H. W. Guesgen J. Hertzberg

A Perspective of Constraint-Based Reasoning

An Introductory Tutorial

Springer-Verlag
Berlin Heidelberg New York
London Paris Tokyo
Hong Kong Barcelona
Budapest

Series Editor

Jörg Siekmann
University of Saarland
German Research Center for Artificial Intelligence (DFKI)
Stuhlsatzenhausweg 3, W-6600 Saarbrücken 11, FRG

Volume Editors

Hans Werner Guesgen
Joachim Hertzberg
Gesellschaft für Mathematik und Datenverarbeitung (GMD), Schloß Birlinghoven
Postfach 12 40, W-5205 St. Augustin 1, FRG

CR Subject Classification (1991): I.2.8, I.2.3-4

ISBN 3-540-55510-2 Springer-Verlag Berlin Heidelberg New York
ISBN 0-387-55510-2 Springer-Verlag New York Berlin Heidelberg

Printed in Germany

Typesetting: Camera ready by author/editor
Printing and binding: Druckhaus Beltz, Hemsbach/Bergstr.
45/3140-543210 - Printed on acid-free paper

Preface

Constraints are a recurring topic in Artificial Intelligence (AI) about which a wealth of literature has been written. Constraint techniques have been used in many computer programs, and a considerable amount of sub-concepts of constraints, specializations and algorithms have been produced, the relationships between which aren't always as apparent as one would like them to be. So, there seems to be some demand for a more general view of the field.

Our text is an attempt to present many facets of the field in a uniform way. It is based on the material for two tutorials we gave at the 1991 conference of the British Society for the Study of AI and Simulation of Behaviour (AISB91) and at the annual German Workshop on AI in 1991 (GWAI-91).

We would not have been able to write this text outside the creative atmosphere of AI groups like the ones at the German National Research Center for Computer Science (GMD) at Sankt Augustin, Germany, and at the International Computer Science Institute (ICSI) at Berkeley, California, where you have people sitting next-door who are engaged in all these interesting things like nonmonotonic reasoning, connectionism, Boltzmann machines, or reflective systems. In addition, we received fruitful comments and questions from the tutorial participants.

Thanks to Thomas Christaller for allowing us to close our doors for some weeks to write this text. Moreover, we are grateful to our colleagues in the qwertz project on AI planning methods for doing without us for some time: after all, this is not really planning.

The following persons have read preliminary versions of the whole text or of parts of it, resulting in innumerable large and small hints concerning the content, the presentation, or the English style of this text: Ursula Bernhard, Gerd Brewka, Rina Dechter, Jerry Feldman, Manfred Fidelak, Jürgen Kopp, Peter Ladkin, Marc Linster, Christoph Lischka, Gerd Paaß, and Jörg Siekmann.

The work presented in this text is partially funded by the German Federal Ministry for Research and Technology (BMFT) in the joint project TASSO under grant number ITW8900A7. TASSO is also part of the GMD *Leitvorhaben* Assisting Computer (AC).

Sankt Augustin,
March 1992

Hans Werner Guesgen
Joachim Hertzberg

Contents

Chapter 1

Introduction

1.1 What this Text Is About

Much of AI research is about problem solving strategies, for example, developing heuristic search algorithms, reasoning on representations of human knowledge, or performing planning tasks automatically. As a result, several techniques have been crystallized, one of which this text is about: constraint satisfaction or reasoning based on relations.

Constraint-based reasoning is used to solve a widespread field of problems. Some researchers had a certain task in mind and tried to find a method to solve that task efficiently. The Waltz algorithm [Waltz, 1972] is an example from the domain of computer vision. It was intended to interpret polyhedron diagrams as 3D objects in the blocks world. Other domains are, for example,

- circuit analysis [Stallman and Sussman, 1977],
- planning experiments in molecular biology [Stefik, 1981],
- job-shop scheduling [Fox *et al.*, 1982],
- temporal reasoning [Allen, 1983],
- diagnosis [Davis, 1984; Geffner and Pearl, 1987; de Kleer and Williams, 1986],
- and logic programming [Dincbas *et al.*, 1987; Jaffar and Lassez, 1987].

Soon it became clear that constraint satisfaction methods as such are useful and should be provided in form of computer programming languages. An example of such a language is described in G.L. Steele's thesis [1980]. The language is an extension of Lisp by constructs for defining and satisfying constraints. It has many successors, for example, [Gosling, 1983] or [Guesgen, 1989a]. More recently, constraint techniques have been incorporated into logic programming languages, yielding a whole new field of research and application: constraint logic programming [van Hentenryck, 1989].

Not only have the practical aspects of constraint satisfaction been explored in various ways, but also the theoretical ones. The theoretical work has led to results, for example, about the complexity and reducibility of constraint satisfaction problems [Mackworth, 1977; Freuder, 1978] and about the relation between constraint-based reasoning and databases or logic, e.g., [Bibel, 1988; de Kleer, 1989; Guesgen and Ladkin, 1990].

Constraint satisfaction techniques have become part of almost all introductory books on AI, and they can be found in dictionaries, for example, the Encyclopedia of Artificial Intelligence [Shapiro, 1987]. Moreover, whole books have been dedicated to constraint satisfaction; so has been this one. But it is different from others in an important aspect.

