

# Prioritized Reasoning in Logic Programming

L. Caroprese and I. Trubitsyna and E. Zumpano

DEIS, University of Calabria  
Rende, Italy 87030

email: {caroprese, irina, zumpano}@deis.unical.it

## Abstract

This work addresses the issue of prioritized reasoning in the context of logic programming. The case of preference conditions involving atoms is considered and a refinement of the comparison method of the ASO semantics is presented. The paper introduces the concept of “choice”, as a set of preference rules describing common choice options in different contexts and proposes a new semantics that looks beyond the single preference rules, considering them as a tool for choice representation. More specifically, given a preference rule, its head atoms correspond to the *choice options*, whereas its body specifies the *choice context*, i.e. the decisions which have to precede this choice. Finally, a system, called CHOPPER, able to realize both the ASO semantics and its extensions is proposed.

## Introduction

Logic programming can be used for expressing and solving a large variety of AI problems. The use of preference rules in logic programming gives a powerful tool for expressing desiderata on final solutions in order to establish the best ones. The most common form of preference consists in specifying preference conditions among rules, whereas, some recent proposals admit the expression of preference relations among atoms. See (?) for a survey on this topic. This work contributes to realizing prioritized reasoning in logic programming in the presence of preference conditions involving atoms. This topic has been investigated in (?) and (?), proposing the ASO and PLP semantics, respectively. The PLP semantics considers only common preferences in order to compare two solutions, whereas the ASO semantics evaluates the degree of satisfaction of all preference rules to determine the preferred answer sets and can establish preference relation between two solutions directly.

**Example 1** Consider the prioritized program  $\langle \mathcal{P}_1, \Phi_1 \rangle$ .  $\mathcal{P}_1$  describes possible menus by means of three rules  $r_1, r_2$  and  $r_3$ , where  $\oplus$  states that exactly one of the head’s atoms has to be taken in the answer set, and three constraints  $c_1, c_2$  and  $c_3$ , i.e. rules with empty heads satisfied if the body is false, (e.g.  $c_1$  states that *beef* and *red* cannot be simultaneously present). The preference rules  $\varrho_1, \varrho_2$  and  $\varrho_3$  state

respectively that (i) *white* is better than *red* in the presence of *fish*; (ii) *red* is better than *white* in the presence of *beef*; and (iii) *pie* is better than *ice-cream*.

$$\begin{array}{ll} r_1 : \text{fish} \oplus \text{beef} \leftarrow & c_1 : \leftarrow \text{beef}, \text{red} \\ r_2 : \text{red} \oplus \text{white} \leftarrow & c_2 : \leftarrow \text{beef}, \text{pie} \\ r_3 : \text{pie} \oplus \text{ice-cream} \leftarrow & c_3 : \leftarrow \text{fish}, \text{white} \\ \varrho_1 : \text{white} > \text{red} \leftarrow \text{fish} & \\ \varrho_2 : \text{red} > \text{white} \leftarrow \text{beef} & \\ \varrho_3 : \text{pie} > \text{ice-cream} \leftarrow & \end{array}$$

$\mathcal{P}_1$  has three answer sets (solutions)  $S_1 = \{\text{fish, red, pie}\}$ ,  $S_2 = \{\text{fish, red, ice-cream}\}$ ,  $S_3 = \{\text{beef, white, ice-cream}\}$ .  $\square$

The ASO semantics associates to each answer set  $S$  a *satisfaction vector*  $V_s = (v_s(\varrho_1), \dots, v_s(\varrho_n))$ , where: a)  $v_s(\varrho_j) = I$ , if  $\varrho_j$  is *Irrelevant* to  $S$ , i.e. (i) the body of  $\varrho_j$  is not satisfied in  $S$  or (ii) the body of  $\varrho_j$  is satisfied, but none of the  $C_i$ ’s is satisfied in  $S$ ; b)  $v_s(\varrho_j) = \min\{i : S \models C_i\}$ , otherwise.

