Foreword

Motivations for the approach of the present book

Since the 1970s, when it was identified as a class of problems with its own specificities, Constraint Satisfaction has quickly evolved into a major area of Artificial Intelligence (AI). Two broad families of very efficient algorithms (with many freely available implementations) have become widely used for solving its instances: general purpose structured search of the "problem space" (e.g. depth-first, breadth-first) and more specialised "constraint propagation" (that must generally be combined with search according to various recipes).

One may therefore wonder why they would use the computationally much harder techniques inherent in the approach introduced in the present book. It should be clear from the start that there is no reason at all if speed is the first or only criterion, as may legitimately be the case in such a typical Constraint Satisfaction Problem (CSP) as scene labelling.

But, instead of just wanting a final result obtained by any available and/or efficient method, one can easily imagine additional requirements of various types and one may thus be interested in how the solution was reached, i.e. by the *resolution path*. Whatever meaning is associated with the quoted words below, there are several inter-related families of requirements one can consider:

– the solution must be built by "constructive" methods, with no "guessing";

– the solution must be obtained by "pure logic";

– the solution must be "pattern-based", "rule-based";

– the solution must be "understandable", "explainable";

– the solution must be the "simplest" one.

Vague as they may be, such requirements are quite natural for logic puzzles and in many other conceivable situations, e.g. when one wants to ask explanations about the solution or parts of it.

Starting from the above vague requirements, Part I of this book will elaborate a formal interpretation of the first three, leading to a very general, pattern-based resolution paradigm belonging to the classical "progressive domain restriction" family and resting on the notions of a *resolution rule* and a *resolution theory*.

Then, in relation with the last purpose of finding the "simplest" solution, it will introduce ideas that, if read in an algorithmic perspective, should be considered as defining a new kind of search, "*simplest-first search*" – indeed various versions of it based on different notions of logical simplicity. However, instead of such an algorithmic view (or at least before it), a pure logic one will systematically be adopted, because:

– it will be consistent with the previous purposes,

– it will convey clear non-ambiguous semantics (and it will therefore include a unique complete specification for possibly multiple types of implementation),

– it will allow a deeper understanding of the general idea of "simplest-first search", in particular of how there can be various underlying concrete notions of logical simplicity and how these have to be defined by different kinds of resolution rules associated with different types of chain patterns. At this point, it may be useful to notice that the classical structured search algorithms are not compatible with pure logic definitions (as will be explained in the text).

Simplest-first search and the rating of instances

In this context, there will appear the question of rating and/or classifying the instances of a (fixed size) CSP according to their "difficulty". This is a much more difficult topic than just solving them. The families of resolution rules introduced in this book (by order of increasing complexity) will go by couples (corresponding to two kinds of chains with no OR-branching but with different linking properties, namely T-whips and T-braids); for each couple, there will be two ratings, defined in pure logic ways:

– one based on T-braids, allowing a smooth theoretical development and having good abstract computational properties; we shall devote much time to prove the *confluence property* of all the braid and T-braid resolution theories, because it justifies a "*simplest-first" resolution strateg*y (and the associated "simplest-first search" algorithms that may implement it) and it allows to find the "simplest" resolution path and the corresponding rating by trying *only one path*;

– one based on T-whips, providing in practice an easier to compute good approximation of the first when it is combined with the "simplest-first" strategy. (The quality of the approximation can be studied in detail and precisely quantified in the Sudoku case, but it will also appear in intuitive form in all our other examples.)

We shall explain in which restricted sense all these ratings are compatible. But we shall also show that each of them corresponds to a different legitimate pure logic view of simplicity.

In chapter 11, we shall analyse the scope of the previously defined resolution rules in terms of a search procedure with no guessing, Trial-and-Error (T&E), and of the depth of T&E necessary to solve an instance. There are universal ratings, respectively the B and the BB ratings, for instances in T&E(1) and T&E(2) (i.e. requiring no more than one or two levels of Trial-and-Error). Universality must be understood in the sense that they assign a finite rating to all of these instances, but not in the sense that they could provide a unique notion of simplicity. For instances beyond T&E(2), it is questionable whether a "pure logic" solution, with all the complex and boring steps that it would involve, would be of any interest; moreover, it appears that there may be many different incompatible notions of "simplest"; in chapter 12, we shall introduce the notion of a pattern of proof and, based on it, we shall re-assess our initial requirements. The main purpose is to provide hints about the scope of practical validity of our approach.

