# 12. Final remarks

In these final retrospective remarks, which are intended neither as a summary nor as a conclusion, we shall first highlight and comment some overlapping facets of what has been achieved for the general CSP (with a few open questions). Considering that a third of this book is an illustration of the general theory with the Sudoku CSP, the second section will be a quick review, mainly for the readers of *HLS*, of what is new with respect to *HLS2*.

## 12.1. About our approach of the finite CSP

### *12.1.1. About the general distinctive features of our approach*

There are five main inter-related reasons why this book diverges radically from the current literature on the finite CSP[1]:

– almost everything in our approach, in particular all our definitions and theorems, is formulated in terms of *pure logic*, independently of any algorithmic implementation; (apart from the obvious logical re-formulation of a CSP, the current literature on CSPs is mainly about algorithms for solving them and comparisons of such algorithms);

– we systematically use redundant sets of CSP variables;

– we fix the main parameter defining the "size" of a CSP and we are not concerned with the usual theoretical perspectives of complexity, such as NP-completeness of a CSP with respect to its size;

– we nevertheless tackle questions of complexity, in terms of the statistical distribution of the *minimal instances* of a fixed size CSP; although all our rules are valid for all the instances of a CSP, without any kind of restriction, we grant minimal instances a major role in all our statistical analyses and classification results; the thin layer of instances they define in the whole forest of possible instances (see chapter 6 for this view) allows to discard secondary problems that multi-solution or over-constrained instances would raise; (the notion of minimality is almost unknown in the CSP world);

[1] We are not suggesting that our approach is better than the usual ones; we are aware that our purposes are non-standard and they may be irrelevant when speed of resolution is the main criterion.

– last but not least, our *purposes* lie much beyond the usual ones of finding a solution or finding the fastest algorithm for it. Here, *instead of the solution as a result, we are interested in the solution as a proof of the result, i.e. in the resolution path*. Accordingly, we have concentrated on finding *understandable, meaningful, pure logic* resolution paths – though these words do not have a clear pre-assigned meaning.

We have taken this purpose into account in Part I by interpreting the "pure logic" requirement literally – i.e. as a solution completely defined in terms of mathematical logic (with no reference to any algorithmic notions). Thus, we have introduced a general resolution paradigm based on progressive candidate elimination (which amounts to progressive domain restriction, a classical idea in the CSP community), but in which each of these eliminations is justified by a single, well defined *resolution rule* of a given *resolution theory* and is interpreted in modal (non algorithmic) terms. We have established a clear logical (intuitionistic) status for the notion of a candidate (a notion that does not pertain to the CSP Theory). Moreover, we have shown that the modal operator that naturally appears in any formal definition of a candidate can be "forgotten" when we state resolution rules, provided that we work with intuitionistic (or constructive) instead of classical logic (which is not a restriction in practice).

Once this logical framework is set, a more precise purpose can be examined, not completely independent from the original vague "understandable" and "meaningful" ones: one may want the *simplest* pure logic solution. As is generally understood without saying when one speaks of the simplest solution to a mathematical problem, we mean neither easiest to discover for a human being nor computationally cheapest, but simplest to understand for the reader. Even with such precisions, we have shown that "simplest" may still have many different, all logically grounded, meanings, associated with different (purely logical) ratings of instances.

Taking for granted that hard minimal instances of most fixed size CSPs cannot be solved by elementary rules but they require some kind of chain rules (with the classical xy-chains of Sudoku as our initial inspiration), we have refined our general paradigm by defining families of resolution rules of increasing (logical and computational) complexity, valid for any CSP: some reversible (Bivalue Chains, Reversible Subset Chains, Reversible g-Subset Chains) and some oriented, much more powerful ones (whips, g-whips, $S_p$-whips, $gS_p$-whips, $W_p$-whips and similar braid families).

The different resolution paths obtained with each of these families correspond to different legitimate meanings of "simplest" (when they lead to a solution) and, in spite of strong subsumption relationships, we have shown that none of them can be completely reduced to another. Said otherwise: there does not seem to be any universal notion of simplicity for the resolution of a CSP; this is exemplified by the various ratings ascribed to some CSP instances in several chapters.

### *12.1.2. About our resolution rules (whips, braids, …)*

Regarding these new families of chain rules, now reversing the history of our theoretical developments, four main points should be recalled:

– We have introduced a formal definition of Trial-and-Error (T&E), a procedure that, in noticeable contrast with the well known structured search algorithms (breadth-first, depth-first, …) and all their CSP specific variants implementing some form of constraint propagation (arc-consistency, path-consistency, MAC, …), allows no "guessing", in the sense that it accepts no solution found by sheer chance during the search process: a value for a CSP variable is accepted only if all its other possible values have been tested and each of them has been proven to lead to a contradiction.

