reasoning).

Mueller says that commonsense reasoning is the sort of reasoning we all perform about the everyday world. Commonsense reasoning is very complex and it requires a large amount of knowledge about the world and the ability to use that knowledge. Mueller defines commonsense reasoning as a process that involves taking information about certain aspects of a scenario in the world and making inferences about other aspects of the scenario based on our commonsense knowledge [Mueller, 2006]. Although Mueller emphasizes on commonsense reasoning, he also mentions that we have to construct representations of commonsense knowledge that can be used be the reasoning method. Moreover, John McCarthy [McCarthy, 1984] distinguishes between commonsense knowledge (what everyone knows) and commonsense reasoning (the human ability to use commonsense knowledge). Thus, the reader may ask: What are ordinary people reasoning *about*? And what is the *means* by which they do so? The answer of both questions is in the following sections.

3.1.1 The domains of commonsense knowledge

Some years ago, McCarthy [McCarthy, 1984] listed the topics relevant to commonsense knowledge that still need research in the field of knowledge representation and reasoning. The topics include time and space; causality; approximate or qualitative theories of motion, force, substances, and energy; continuous change; and quantities. In summary, what "everyone" knows is mainly the understanding of changes, actions, cause, and effect in the physical world. In addition, the domain of commonsense knowledge includes folk psychology concepts like goals, beliefs, and desires. The commonsense knowledge that people share help them to describe, predict and explain everyday events. Our knowledge in these fields is well established even in the early years of our infancy. Moreover, the acquisition of concepts like causality, the persistence of objects and properties over time, and even a theory of mind are main concerns of developmental psychology. As defined in *Wikipedia Online Dictionary*, development psychology seeks to understand how people come to perceive, understand, and act within the world and how these processes change as they age. It mainly focuses on the development of the human mind through the life span.

We can conclude that the domain of commonsense knowledge is defined by exclusion, contrasting it with "highly structured domains" such as mathematics and other subjects about which we receive direct instruction as part of our formal education. Commonsense knowledge is what people come to know in the process of growing and living in the world [R.Elio, 2002]. John McCarthy [McCarthy, 1990a] says that commonsense knowledge includes the basic facts about events and their effects, facts about knowledge and how it is obtained, facts about beliefs and desires. In addition, it includes the basic facts about material objects and their properties.

3.1.2 Reasons to emphasize commonsense knowledge

One may ask, why researchers need to emphasize commonsense knowledge rather than the knowledge contained in scientific theories (or specialized knowledge). John McCarthy [McCarthy, 1990a] gives three reasons to support the study of commonsense knowledge:

- Scientific theories represent knowledge divided up into categories. That is, when developing and presenting a scientific theory, there is a commonsense pre-scientific stage. In this part of the process, it is decided what phenomena are to be studied and how the formal terms relate to the commonsense world. In sum, to use science, common sense is required.
- Commonsense reasoning is required for solving problems in the commonsense world. The explanation is that the commonsense world is characterized by a different $informatic\ situation$ than that *within* any formal scientific theory. In the $commonsense\ world\ informatic\ situation$, the reasoner does not know what facts are relevant to solving his problem. Some unexpected events may arise that involve using parts of the knowledge thought to be not relevant previously.
- The informal metatheory of any scientific theory has a commonsense informatic character. For example, mathematicians invented the notion of a group in order to make parallels between different domains. The thinking involved in this process has a commonsense character.

3.1.3 The nature of commonsense reasoning

Our ultimate goal is to make computer programs reason about the world [McCarthy, 1959]. After the commonsense knowledge domain has been defined, the next step toward the goal might be:

- write down (axiomatize) what everyone knows about some domain (for example, space shuttles), and
- apply a formal proof procedure for this axiomatization as the machinery of reasoning.

However, two immediate matters arise. First, there is any coherent "independent-of-use" content base for commonsense knowledge that could be itemized, axiomatized, and then used by any algorithm for any purpose. Second, even if we imagined we could write down "independent-of-purpose" axioms that represented a subdomain of commonsense knowledge, we could not write it *all* down [R.Elio, 2002].