For the previous example, the satisfaction vectors are:  $V_{S_1} = (2, I, 1)$ ,  $V_{S_2} = (2, I, 2)$ ,  $V_{S_3} = (I, 2, 2)$ .

The satisfaction vectors are then used to compare the answer sets under the assumption that  $I$  is equal to 1 (i.e.  $v_{S_j}(\varrho_i) = I$  is equivalent to  $v_{S_j}(\varrho_i) = 1$ ).

More specifically, let  $S_1$  and  $S_2$  be two answer sets, then (i)  $S_1 \geq S_2$  if  $V_{S_1} \leq V_{S_2}$ , i.e. if  $v_{S_1}(\varrho_i) \leq v_{S_2}(\varrho_i)$  for every  $i \in [1..n]$ ; (ii)  $S_1 > S_2$  if  $V_{S_1} < V_{S_2}$ , i.e. if  $V_{S_1} \leq V_{S_2}$  and for some  $i \in [1..n]$   $v_{S_1}(\varrho_i) < v_{S_2}(\varrho_i)$ .

$S$  is an *optimal answer set* of an ASO program  $\langle \mathcal{P}, \Phi \rangle$  if there is no answer set  $S'$  of  $\mathcal{P}$  such that  $S' > S$ .

Therefore, as a result of the comparison of the satisfaction vectors, the ASO semantics returns for the previous example  $S_1$  and  $S_3$  as preferred solutions. Observe that this result does not seem intuitive. In fact, the preference rules specify the preferences among drinks ( $\varrho_1, \varrho_2$ ) and desserts ( $\varrho_3$ ). As both solutions  $S_1$  and  $S_3$  have the second best choice of the drink (in the presence of *fish* and *beef* respectively), and the choice of the *dessert* is better satisfied by  $S_1$ , then  $S_1$  should be the preferred answer set. We want to stress that, the ASO semantics considers each preference rule separately and cannot establish the preference order between  $S_1$  and  $S_3$  owing to opposite satisfaction degrees of  $\varrho_1$  and  $\varrho_2$ , even if they refer to the same choice.

## ASO<sub>Ch</sub> Semantics

This paper introduces a new semantics, called ASO<sub>Ch</sub>, which extends the ASO semantics in order to avoid the problem above described. Differently from the ASO semantics, the ASO<sub>Ch</sub> semantics looks beyond the single preference rules, considering them as a tool for choice representation: head atoms correspond to the *choice options*, whereas the body of preference rule specifies the *choice context*, i.e. the decisions which have to precede this choice.

The ASO<sub>Ch</sub> semantics considers a prioritized program of the form  $\langle \mathcal{P}, \Phi \rangle$ , where  $\mathcal{P}$  is a (disjunctive) logic program and  $\Phi$  consists of a finite set of preference rules of the form  $\varrho : C_1 > \dots > C_k \leftarrow \text{body}$ , where *body* is a conjunction of literals, i.e. atoms or negation of atoms, and  $C_i$ s is a disjunction of atoms. All atoms in  $C_i$  are equally good choice options; atoms in  $C_i$  are preferred to the atoms in  $C_j$ , for  $i < j$  and  $i, j \in [1..k]$ . The evaluation strategy consists in the identification of choices and in the comparison of choices instead of single rules in order to select preferred solutions. More specifically, the partition of  $\Phi$  into a set of choices (subset of preference rules), denoted by  $\text{Ch}(\Phi)$ , is performed following the *choice identification strategy*: two preference rules  $\varrho_1$  and  $\varrho_2$  define the same choice  $\text{Ch}$ , denoted by  $\varrho_1, \varrho_2 \in \text{Ch}$ , if (i)  $\varrho_1$  and  $\varrho_2$  have at least one common atom in their heads; and (ii)  $\exists \varrho_3$  such that  $\varrho_1, \varrho_3 \in \text{Ch}$  and  $\varrho_3, \varrho_2 \in \text{Ch}$ . Thus, the choice identification process is based on the *rules identification* and the *transitive* properties, expressed by the first and second conditions, respectively.

