Semiring-based Constraint Logic Programming*

Stefano Bistarelli[1], Ugo Montanari[1], and Francesca Rossi[2]
{bista,ugo,rossi}@di.unipi.it

[1] University of Pisa, Dipartimento di Informatica
[2] University of Padova, Dipartimento di Matematica

Abstract. We extend the Constraint Logic Programming (CLP) formalism in order to handle semiring-based constraint systems. This allows us to perform in the same language both constraint solving and optimization. In fact, constraint systems based on semirings are able to model both classical constraint solving and more sophisticated features like uncertainty, probability, fuzziness, and optimization. We then provide this class of languages with three equivalent semantics: model-theoretic, fix-point, and proof-theoretic, in the style of CLP programs.

1 Introduction

Classical constraint satisfaction problems (CSPs) are a very expressive and natural formalism to specify many kinds of real-life problems. However, they also have evident limitations, mainly when they are used to represent real-life scenarios where the knowledge is not completely available nor crisp. In fact, in such situations, the ability of stating whether an instantiation of values to variables is allowed or not is not enough or sometimes not even possible. Recently, a proposal which extends classical CSPs in this direction has been developed [**?**,**?**], which is able to model many desired features, like fuzziness, probability, uncertainty, partiality, hierarchy and optimization. This framework is based on the observation that a semiring (that is, a domain plus two operations satisfying certain properties) is all what is needed to describe many constraint satisfaction schemes. In fact, the domain of the semiring provides the levels of consistency (which can be interpreted as cost, or degrees of preference, or probabilities, or others), and the two operations define how to combine constraints together. In particular, from one of the operations we can derive a partial order $\leq$ among the elements of the semiring which allows us to compare different elements: if $a \leq b$ then it means that b is *better than or equal to* a. This is crucial in situations which involve some kind of optimization. Constraint problems described according to this framework are called SCSP (for Semiring-based Constraint Satisfaction Problems).

Constraint logic programming (CLP) languages extended logic programming (LP) by replacing term equalities with constraints and unification with constraint solving. Programming in CLP means choosing a constraint system for a specific class of constraints and embedding it into a logic programming engine. This approach is very flexible since one can choose among many constraint systems

* Research supported by MURST as part of the project "Tecniche formali per la specifica, l'analisi, la verifica, la sintesi e la trasformazione di sistemi software"

without changing the overall programming language, and has shown to be very successful in specifying and solving complex problems in terms of constraints of various kind. However, it can handle only classical constraint solving. Thus it is natural to try to extend the CLP formalism in order to be able to handle also SCSP problems. In [**?**] we will call such an extension SCLP.

In passing from CLP to SCLP languages, we will replace classical constraints with the more general SCSP constraints. By doing this, we also have to modify the notions of interpretation, model, model intersection, and others, since we have to take into account the semiring operations and not the usual CLP operations.

After describing the syntax of SCLP programs, we will define three equivalent semantics for such languages: model–theoretic, fix-point, and operational. These semantics are conservative extensions of the corresponding ones for LP, since by choosing a particular semiring (the one with just two elements, *true* and *false*, and the logical *and* and *or* as the two semiring operations) we get exactly the LP semantics. The extension is in some cases predictable but it possesses some crucial new features. For example, the presence of a partial order among the semiring elements (and not a *total* order like it is in the LP/CLP case, where we just have two comparable elements) brings some conceptual complexity in some aspects of the semantics. In fact, in the operational semantics, there could be two refutations for a goal which lead to different semiring elements which are not comparable in the partial order. In this case, these elements have to be combined in order to get the solution corresponding to the given goal, and their combination could be not reachable by any derivation path in the search tree. This means that any constructive way to get such a solution by visiting the search tree would have to follow all the incomparable paths before being able to find the correct answer. However, in practice classical branch and bound techniques can be adapted to this framework to cut some useless branches. Notice also that, since we are not allowing function symbols in this paper, we can consider only a finite number of finite refutations. In fact, it is possible to identify a bound N to the depth of refutations s.t. those deeper than N do not bring more information due to the properties of the semiring operations.

We also show the equivalence of the three semantics. In particular, we prove that, given the set of all refutations starting from a given goal, it is possible to derive the declarative meaning of both the existential closure of the goal and its universal closure.