– With the "T&E vs braids" theorem and its "T&E(T) vs T-braids" extensions to various resolution theories T, we have proven that a solution obtained by the T&E(T) procedure can always be replaced by a "pure logic" solution based on T-braids, i.e. on sequential patterns accepting simpler patterns taken from the rules in T as their building blocks.

– Because its importance could not be over-estimated, we have proven in great detail that all our generalised braid resolution theories (braids, g-braids, $S_p$-braids, $gS_p$-braids, $B_p$-braids, …) have the confluence property. Thanks to it, we have justified the idea that these types of logical theories can be supplemented by a "simplest first" strategy, defined by ascribing in a natural way a different priority to each of their rules. When one tries to compute the rating of an instance and to find the simplest, pure logic solution for it, in the sense that it has a resolution path with the shortest possible braids in the family (which the T&E procedure alone is unable to provide), this strategy allows to consider only one resolution path; without this property, all of them should *a priori* be examined, which would add an exponential factor to computational complexity. Even if the goal of maximum simplicity is not retained, the property of stability for confluence of these T-braid resolution theories remains useful, because it guarantees that valid eliminations and assertions occasionally found by any other consistent opportunistic solving methods cannot introduce any risk of missing a solution based on T-braids.

– With the statistical results in chapter 6, we have also shown that, in spite of a major structural difference between whips and braids (the "continuity" condition), whips (even if restricted to no-loop ones) are a very good approximation of braids (at least in the Sudoku case), in the double sense that: 1) the associated W and B ratings are rarely different when the W rating is finite and 2) the same "simplest first" strategy, *a priori* justified for braids but not for whips, can be applied to whips, with the result that a good approximation of the W rating is obtained after considering only one resolution path (i.e. the concrete effects of non confluence of the whip resolution theories appear only rarely). This is the best situation one can desire for a restriction: it reduces structural (and computational) complexity but it entails little difference in classification results.[2] Of course, much work remains to be done to check whether this proximity of whips and braids is true for CSPs other than Sudoku and for all the types of extended whips and braids defined in this book.

### *12.1.3. About human solving based on these rules*

The four above-mentioned points have their correlates regarding a human trying to solve an instance of a CSP "manually" (or should we say "neuronally"?), as may be the "standard" situation for some CSPs, such as games:

– It should first be noted that T&E seems to be the most natural resolution method for a human who is unaware of more complex possibilities and who does not accept guessing. This was initially only a vague intuition. But, with time, it has received some concrete support from our experience in the Sudoku micro-world (with friends, students, contacts, or from questions of newcomers on forums), considering the way new players spontaneously re-invent it without even having to think of it. Indeed, it does not seem that they reject guessing *a priori*; they start by using it and they feel unsatisfied about it after some time; "no-guessing" appears as an additional *a posteriori* requirement.

– The "T&E vs braids" theorem means that the most natural T&E solving technique, in spite of being strongly anathemised by some Sudoku experts, is not so far from being compatible with the abstract "pure logic" requirement. Moreover, its proof shows that a human solver can always modify a T&E solution in order to present it as a braid solution. Thanks to the subsumption theorems or to the more general "T&E(T) vs T-braids" theorem, this remains true when he learns more elaborate techniques (such as Subset or g-Subset rules) and starts to combine them with T&E.

– Finding the shortest braid solution is a much harder goal than finding any solution based on braids and this is where the main divergence with a solution obtained by mere T&E occurs. For the human solver who started with T&E, it is nevertheless a natural step to try to find a shorter (even if not the shortest) solution. An obvious possibility consists of excising the useless branches; but one can also look for alternative braids, either for the same elimination or for a different one.

---

[2] By contrast, the "reversibility" condition often imposed on chains (never clearly formulated before *HLS* but widely accepted in some Sudoku circles) is very restrictive and it leads some players to reject solutions based on non-reversible (or "oriented") chains (such as whips and braids) and to the (in our opinion, hopeless for really hard instances) search for extremely complex patterns (such as all kinds of what we would call extended g-Fish patterns: finned, sashimi, chains of g-Fish, …). This said, we acknowledge that Reversible Subset Chains (Nice Loops, AICs) may have some appeal for moderately complex instances.