After constructing the set of choices  $\text{Ch}(\Phi)$ , the preferred solutions of  $\langle \mathcal{P}, \Phi \rangle$  are computed by associating to each answer set  $S$  of  $\mathcal{P}$  a satisfaction vector reporting the degree of satisfaction of each choice  $\text{Ch} \in \Phi$ .

The evaluation strategy of the ASO<sub>Ch</sub> semantics is illustrated by the following example.

### Example 2 (Example 1 continued)

In order to extract the set of preferred solutions of  $\langle \mathcal{P}_1, \Phi_1 \rangle$ , the ASO<sub>Ch</sub> semantics aggregates  $\{\varrho_1, \varrho_2\} \in \Phi_1$ , having common atoms in their heads, into the same choice, say  $\text{Ch}_{\text{dr}}$ , and the latter preference rule  $\varrho_3$  into another choice, say  $\text{Ch}_{\text{d}}$ . Consequently, the set of preference rules  $\{\varrho_1, \varrho_2, \varrho_3\} \in \Phi$  is partitioned into two different choices:  $\text{Ch}_{\text{dr}} = \{\varrho_1, \varrho_2\}$  and  $\text{Ch}_{\text{d}} = \{\varrho_3\}$  that model the choice of drink and dessert respectively. The choice satisfaction vectors associated to the solutions of  $\mathcal{P}_1$  are  $V_{S_1} = (2, 1)$ ,  $V_{S_2} = (2, 2)$ ,  $V_{S_3} = (2, 2)$ , consequently,  $S_1$  is the preferred answer set owing to the dessert choice.  $\square$

It can be shown that the computational complexity of the ASO<sub>Ch</sub> semantics is not increased w.r.t. the ASO semantics. Note that in the ASO<sub>Ch</sub> semantics the head atoms of preference rules describe the possible options of the corresponding choice, whereas their *feasibility*, i.e. the really possibility of choosing these options, is determined by the choice context and by the constraints present in logic program. A variant of the ASO<sub>Ch</sub>, called ASO<sub>FCh</sub> that evaluates the choices on the basis on their really possible (feasible) options has been introduced in (?).

## The CHOPPER system

CHOPPER (CHOice OPTimizer for PrioritEd Reasoning) is an answer set optimization system realizing prioritized reasoning based on the choice evaluation, implemented at the University of Calabria. The system prototype has been developed on top of the well-known DLV prover by using Java 2 Platform. In particular, the answer set evaluation is performed with the DLV system, whereas the prioritized reasoning and the user interface are realized by means of personalized java procedures.

CHOPPER receives in input the prioritized program and the specification of the semantics to be applied and extracts and returns the preferred stable models as a result. The overall architecture of CHOPPER prototype is reported in Figure 2.

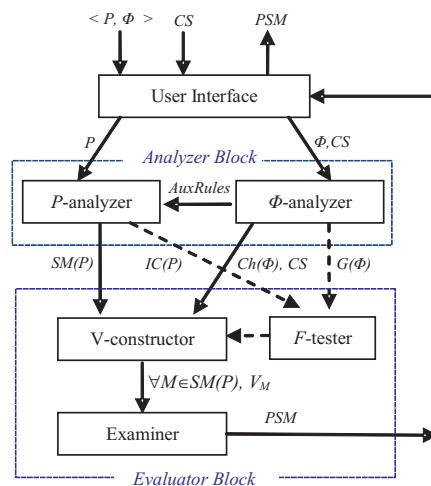


Figure 1: The CHOPPER System.

The system can be used by means of a User Interface -  $\mathcal{U}\mathcal{I}$  - which allows (i) to specify the prioritized program  $\langle \mathcal{P}, \Phi \rangle$ , (ii) to specify the semantics -  $\mathcal{CS}$  - chosen among the ASO, ASO<sub>Ch</sub> or ASO<sub>FCh</sub> semantics, and (iii) to visualize the obtained result, i.e. the set of preferred stable models -  $\mathcal{PSM}$ .