Consider, for example, a simple commonsense problem: what knowledge is used in formulating a simple plan to buy an airplane ticket, with the intention to go to a different country? We could prove such a plan would succeed only if we knew at least the following: that the ticket is bought for the desired country (and not elsewhere); that the depart date and the arrival date are exactly as wanted; that the person arrive at the airport on time; and the airplane itself is not damaged. Furthermore, we must know that a person cannot be at two place at once, that is, a person cannot be both in the airport and in the house. In order to list what is known and assumed, we use both specific knowledge of the domain (travel) and general understanding of time, materials and forces. The question is when do we stop writing down all specifications relevant to this problem?

In addition, very often people do not have complete knowledge of a situation and they tend to make default assumptions. Those assumptions are about what is usually or typically true and not on facts known to be true with certainty. But some problems may arise when trying to use default assumptions in reasoning. After all, the reasoner may have different default assumptions that in turn can lead to inconsistent conclusions. For example, I might believe these two rules: Normally, international graduate students do not attend the recreational sport center, and normally, students attend every activity in the recreational sport center. If in a particular situation, both assumptions apply to the same student, then I would have a contradictory belief. This inconsistency leads to another problem, which is the matter of formalizing how to decide which of the beliefs to adopt. In summary, new information may arrive and contradict with the accepted beliefs. In order to deal with that situation, the reasoner may have to reject some of his previous believes in order to have a consistent model. Thus, we can conclude that, much of the everyday, commonsense reasoning has *non-monotonic* features: the conclusions based on old beliefs may no longer be valid with the arrival of new information [Elio, 2002]. We talk more about *non-monotonic* reasoning in the following section.

3.2 Uses of non-monotonic reasoning

Formalized non-monotonic reasoning was first proposed in the late 1970s. Mathematical logic, on the other hand, is monotonic, that is: if we have $A \vdash p$ and $A \subset B$, then we also have $B \vdash p$. However, while much human reasoning is monotonic, some human commonsense reasoning is not. This is the reason why the study of non-monotonic reasoning is important [McCarthy, 1990a]. Non-monotonic reasoning has several uses as described in [McCarthy, 1990b]:

- As a communication convention.
- As a database or information storage convention.
- As a rule of conjecture.
- As a representation of a policy.
- As a very streamlined expression of probabilistic information when numerical prob-

abilities are unobtainable.

- Auto-epistemic reasoning.
- Both common sense physics and common sense psychology use non-monotonic rules.

3.3 Properties of commonsense reasoning

In this section, we describe with more details, what is a commonsense reasoning. Every day of our lives, we perform commonsense reasoning. We can easily predict that, if a child comes back to home, then after that, he/she will be in the home. If the same child walks into the house with a tennis ball, then the ball will be in the home also. Humans can make conclusions like these very easily, but commonsense reasoning is not a simple task. Indeed, it is a very complex one.

In order to reason about the world, we need to have enough knowledge about the world and the ability to use that knowledge. Normally, we have knowledge about things around us, such as objects, actions, time and space and we can use that knowledge to predict what is going to happen at the next moment. That is, we can observe things and plan actions.

We want to make computers perform a commonsense reasoning in the same way as we do. As a result, we need to automate the commonsense reasoning. One approach is to use logic to accomplish this task. For example, ASP is a formalism, that can be used to reason about different scenarios, which involve common sense.

As defined in [Mueller, 2006], *commonsense reasoning* is a process that involves taking information about certain aspects of a scenario in the world and making inferences about other aspects of the scenario based on our *commonsense knowledge*, or knowledge of how the world works. No matter which method is chosen for automated commonsense reasoning, it should contain the following: [Mueller, 2006]

- *Representation*. The method must represent scenarios in the world and must represent commonsense knowledge about the world.
- *Commonsense entities*. The method must represent objects, agents, properties, events, and time.
- *Commonsense domains*. The method must represent and reason about time, space, and mental states.
- *Commonsense phenomena*. The method must address the commonsense law of inertia, concurrent events, indirect effects, preconditions, and triggered events.
- *Reasoning*. The method must support default reasoning, temporal projection, abduction, and postdiction.