– As for the fourth point, a human solver is very likely to have spontaneously the idea of using the continuity condition to guide his search for a contradiction on some target Z: it means giving a preference to the last result obtained.

Finally, for a human solver, the transition from the spontaneous T&E procedure to the search for whips can be considered as a very natural process. Looking for Subsets and then for g-Subsets can also be considered as a natural, though different, evolution. And the two can be combined. Once more, there is no unique way of defining what "the best solution" may mean.

Of course, a human player can follow a very different learning path, starting with xy-chains and progressively trying to spot patterns from the ascending sequence of more complex rules following our discovery path in *HLS*.

### *12.1.4. About the similarity between Subsets and whips/braids of same size*

We have noticed a remarkable formal similarity between Subsets and whips/braids of same size (see Figure 11.3 and comments there). It has appeared in very explicit ways in the proofs of confluence and of the generalised "T&E vs braids" theorems for the $S_p$-braids and $B_p$-braids.

But whips/braids have a much greater resolution power than Subsets of same size, at least in the Sudoku CSP, as shown by the general subsumption theorems in section 8.7 and the specific statistical results in Table 8.1. As mentioned in section 8.7.3, these results indicate that the definition of Subsets is much more restrictive than the definition of whips. In Subsets, transversal sets are defined by a single constraint. In whips, the fact of being linked to the target or to a given previous right-linking candidate plays a role very similar to each of these transversal sets. But being linked to a candidate is much less restrictive than being linked to it via a pre-assigned constraint; in this respect, the three elementary examples for whips of length 2 in sections 8.7.1.1 and 8.8.1 are illuminating. As shown by the subsumption and almost-subsumption results in section 8.7, the few cases of Subsets not covered by whips because of the restrictions related to sequentiality are too rarely met in practice to be able to compensate for this.

For the above reasons, we conjecture that, in any CSP, whips/braids have a much greater resolution power than Subsets of same length p, for small values of p; for larger values of p, it is less clear because there may be an increasing number of cases of non-subsumption but there may also be more ways of being linked to a candidate. Probably, much depends on how many different constraints a given candidate can participate in. This is an area where much more work is necessary.

### *12.1.5. About minimal instances and uniqueness*

Considering that, most of the time, we restrict our attention to minimal instances that (by definition) have a unique solution, one may wonder why we do not introduce any "axiom" of uniqueness. Indeed, there are many reasons:

– it is true that we restrict all our statistical analyses of resolution rules to minimal instances, for reasons that have been explained in the Introduction; but it does not entail that validity of resolution rules should *per se* be limited to minimal instances; on the contrary, they should apply to any instance; in a few examples in this book, our rules have even been used to prove non-uniqueness or non-existence of solutions;

– as mentioned in the Introduction, from the point of view of Logic, uniqueness cannot be an *axiom*, at least not an axiom that could impose uniqueness of a solution; it can only be an *assumption*; moreover, when incorrectly applied to a multi-solution instance, the *assumption* of uniqueness can lead, via a vicious circle, to the erroneous *conclusion* that an instance has a unique solution; we have given an example in *HLS1*, section XXII.3.1 (section 3.1 of chapter "Miscellanea" in *HLS2*);

– uniqueness is not a constraint the CSP solver (be he human or machine) is expected or can choose to satisfy; in some CSPs or some situations (such as for statistical analyses or for games like Sudoku), uniqueness may be a requirement to the provider of instances (he should provide only "well formed", i.e. minimal, instances); the CSP solver can then decide to trust his provider or not; if he does and he uses rules based on it in his resolution paths, then uniqueness can best be described as an oracle; for this reason, in all the solutions we have given, uniqueness is never assumed, but it is proven constructively from the givens;

– the fact is, there is no known way of exploiting the assumption of uniqueness for writing any general resolution rule for uniqueness; and we can take no inspiration in the Sudoku case, because all the known rules for uniqueness are based on Sudoku specific constraints;

– in the Sudoku case, the known rules of uniqueness, when added to a resolution theory with the confluence property, destroy confluence (see *HLS* for an example);

– still in the Sudoku case, it does not seem that the known rules for uniqueness have much resolution power; there is no known example that could be solved if they are used but that could not without them.

Of course, we are not trying to deter anyone from using uniqueness, if they like it, in CSPs for which it allows to formulate specific rules, such as Sudoku (where it has always been a very controversial topic, but it has also led to the definition of smart rules); in some rare cases, it can simplify the resolution paths. We are only explaining why we chose not to use it in our theoretical approach.