Examples from logic puzzles

Mainly because they can be described shortly and they are easy to understand with no previous knowledge, all the examples dealt with in this book will be logic puzzles: Latin Squares, Sudoku, N-Queens…, with a special status granted to Sudoku for reasons that will be explained in the Introduction. But they have been selected in such a way that they make us tackle very different types of constraints, so that this choice should not suggest a lack of generality in our approach: *transitive* constraints in Futoshiki, *non-binary arithmetic* constraints in Kakuro, *topological* and *geometric* constraints in Map colouring or path finding (Numbrix® and Hidato®).

In several places, we shall even give results that are only valid for 9×9 Sudoku (e.g. the unbiased whip classification results of minimal instances in chapter 6 and the analysis of extreme instances in chapter 11), for the purpose of illustrating with precise quantitative data questions that cannot yet be tackled with such detail in other CSPs and that call for further studies, such as:

– the difficulty (much beyond what one may imagine) of finding uncorrelated unbiased samples of minimal instances of a CSP, a pre-requisite for any statistical analysis; the way we present it shows that it is likely to appear in many CSPs; the final chapters on various other CSPs show that this is indeed true for them; (a related well known problem is that of finding the hardest instances of a CSP);

– the surprisingly high resolution power of short whips for instances in T&E(1);

– the concrete application of various classification principles to the extreme instances.

The "Hidden Logic of Sudoku" heritage [mainly for the readers of* HLS*]

The origins of the work reported in this book can be traced back to my choice of Sudoku as a topic of practical classes for an introductory course in Artificial Intelligence (AI) and Rule-Based Systems in early 2006. As I was formalising for myself the simplest classical techniques (Subset rules, xy-chains) before submitting them as exercises to my students, I had two ideas that kept me interested in this game longer than I had first expected: logical symmetries between three well-known types of Subset rules (Naked, Hidden and Super-Hidden, the last of which are commonly known as "Fish") and a simple non-reversible extension (xyt-chains) of the well-known reversible xy-chains. As time passed, the short article I had planned to write grew to the size of a 430-page book: *The Hidden Logic of Sudoku – HLS* in the sequel (first edition, *HLS1*, May 2007; second edition, *HLS2*, November 2007).

The present book inherits many of the ideas I first introduced in *HLS* but it extends them to any finite CSP. Based on the classical idea of *candidate elimination*, *HLS* provided a clear logical status for the notion of a candidate (which does not pertain to the original problem formulation) and it introduced the notions of a resolution rule and a resolution theory. All the concepts were strictly formalised in Predicate Logic (FOL) – more precisely in Multi Sorted First Order Logic (MS-FOL) – which (surprisingly) was a new idea: previously, all the books and Web forums had always considered that Propositional Logic was enough. Indeed, *HLS* had to make a further step, because intuitionistic (or, equivalently, constructive) logic is necessary for the proper formalisation of the notion of a candidate.

Notwithstanding the more general formulation, the "pattern-based" conceptual framework developed in this book is very close to that of *HLS*. From the start, the framework of *HLS* was intended as a formalisation of what had always been looked for when it was said that a "pure logic solution" was wanted. The basic concepts appearing in the resolution rules introduced in *HLS* were grounded in the most elementary notions used to propose or solve a puzzle (numbers, rows, columns, blocks, …); the more elaborate ones (the various types of chain patterns) were progressively introduced and strictly defined from the basic ones. Because the concepts of a *candidate* and of a *link* between two candidates were enough to formulate most of the resolution rules, extending them to any CSP was almost straightforward. The additional requirement that appeared in *HLS* in relation with the idea of rating, that of finding the *simplest* resolution path, is also tackled here according to the same general principles as in *HLS*.