3.3.1 Types of reasoning

There are different types of reasoning: deduction, abduction, postdiction, and model finding. *Deduction* or *temporal projection* consists of finding the state that is obtained from doing a sequence of actions. *Abduction* is finding what events might lead from an initial state to a final state. *Planning* is to create a sequence of actions to achieve a final state, called *goal*, beginning with the initial state. *Postdiction* is finding the initial state given a sequence of events and a final state. Finally, *model finding* consists of creating models from states and events [Mueller, 2006].

3.4 Artificial Intelligence and common sense

The long term goal of Artificial Intelligence (AI) is humanlevel AI. That means the creation of computer programs with at least the intellectual capabilities of humans. As explained in [McCarthy, 2003], there are two approaches to seeking this objective.

- *The biological approach*: Builds agents that imitate characteristics of the physiology or psychology of humans.
- *The engineering approaches:* Regards the world as presenting certain kinds of problems to an agent trying to survive and achieve goals.
  - *The logical approach:* It is a variety of the engineering approach. Represents what is known in logical formulas and infers that certain actions or strategies are appropriate to achieve its goals.

John McCarthy realizes that the logical agents of the next 49 years need at least:

- Continued existence over time
- Improved ability to reason about action and change
- More elaboration tolerant formalisms
- The ability to represent and reason about approximately defined entities
- Enough selfawareness and introspection to learn from the successes and failures of their previous reasoning
- Domain-dependent control of theorem provers and problem solvers
- *Identifying the most basic commonsense knowledge and getting it into the computer*

McCarthy also mention that the logical approach is ahead. That is, it has faced and partly solved problems such as identifying and representing as data and programs information about how the world works. The logical approach has the advantage that when we achieve humanlevel AI, we will be able to explain how intelligence works. On the other hand, some of the evolutionary approaches might achieve an intelligent computer without understanding how it works [McCarthy, 2003]. Finally, the relation between AI and commonsense knowledge is the following:

> No one can convincingly say that given a billion dollars he could reach humanlevel. A critical level of AI will be reached, as Douglas Lenat pointed out, when an AI system has enough basic common sense information to be able to get more information by reading books [McCarthy, 2003].

3.5 Some quotations

In this section, we present some quotations, previously given in [F.J.Pelletier, 1997].

McDermott and Doyle 1980:

> In artificial intelligence, studies of...common sense have led to knowledge representations which explicitly and implicitly embody much information about typical cases, defaults, and methods for handling mistakes...[One problem] is the problem of maintaining a set of facts which, although expressed as universally true, have exceptions. ...Such...cases include many forms of inferences, default assumptions, and observations.

Lifschitz 1985:

> Research in the theory of commonsense reasoning has revealed a fundamental difference between how universal assertions are used in mathematics on the one hand, and in the area of commonsense knowledge on the other. In mathematics, when a proposition is claimed to be universally true, the assertion includes a complete list of conditions on the objects involved under which the proposition is asserted. But in everyday life we often assert that a certain proposition is true “in general”; we know that there are exceptions, and we can list some of them, but the list of exceptions is not a part of the assertion. ...The language of predicate logic has been created primarily for the purpose of formalizing mathematics. ...If we want to use that language for representing commonsense knowledge then methods for formalizing assertions about exceptions have to be developed. The study of such methods belongs to the area of non-monotonic logic.

Moore 1985:

> It has been generally acknowledged in recent years that one important feature of ordinary commonsense reasoning that standard logics fail to capture is its non-monotonicity. ...Autoepistemic logic is intended to model the beliefs of an agent reflecting upon his own beliefs. ...We are trying to model the beliefs of a rational agent.

McCarthy 1986:

> Our long-term goal...is to express these ["common-sense"] facts in a way that would be suitable for inclusion in a general-purpose database of common-sense knowledge. ...Common-sense knowledge must be represented in a way that is not specific to a particular application. ...Both common-sense physics and common-sense psychology use non-monotonic rules. An object will continue in a straight line if nothing interferes with it. A person will eat when hungry unless something prevents it.

Luger and Stubblefield 1989:

> Non-monotonicity is an important feature of human problem solving and common-sense reasoning.