### *12.1.6. About minimal instances vs density and tightness of constraints*

Two global parameters of a CSP, its "density of constraints" and its "tightness", have been identified in the classical CSP literature. Their influence on the behaviour of general-purpose CSP solving algorithms has been studied extensively and they have also been used to compare such algorithms. (As far as we know, these studies have been about unrestricted CSP instances; we have been unable to find any reference to the notion of a minimal instance in constraint satisfaction.)

Definitions: the *density of constraints* of a CSP is the ratio between the number of label pairs linked by a constraint (supposing that all the constraints are binary) and the total number of label pairs; the *tightness* of a CSP is the ratio between the number of label pairs linked by a "strong" constraint (i.e. a constraint due to a CSP variable) and the number of label pairs linked by a constraint.

Density reflects the intuitive idea that the nodes of a graph (here, the graph of labels) can be more or less tightly linked by the edges (here the binary contradictions); it also evokes a few general theorems relating the density and the diameter of a random graph (a topic that has recently become very attractive because of communication networks). Tightness evokes the difference we have mentioned between Sudoku (tightness 100%) and n-Queens (tightness ~ 50%, depending on n).

In the context of this book, relevant questions related to these parameters should be about their influence on the scope of the various types of resolution rules with respect to the set of minimal instances of the CSP. However, how these two parameters should be defined in this context is less obvious than it may seem at first sight. The question is, should one compute these parameters using all the labels of the CSP or only the actual candidates?

Taking the 9×9 Sudoku example (again!), the computation is easy for labels: there are 729 labels (all the nrc triplets) and each label is linked to 8 different labels on each of the n, r, c axes, plus 4 remaining labels on the b axis. Each label is thus linked to the same number (28) of other labels and one gets a density equal to 28/728 = 3.846%. More generally, for n×n Sudoku with $n = m^2$, density is:
$(4m^2\text{-}2m\text{-}2)/(m^6 \text{ -}1)$; it tends rapidly to zero (as fast as $4/n^2$) as the grid size n increases.

However, considering the first line of each resolution path in this book, one can check that for a minimal puzzle, after the initial Elementary Constraint Propagation rules have been applied (i.e. after the straightforward initial domain restrictions), the number of candidates remaining in the resolution state $RS_P$ of an instance P is much smaller. As all that happens in a resolution path depends only on $RS_P$, a definition of density based on the candidates in $RS_P$ can be expected to be more relevant. But, the analysis of the first series of 21,375 puzzles produced by the controlled-bias generator, leads to the following conclusions, showing that neither the number of candidates in $RS_P$ nor the density of constraints in $RS_P$ have any significant correlation with the difficulty of a puzzle P (measured by its W rating):

– the number of candidates in $RS_P$ has mean 206.1 (far less than the 729 labels) and standard deviation 10.9; it has correlation coefficient -0.20 with the W rating;

– the density of constraints in $RS_P$ has mean 1.58% (much less than when computed on all the labels) and standard deviation 0.05%; its has correlation coefficients -0.16 with the number of candidates in $RS_P$ and -0.06 with the W rating.

Can tightness give better or different insights? This parameter plays a major role in the left to right extension steps of the partial chains of all the types defined in this book. In n×n Sudoku or n×n LatinSquare, tightness is 100%, whatever the value of n; these examples may therefore not be used to investigate this parameter. If there are few CSP variables, there may be few chains. However, from the millions of Sudoku puzzles we have solved, problems that appear for the hardest ones solvable by whips or g-whips arise from two opposite causes: not only because there are too few partial whips or g-whips (and no complete ones), but also because there are too many useless partial whips or g-whips (eventually leading to memory overflow problems).

One idea that needs be explored in more detail is that the possible effects of initial density or tightness of constraints are minimised by considering only the thin layer of minimal puzzles (as is the case for the number of givens).

### *12.1.7. About a strategic level*

We have used the confluence property to justify the definition of a "simplest first strategy" for all the braid (and, by extension, the whip) resolution theories. This strategy fits the goals of finding the simplest solution (keeping the above comments on "simplest" in mind) and of rating an instance.