On the practical puzzle solving side, *HLS1* introduced new resolution rules, based on natural generalisations of the famous xy-chains, such as xyt-, xyz- and zyzt- chains; contrary to those proposed in the current Sudoku literature, these were not based on "Subsets" (or almost locked sets – "ALS") and most of these chains were not "reversible"; the systematic clarification and exploitation of all the generalised symmetries of the game and the combination of my first two initial ideas had also led me to the "hidden" counterparts of the previous chains (hxy-, hxyt- hxyzt- chains). Later, I found further generalisations (nrczt- chains and lassoes), pushing the idea of supersymmetry to its maximal extent and allowing to solve almost any puzzle with short chain patterns. Giving a more systematic presentation of these new "3D" chain rules was the main reason for the second edition (*HLS2*).

Still later, I introduced (on Sudoku forums) other generalisations (that, in the simplified terminology of the present book and in a formulation meaningful for any CSP, will appear as whips, braids, g-whips, S_p-whips, W_p-whips, …). These may have justified a third edition of *HLS*, but I have just added a few pages to my *HLS* website instead – concentrating my work on another type of generalisation.

It appeared to me that most of what I had done for Sudoku could be generalised to any finite CSP [Berthier 2008a, 2008b, 2009]. But, once more, as I found further generalisations and as the analysis of additional CSPs with different characteristics was necessary to guarantee that my definitions were not too restrictive, the normal size of journal articles did not fit the purposes of a clear and systematic exposition; this is how this work grew into a new book, "*Constraint Resolution Theories*" (*CRT*, November 2011).

As for the resolution rules themselves, whereas *HLS* proceeded by successive generalisations of well-known elementary rules for Sudoku into more complex ones, in *CRT* and in the present book, we start (in Part II) from powerful rules meaningful in any CSP (whips, in chapter 5) equivalent (in the Sudoku case) to those that were only reached at the end of *HLS2* (nrczt- chains and lassoes).

As a result, in this book, patterns such as Subsets, with much less resolution power than whips of same size and with more complex definitions in the general CSP than in Sudoku, come after bivalue-chains, whips and braids, and also after their "grouped" versions, g-whips and g-braids. Moreover, Subsets are introduced here with purposes very different from those in *HLS*:
1) providing them with a definition meaningful in any CSP (in particular, independent of any underlying grid structure);
2) showing that whips subsume most cases of Subsets in any CSP;
3) illustrating by Sudoku examples how, in rare cases, Subset rules can nevertheless simplify the resolution paths obtained with whips;
4) defining in any CSP a "grouped" version of Subsets, g-Subsets; surprisingly, in the Sudoku case, g-Subsets do not lead to new rules, but they give a new perspective of the well-known Franken and Mutant Fish; this could be useful for the purposes of classifying these patterns (which has always been a very obscure topic);
5) showing that, in any CSP, the basic principles according to which whips are built can be generalised to allow the insertion of Subsets into them (obtaining S_p-whips), thus extending the resolution power of whips towards the exceptionally hard instances.

What is new with respect to "Constraint Resolution Theories" [mainly for the readers of* CRT*]

This book can be considered as the second, revised and largely extended edition of *Constraint Resolution Theories* (*CRT*). Following a colleague's advice, we changed the title (which seemed too technical) so that it includes the "Constraint Satisfaction" key phrase referring to its global domain; "Pattern-Based" was then a natural choice for qualifying our approach, while the explicit reference to "Logic Puzzles" became almost necessary with the addition of all the examples in part IV to the already existing Sudoku content. Apart from this cosmetic change, there are three different degrees of newness with respect to *CRT*, in increasing magnitude.