Tanimoto 1990:

> In everyday life, people seem to reason in ways that do not adhere to a monotonic structure. For example, consider the following: "Helen was attending a party..." Here Helen has performed non-monotonic reasoning. ...There is good reason for people to employ such non-monotonic reasoning processes. We often need to jump to conclusions in order to make plans, to survive; and yet we cannot anticipate all of the possible things that could go wrong with our plans or predictions. We must make assumptions about things we don't specifically know. Default attributes are a powerful kind of knowledge, since they permit useful conclusions to be made, even if those conclusions must sometimes be revoked.

Brewka 1991:

> To formalize human commonsense reasoning something different [from classical logic] is needed. Commonsense reasoning is frequently not monotonic. In many situations we draw conclusions which are given up in the light of further information.

Ginsberg 1987, Intro.; also Ginsberg 1993:

> A key property of intelligence-whether exhibited by man or by machine-is flexibility. This flexibility is intimately connected with the defeasible nature of commonsense inference...we are all capable of drawing conclusions, acting on them, and then retracting them if necessary in the face of new evidence. If our computer programs are to act intelligently, they will need to be similarly flexible.

Kraus et al. 1990:

> In everyday life, it seems clear that we, human beings, draw sensible conclusions from what we know and that, in the face of new information, we often have to take back previous conclusions, even when the new information we gathered in no way makes us want to take back our previous assumptions. ...It is most probable that intelligent automated systems will have to perform the same kind of (non-monotonic) inferences.

Reiter 1978:

> Most of what we know about the world, when formalized, will yield an incomplete theory precisely because we cannot know everything-there are gaps in our knowledge. The effect of a default rule is to implicitly fill in some of those gaps by a form of plausible reasoning...Default reasoning may well be the rule, rather than the exception, in reasoning about the world since normally we must act in the presence of incomplete knowledge... Moreover, ... most of what we know about the world has associated exceptions and caveats.

Gabbay 1993:

> [The perfect paper on the topic would] build a complex formal model for describing human practical reasoning interactions, show the model is plausible and highlight the role of nonmonotonicity in such a model. The emphasis [would] not [be] on the actual practical modeling of human reasoning (which is a major serious task), but on building enough of a model to distinguish the (what is currently known as) nonmonotonic component from other (human reasoning) components.

From all these quotations, we can conclude that the traditional classical logic is not enough to describe human commonsense reasoning. We need the ability to employ default reasoning, because human behavior has non-monotonic flexibility. In the next section we explain again why Answer Set Programming is a good language for formalizing commonsense knowledge and reasoning.

3.6 Common sense and Answer Set Programming

Computer scientists create elegant and powerful reasoning methods. However, the importance of these methods pales in comparison with the importance of the body of domain knowledge. In order to explain this principle, Edward Feigenbaum gives the following example:

> As I write, I can look out my window and see the Stanford Mathematics Department on my left and the Stanford Medical School on my right. In the Math building are some of the most powerful reasoners in the world, at least one of whom has won the Fields Medal. If I were, at this moment, to have a threatening medical event, I would ask my CS colleagues to rush me to the Medical School, not the Math Department. The powerful reasoning of the mathematicians would be essentially totally ineffective compared with the knowledge of medicine of the doctors in the Emergency Room of the Medical School's hospital. For the doctors to apply their knowledge effectively to my case will require some simple reasoning, but not logically powerful and elegant methods [Feigenbaum, 2003].

The previous example emphasizes the fact that there are many formalisms for knowledge representation. However, we need one, which is not too sophisticated. A-Prolog is such a declarative language suitable for commonsense reasoning. We first explain the difference between algorithmic and declarative languages.