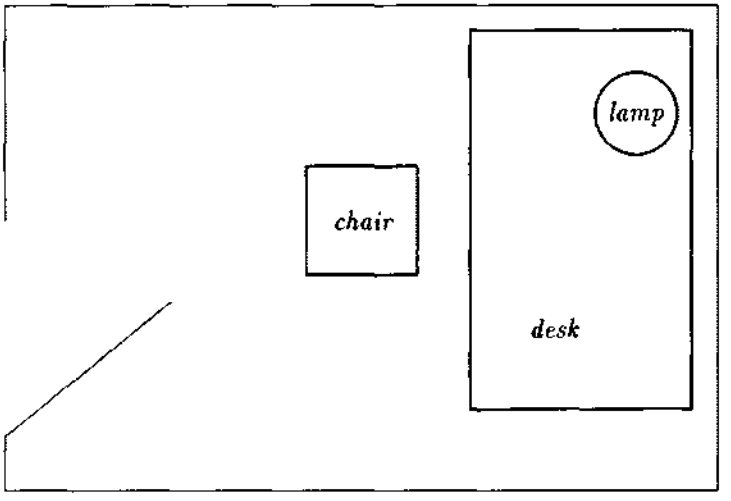
Working on practical examples with constraint languages has soon shown that using "mere" constraints is insufficient in several respects. Practical problems tend to yield, for example, inconsistent constraint networks, i.e., the problem is overspecified. In such a case, the constraint language is of little help. Or, as another example, domains may be infinite, rendering constraint approaches inapplicable. There are special adaptations of constraint approaches for these problems, but they are incompatible at first sight. Now, this text is different from others in that it presents all approaches under a common, generalizing view: dynamic constraints.

Actually, dynamic constraints are nothing really new, as we will show that they are no less and no more expressive than the ordinary constraints which you may have heard of already. They are just a new way of viewing constraints as something which has emerged from the attempt to put different constraint approaches under a one roof. However, housing them under this roof also provides new insights about the different approaches separately from each other.

Why did we take the time to write this text? We had the feeling that the possibility of presenting many different areas and approaches of constraint-based reasoning in a uniform way might be interesting in two respects. First, it seems to be a very practical basis for teaching constraint-based reasoning. Second, we think that a uniform view of the constraint world is a good basis for constraint research because it may shed light on aspects or relations of different areas of constraint reasoning that have been neglected so far, thereby simply stimulating new ideas.

One area for such ideas might have to do with the massively parallel implementation of constraint-based reasoning. Certainly this is also feasible with ordinary constraints, but it is the uniformity of dynamic constraint networks, again, that makes following this idea particularly suggestive in our framework. Consequently the chapters on the implementation of constraint-based reasoning contain some hints pointing into this direction.

It must be noted, however, that this text is *not* intended to serve as a self-contained textbook on constraint-based reasoning. It covers most areas of constraint-based reasoning, but in a real textbook, there would have to be, for example, additional parts on applications of constraints, or a more comprehensive treatment on constraint logic programming, which we just sketch as related work (in section 3.4). Moreover, this text does not have the degree of didactic sophistication we would expect from a book serving primarily learning or teaching needs. Let us regard it as a coherent, not really exhaustive text about some interesting view of the field of constraint-based reasoning.



The chair is on the left of the desk on which there is a lamp.

Figure 1.1: Sketch of an office from a bird's-eye view.

1.2 How this Text Is Organized

Chapter 1 which you are reading at the moment is the introduction and also an illustration of our motivation for writing this text. In addition, it shows how the text is organized.

Chapter 2 presents our demo domain for constraint-based reasoning: the office world, or in more precise terms, reasoning about drawings of Spartan office interiors as shown in figure 1.1. Although we use a well-known approach, namely Allen's temporal logic as adapted to spatial relations, the reader should at least a browse through this chapter, since the terminology is slightly changed.

Chapter 3 is the key chapter concerning the notion of dynamic constraints. It defines them and transfers the usual concepts such as solution or consistency to dynamic networks. In addition, it shows their relationship to ordinary constraints.

Chapter 4 is about a special form of constraint manipulation: relaxation which usually deals with inconsistent constraint problems. It is a demonstration of how to represent constraint relaxation in dynamic constraint networks. This concludes the first part of the text, which discusses the relevant notions and presents the material on a more conceptual level.

Chapter 5 starts the second part of the text that deals with concepts and algorithms for *solving* constraint satisfaction problems. To begin with, we describe good old backtracking and some improvements of it.

Chapter 6 refers to a technique that has become well-known as Waltz filtering and that may be viewed as a preprocessing tool for solving constraint satisfaction problems. For the purpose of computing solutions of a given problem rather than only preprocessing it, we extend traditional Waltz filtering by something which is called tagging.

Chapter 7 is the continuation of chapter 6, showing parallel and massively parallel implementations of Waltz filtering. We start with a simple parallel algorithm for filtering with and without tagging, and we end up with a scheme for transforming an arbitrary binary constraint problem into a connectionist network.

Chapter 8 discusses two approaches to finding good or even optimal solutions of constraint networks, i.e., to viewing constraint satisfaction as an optimization problem: simulated annealing and Boltzmann machines. Again, we discuss not only a serial implementation but also a massively parallel one.

Chapter 9 concludes the book with some remarks about spatial reasoning as such—and not just as a demo domain as in this text.

1.3 A Note for the Hasty Reader

Had we thought that our ideas could have been boiled down to 20 pages, we wouldn't have written a book but a 20 pages paper. So, to get all of it, you have to read the whole text.

However, provided that you just want to get an idea of what dynamic constraints are and provided that you are fit in classical constraint approaches, we recommend that you start with the key chapter 3, occasionally referring to the description of our demo domain in chapter 2. After that, you may look at the chapters which are most interesting for you; there are few hard dependencies. We recommend, however, to look at the relaxation chapter 4 before reading the optimization chapter 8 and to consider the chapters 6 and 7, both on filtering approaches with or without tagging, as a unit.