Other systematic strategies can also be imagined. One of them is considering subsets of CSP variables of "same type" and restricting all the rules to such subsets. This is what we have done for Sudoku in *HLS1*, with the 2D rules. It is easy to see that, as the "2D" rules are the projections of the "3D" ones presented here, all the 2D-braid theories (in the four rc, rn, cn and bn spaces) are stable for confluence and have the confluence property; it is therefore also true of their union. In *HLS1*, we have shown that 97% of the Sudogen0 puzzles can be solved by such 2D rules (the real percentage may be a little less for an unbiased sample). We still consider these rules as interesting special cases that have an obvious place in the "simplest first" strategy and that may be easier to find and/or to understand for a human player.

Now, it is very unlikely that any human solver would proceed in such a systematic way as described in the above strategies. He may prefer to concentrate on some aspect of the puzzle and try to eliminate a candidate from a chosen cell (or group of cells). As soon as he has found a pattern justifying an elimination, he applies it. This could be called the opportunistic "first-found-first-applied strategy". And, thanks to stability for confluence, it is justified in all the generalised braid resolution theories defined in this book (there can be no "bad" move blocking the way to the solution).

What may be missing however in our approach is more general "strategic" knowledge[3] for orienting the search: when should one look for such or such pattern? But the fact is, we have no idea of which criteria could constitute a basis for such knowledge. Moreover, even in the most studied Sudoku CSP, whereas there is a plethora of literature on resolution techniques (sometimes misleadingly called strategies), nothing has ever been written on the ways they should be used, i.e. on what might legitimately be called strategies. In particular, one common prejudice is that one should first try to eliminate bivalue/bilocal candidates (i.e., in our vocabulary, candidates in bivalue rc, rn, cn or bn cells). Whereas this may work for simple puzzles, it is almost never possible for complex ones. This can easily be seen by examining some of the examples of this book, with the long sequences of whip eliminations necessary before a Single is found: if any of these eliminations had occurred for a candidate from a bivalue/bilocal cell, then it would have been immediately followed by a Single.

### *12.1.8. About ratings and the requirement for the "simplest" solution*

Our initial motivations included a "pure logic solution" and a "simplest solution". If the first goal has been reached in Part I, one may wonder what the second goal has become.

For any instance P of any CSP, several ratings of P have been introduced. All of them are defined in pure logic terms and are intrinsic properties of P; moreover, they have been shown to be largely mutually consistent, i.e. "most of the time" they assign the same ratings to P (at least for Sudoku). Moreover, for any CSP whose minimal instances can be solved with at most two levels of Trial-and-Error, the BB rating, a rating that can thus be considered as universal, has even been defined.

But what the multiplicity of these logically grounded ratings shows is that there is one thing all our formal analyses cannot do in our stead: choosing what should be considered as "simplest". And this can only depend on one's specific goals. Let us illustrate this with the Sudoku CSP. Whether these remarks would apply to other CSPs, or how they should be modified, remains an open question.

---

[3] [Laurière 1978] presents a different perspective, based on general-purpose heuristics.

If one is interested in the simplest solution for all the minimal puzzles, then, considering the statistical results of chapter 6, a whip solution would certainly be the simplest one, *statistically*; a g-whip solution would be a good alternative. “Statistically” means that, in rare cases, a better solution based on Subsets or g-Subsets or Reversible Subset Chains could be found.

If one is interested in providing examples of some particular set of techniques or promoting them, then a solution considered as the simplest must (tautologically) use only these techniques; the job will then be to provide nice examples of such puzzles; this is the approach implicitly taken by most puzzle providers and most databases of “typical examples” associated with solvers. Unfortunately, apart from those here and in *HLS*, we lack both formal studies of such sets of techniques and statistical analyses of their scopes.

If one is interested in the “hardest” puzzles, then the first thing should be to specify what is meant by “hardest” (in particular with respect to which rating); puzzles harder than the “hardest” known ones (wrt the SER rating) keep being discovered; one can consider that Part III of this book (apart from chapter 8) is dedicated to resolution rules for the hardest puzzles; it shows that different techniques of increasing complexity can be used to deal with them. Much seems to depend on two parameters: the maximal depth d of Trial-and-Error necessary to solve these instances and the maximal look-ahead l necessary to solve them at depth d-1. (Even for Sudoku, although we can reasonably conjecture that d=2, we have no formal proof of this; and we have no estimate for l, except that $l \geq 6$).) However, for the very hardest puzzles, it may happen that the whole requirement of simplicity becomes merely meaningless: the existence of extremely rare but very hard instances that cannot be solved by any “simple” rules is a fact that cannot be ignored.

## 12.2. About the Sudoku CSP, beyond *HLS2* [for the readers of *HLS*]

This section is a quick review of the main points related to the Sudoku CSP that are new with respect to *HLS*.