3.6.1 Algorithmic versus declarative languages

There are two types of languages: algorithmic (or procedural) and declarative. Programs written in algorithmic languages describe sequences of actions for a computer to perform. That is, the designer explicitly communicates to the computer "how" to accomplish the desired task. On the other side, declarative programs are collections of statements, which describe objects of a domain and their properties. In this case, the programmer communicates to the machine "what" has to be done. The semantic of a declarative program Π is normally given by defining its models, which are possible states of the world compatible with Π. The statements which are true in all such models constitute the set of valid consequences of Π. In fact, declarative programming consists in representing knowledge, about the domain relevant to the programmes goals, by a program Π and in reducing various programming tasks to finding models or computing consequences of Π. Usually, the models are found and/or consequences are computed by $inference\ engines$. There are two important requirements, which should be satisfied by a declarative programming language:

- A declarative language should allow construction of *elaboration tolerant* knowledge bases. This is, the bases in which small changes in the informal body of knowledge correspond to small changes of the formal base representing this knowledge. The requirement can be satisfied if we use *non-monotonic* languages. This property is very important for representing commonsense knowledge about the world. As we discussed earlier, in commonsense reasoning, additions to the agents knowledge are common and sometimes inferences are based on the absence of knowledge. Modeling such reasoning in *non-monotonic* languages seems to lead to simpler and more elaboration tolerant representations.
- The inference engines associated with the language should be sufficiently general and efficient.

However, it may not be possible to design a knowledge representation language suitable for *all* possible domains and problems [Gelfond, 2002].

3.6.2 A-Prolog

There are different mathematical approaches for knowledge representation. However, A-Prolog is the most accepted in the society of computer scientists. It is a language of logic programs under the answer set semantics. It can be viewed as a declarative language with roots in logic programming, syntax and semantics of standard Prolog, and in the work on non-monotonic logic. However, Prolog has many extra logical features, which makes it both declarative and procedural. Circumscription is a rule of conjecture that can be used by a person or program for jumping to certain conclusions. That is, the objects that can be shown to have a certain property P by reasoning from certain facts A are all the objects that satisfy P. Circumscription is non monotonic and can formalize common sense reasoning [McCarthy, 1980]. However, circumscription is too complex for our needs. Similar situation occurs with the logic for default reasoning proposed by Reiter [Reiter, 1980]. Other complex non-monotonic logics exist such as the non-monotonic logic I [McDermott and Doyle, 1980]. On the other hand, if we look at the classical logic, we distinguish its monotonic characteristics, which is not desirable for commonsense knowledge representation. Therefore, A-Prolog differs from other knowledge representation languages by its simplicity and its ability to represent $defaults$ (statements of the form "Elements of a class C normally satisfy property P"). As we mentioned before, during our education, we try to learn defaults, exceptions to those defaults and ways to use that information to conclude facts about the world. In fact, A-Prolog has very well defined theoretical foundations. Its syntax allows to represent defaults and their exceptions in a very elegant way; its consequence relation characterizes the corresponding set of valid conclusions; and its inference mechanisms allow a program to find these conclusions in a "reasonable" amount of time. For the purpose of our investigation, its ability to model commonsense reasoning in a suitable way is the most important one [Gelfond, 2002]. In continuation, we present more comparisons between ASP and other formalisms.

3.7 Comparison with other formalisms

In this section, we present different logic formalisms, which might be used for knowledge representation and reasoning, but we make an immediate comparison with ASP, in order to highlight the reason of our choice.

3.7.1 ASP versus PROLOG

The name PROLOG is ambiguous. It was originally intended as the name for the programming language developed by Alain Colmerauer and Phillipe Roussel in the summer of 1972 [Kowalski, 1988]. The programming language PROLOG (an acronym from PROgramming in LOGic) came as a result of the conclusion of Kowalski and Colmerauer that logic can be used as a programming logic [Colmerauer and Roussel, 1992]. Horn clauses are a subset of first-order logic and they allow fast and simple inferencing through resolution. Although PROLOG deals with Horn clauses, it has also several nondeclarative features, which were added to make it more programmer friendly. We consider ASP as a declarative alternative of PROLOG [Baral, 2003].

The main differences between ASP and PROLOG are listed bellow.

1. The ordering of the literals in the body of a rule matters in PROLOG, because it processes them in a left-to-right fashion. In ASP, the positioning of the literals does not matter. The body of each rule is a set of literals and those preceded by not.

2. The positioning of a rule matters in PROLOG, because it processes them in top-down fashion. In ASP, the ordering of rules does not matter. An ASP program is a set of ASP rules.

3. In PROLOG, queries are processed in top-down fashion, from queries to facts. In

ASP, the query-processing methodology is not part of the semantics. Moreover, most sound and complete interpreters do bottom-up query processing, starting from the facts and going to the conclusions.