Firstly, this book corrects a few typos and errors that remained in *CRT* in spite of careful re-readings; in several places, it also marginally improves or completes the wording and it adds a few remarks or comments; moreover:

– z-chains are no longer included in the analysis of loops in sections 5.8.1 to 5.8.3; instead, the obvious and simpler fact that z-whips subsume z-chains with a global loop is mentioned;

– an unnecessary restriction in the definition of a g-label (section 7.1.1.1) has been eliminated, without modifying the notion of a g-link; this leaves unchanged the definitions of a g-candidate and of predicate "g-linked" (relating a g-candidate and a candidate); as before, these two definitions refer to the full underlying g-label and label (this is why the restriction was unnecessary); nothing else had to be changed in chapter 7 or in any place where g-labels are dealt with; in particular, this does not change the sets of g-labels of the various examples already tackled by *CRT*; however, the restriction made it impossible to apply the initial definition given in *CRT* to g-labels in Futoshiki (see chapter 14);

– the "saturation" or "local maximality" condition in the definition of a g-label has been broadened for an easier applicability to new examples; it has also been isolated by splitting the initial definition into two parts; as it was there only for efficiency purposes, but it had no impact on theoretical analyses, this entails no other changes; however, the efficiency purposes should not be underestimated: section 15.5 shows how essential this condition is in practice in Kakuro;

– section 11.4 of *CRT* (bi-whips, bi-braids, W*-whips and B*-braids) has been significantly reworded, corrected and extended, giving rise to a new chapter of its own (chapter 12);

– a section (17.4) describing our general pattern-based CSP-Rules solver, used for all the examples presented in this book, has been introduced.

Secondly, this book adds a few new results, mainly to the W-whip and B-braid patterns and/or to the Sudoku CSP case study. The following list is not exhaustive:

– very instructive whip[2] examples are given in section 8.8.1; they are the key for understanding why whips can be more powerful than Subsets of same size;

– an example of a non-whip braid[3] in Sudoku is given in section 5.10.5;

– a new graphico-symbolic representation of W-whips is introduced in section 11.2.9, based on the analogy between whips and Subsets;

– the most recent collections of extreme puzzles, harder than most of those already considered in *CRT*, published in the meantime by various puzzle creators, are analysed and their $B_?B$ classifications are given in section 11.4; these new results show that a few puzzles (we have found only three in these collections) require B_7-braids and they provide very strong support to our old conjecture that all the 9×9 Sudoku puzzles can be solved by T&E(2) and to our new one that they can all be solved by B_7-braids;

– occasionally, larger sized Sudoku grids are considered; this allows in particular to show that the universal solvability by T&E(2) is not true for them.

Thirdly and most importantly, chapter 12 and part IV about modelling various logic puzzles are almost completely new; in particular:

– chapter 12, revolving around the notion of a *pattern of proof*, shows that our initial simplicity and understandability requirements may be at variance for instances beyond T&E(1) or gT&E(1); it discusses various options for their interpretation, such as B*-braid solutions; it shows that a pure logic approach is still possible in theory, although the computational complexity may be much higher, depending on which *patterns of proof* one is ready to accept;

– chapter 13, via an illustrative example (the sk-loop in Sudoku), tackles general questions about *modelling resolution rules*; these arise when one wants to formalise new (possibly application-specific) techniques; although part of the material in it has been available for several years on the Sudoku part of our website in a rather technical form, subtle changes (making the presentation much simpler and slightly more general) appear here for the first time;

– chapter 14 on *transitive constraints* and the *Futoshiki* CSP concretely shows how the general concepts and resolution rules defined in this book can be applied to a CSP with significantly different types of constraints (inequalities) than the symmetric ones considered in the LatinSquare, Sudoku and N-Queens examples; it also shows that the few known, apparently application-specific, resolution rules of Futoshiki (ascending chains, hills and valleys) are special cases of these general rules; finally, it indicates how our controlled-bias approach to puzzle generation, at the basis of any unbiased statistical results, can be adapted to it in a straightforward way;

– chapter 15 on *non-binary arithmetic constraints* and the *Kakuro* CSP may be the most important one among our non-Sudoku examples, as it shows that the binary constraints restriction of our approach can be relaxed not only in theory but also in practice and that non-binary constraints can be efficiently managed in application-specifc ways (better than by relying on the standard general replacement method);

– chapter 16 deals with some *topological and geometric constraints* associated with *map colouring* and *path finding* (in Numbrix® and Hidato®); together with chapters 14 and 15, it confirms that our generalisations from Sudoku to the general CSP work concretely – a point in which *CRT* was partially lacking.