### *12.2.1. About the general framework*

The general formal logic framework introduced in this book (all of Part I), when applied to Sudoku, is globally the same as in *HLS*. Only two slight differences appear: we now use Gentzen’s “natural logic” instead of Hilbert’s axioms; and, when dealing with the resolution states (formerly called “knowledge states” in *HLS*), we refer to the logic of necessitation instead of epistemic logic (or logic of knowledge). None of these changes has any practical consequences; in particular, resolution theories should still be understood as theories in intuitionistic (constructive) logic. Our definition of a resolution theory was slightly less precise in *HLS*.

But the main difference with *HLS* is that Sudoku is now considered as a CSP in a more systematic way than it was there; in particular:

– the cells in the Extended Sudoku Board are now explicitly interpreted as representations of CSP variables;

– the nrc notation introduced in *HLS2* as a convenient way of representing chains, now appears as an obvious special case of a natural notation for any CSP;

– basic interactions ("pointing" and "claiming") are systematically written as instances of whip[1];

– Subsets rules are first formulated in a general way, meaningful for any CSP, in terms of CSP variables and transversal sets, before being re-expressed for Sudoku in the usual terms of numbers, rows, columns and blocks; Naked, Hidden and Super-Hidden (Fish) Subsets are not only related by symmetry as they were in *HLS*; from the CSP point of view, they are now the very same rule, because the symmetries have been used at a higher level to introduce new CSP variables; (of course, this does not change anything for all practical purposes);

– the main change brought by this new perspective of Subsets is, it allows to introduce their "grouped" version (g-Subsets) as a natural generalisation (in the same way as several chain patterns have a grouped version) and to re-interpret the well-known Franken and Mutant Fish as g-Subsets.

### *12.2.2. New resolution rules and new ratings*

For the parts specifically dedicated to the Sudoku CSP (about a third of this book), they are very far from constituting a third edition of *HLS* (none of the examples, specialised versions of resolution rules or independence theorems present in *HLS2* has been reproduced here). Indeed, these parts start where *HLS2* ended.

We first introduced the following patterns, resolution rules and/or topics (that were not in *HLS2*) on the Sudoku Player's Forum in 2008; but in this book, most of them are now presented in the more general CSP framework (with a somewhat different, simpler terminology):

– whips (which are a more synthetic view of both nrczt- chains and lassoes);

– braids, with a detailed proof of the confluence property of braid resolution theories;

– definition of the Trial-and-Error procedure (T&E) and proof of the "T&E vs braids" theorem;

– definition of the controlled bias generator; comparison of various kinds of generators; unbiased statistics for the W rating for a much larger sample than in

*HLS*; [even though it was already possible to generate large random samples of minimal puzzles, before *HLS1* the literature had concentrated on isolated examples illustrating specific rules; systematic studies of large collections of puzzles had been lacking; to palliate this deficiency, detailed numerical results about the number of puzzles solved by each type of rule had been given in *HLS*, but they were still based on the biased samples produced by the currently available generators];

– g-whips and g-braids; proofs of the associated confluence property and "gT&E vs g-braids" theorem;

– detailed subsumption theorems for whips and braids, showing that they capture "almost all" but not all the cases of Subset rules;

– Reversible $S_p$-chains, obtained by allowing the insertion of Subsets as right-linking objects in bivalue chains; proof that these chains are the same thing as grouped AICs or Nice Loops (but our definition does not involve the unnecessarily complex notion of a "restricted common"); proof of the confluence property for the associated resolution theories;

– $S_p$-whips and $S_p$-braids, obtained by allowing the insertion of Subsets as right-linking objects in whips and braids; proofs of the associated confluence property and "T&E($S_p$) vs $S_p$-braids" theorem; analysis of their scope;

– $gS_p$-Subsets, the "grouped" version of Subsets, a generalisation allowed by considering Subset patterns ($S_p$-subsets) from the general CSP point of view; they provide a new view of Franken and Mutant Fish;

– Reversible $gS_p$-chains, $gS_p$-whips and $gS_p$-braids, obtained by allowing the insertion of g-Subsets as right-linking objects in bivalue chains, whips and braids;

– $W_p$-whips and $B_p$-braids; proofs of the associated confluence property and "T&E($B_p$) vs $B_p$-braids" theorem; relation of B-braids with T&E(2), providing a finite BB rating for all the known minimal puzzles [and indications that $B_6B$ could also be universal].