4. Because of the characteristics mentioned above, a PROLOG program may get into an infinite loop for even small programs, which is not the case with ASP.
5. The PROLOG operator *cut* does not exist in ASP. It is extra-logical.
6. Getting stuck in a loop and floundering are problems in PROLOG, which are due to the way it deals with negation as failure. ASP does not have these problems, because it uses the answer set semantics to characterize negation as failure [Baral, 2003].

3.7.2 ASP versus Logic programming

ASP is a type of logic programming, where the semantics are fixed to *answer set semantics*. Other alternative semantics are the *stable model semantics* and the $well-founded\ semantics$. The stable model semantics was created by Michael Gelfond and Vladimir Lifschitz in 1988 [Gelfond and Lifschitz, 1988]. The stable models coincide with the answer sets in the case of programs with one atom in the head and without classical negation in the body. On the other hand, the well-founded semantics differs from the stable model semantics in the following:

1. Stable models are two valued (true or false), while well-founded are three valued (true, false, and unknown).
2. Every program under the well-founded semantics has only one model, while some programs under the answer set semantics have multiple stable models and some do not have any.

3. Computing the entailment with respect to the well-founded model is more tractable than computing it with respect to the stable models. However, the latter increases the expressive power of the language.

The multiple stable models or the absence of stable models has its advantages. On one side, the multiple models are useful in enumerating choices (used in planning and in formalizing aggregation). On the other side, the absence of stable models is useful in formulating integrity constraints, whose violation forces elimination of models [Baral, 2003].

3.7.3 ASP versus Default logic

Default logic was created by R.Reiter in 1980 [Reiter, 1980]. Default logic is a pair (W, D), where W is a first-order theory and D is a collection of defaults. It has been shown that ASP and default logic have the same expressiveness. Moreover, ASP is syntactically simpler than default logic and yet has the same expressiveness. Therefore, ASP is more usable [Baral, 2003].

3.7.4 ASP versus Circumscription and Classical logic

Circumscription is a non-monotonic logic created by John McCarthy [McCarthy, 1980] to formalize the commonsense assumption that things are as expected unless otherwise specified. As we mentioned in the beginning of Section 2.2, the connective "←" and the negation as failure operator "*not*" in ASP add structure to the ASP program. For instance, the ASP rule $a \leftarrow b$ is different from the classical logic formula $b \supset a$, and the connective "←" divides the rules of an ASP program into a head and a body (the left-hand side and the right-hand side, respectively). This structure allows us to define different syntactic notions. For example, $splitting$, $stratification$, $signing$, etc. Given these sub-classes, we can study their properties, such as:

- consistency
- coherence
- complexity
- expressiveness
- filter-abducibility
- compilability to classical logic

In addition, the non-classical operator “$\leftarrow$” gives a directionality, which help us to express *causality*. The latter one can not be encoded in classical logic in a direct way [Baral, 2003].

3.8 Discussion

In this chapter we analyzed important concepts of common sense. As a result of our study of commonsense knowledge and reasoning, we propose the following definitions as the most significative for our purpose:

- A program has common sense if it automatically deduces for itself a sufficiently wide class of immediate consequences of anything it is told and what it already knows [McCarthy, 1959].
- Common sense is a sound and prudent but often unsophisticated judgement and the unreflective opinions of ordinary persons [Elio, 2002].
- Commonsense reasoning is a process that involves taking information about certain aspects of a scenario in the world and making inferences about other aspects of

the scenario based on our commonsense knowledge, or knowledge of how the world works [Mueller, 2006].

- Commonsense reasoning is non-monotonic.

The previous definitions say that we need to be able to express the non-monotonicity. In addition, we have concluded that the commonsense reasoning is not sophisticated. That is, we need neither a simple logic, nor a complicated one. We compared ASP with other logical formalisms, in order to highlight its characteristics as a language suitable for commonsense knowledge representation. We concluded that A-Prolog is ideal for our purposes. After we have chosen the language to express common sense, we need to know how to do it? In Chapter 4, we introduce a methodology to follow.