Chapter 10

Conclusions and Questions

Looking Backwards

The main themes shared by the two parts of this thesis are knowledge representation, and efficient automated reasoning on the chosen representation: Part I is concerned with a theoretical analysis of CSP algorithms, described through function iterations; Part II deals with efficient reasoning in the context of modal logics, by refining the way in which modal formulas are represented and passed to theorem provers. These parts are thus closer in rationale and methodology rather than in their contents. Below, we further motivate this claim by outlining the results in this thesis.

Back to Part I. This part describes and analyses a number of efficient algorithms for CSPs: the constraint propagation algorithms. Our aim, there, is purely theoretical: a unifying theory underpinning these algorithms, capable also to differentiate between them. Thus, in Chapter 3, we propose an algorithm schema, SGI, for constraint propagation algorithms. A simple theory for SGI is there developed. One of the primary objectives of our theorisation is declared to be the following (p. 28):

using SGI or some of its variations for describing and analysing how the prune-and-propagate process is carried through by constraint propagation algorithms.

Hence, in Chapter 4, different domains of functions (e.g., domain orderings) are related to different classes of constraint propagation algorithms (e.g., arc consistency algorithms); thus each class of constraint propagation algorithms is associated with a type of function domains, and so separated from the others. Then we analyse each such class: we distinguish functions on the same domains for their different ways of performing pruning (point or set based), and consequently
differentiate between algorithms of the same class (e.g., AC-1 and AC-3 versus AC-4 or AC-5).

Besides and foremost, we also correlate properties of functions (e.g., commutativity or stationarity) to different strategies of propagation in constraint algorithms (see, for instance, AC-1 versus AC-3), and suggest that these can be used for optimisation tasks.

In Chapter 5 the SGI schema is applied to soft CSP algorithms, thereby clarifying some of the similarities and differences between the crisp and soft constraint propagation algorithms. Also, properties of functions, e.g., monotonicity and inflationarity, are studied as separate issues in Chapter 3. Thus their respective roles in connection with certain behaviours of soft constraint propagation algorithms are differentiated. This is an achievement per se: in the soft constraint literature, often, the two properties are studied together and the role of each in the analysis of soft constraint propagation thus gets lost. Furthermore, we also obtain three new general conditions for the termination of semiring-based constraint propagation algorithms via this abstract approach.

Therefore, the adopted schema proves to be suitable for verifying constraint propagation algorithms, classifying them, comparing them, explaining and separating their properties. All these is done through the unifying framework of SGI function iterations. See also Table 4.1, p. 82.

Finally we characterise all the functions used for constraint propagation; in fact the other goal of our theorisation is (see Chapter 3):

abstracting which functions, iterated as in SGI or its variations, perform the task of pruning or propagation of inconsistencies in constraint propagation algorithms.

We accomplish this in Chapter 6, restricting the field of domain or constraint functions to those that are actually traced in the surveyed constraint propagation algorithms.

Back to Part II. In this part we shift perspective and approach, even though this part is concerned with relations and relational structures too, but in the context of modal logics. While the aim in the first part of this thesis is purely theoretical, in Part II our task is described as follows, see Subsection 7.1.1:

determining the satisfiability of modal formulas in an efficient manner.

In Chapter 8, we focus on one way of doing this: we refine the standard translation as the layered translation, and use existing theorem provers for first-order logic on the output of this refined translation. We provide ample experimental evidence on the improvements in performances that are gained through the refinement, see Section 8.5.

The refinement of the standard translation has strong semantic motivations: a strong form of the tree modal property. This property is also used in the basic
algorithm schema in Chapter 9. In fact, that property is behind the proposed algorithms based on constraint satisfaction methods. First modal formulas are encoded into propositional formulas and these into CSPs. The chosen constraint solver thus proceeds “layer by layer” in the encoding of the modal formula and in its candidate models, by applying a CSP solver for propositional satisfiability at each layer.

Chapter 9 brings us back to constraint algorithms, and apply them to modal reasoning problem. It constitutes a first attempt to tackle modal satisfiability by means of constraint propagation algorithms, as explained in Chapter 4, or various refinements of the basic backtracking schema for constraint satisfaction.

Back to the origins. A series of constraint propagation algorithms have been devised in the literature and only lately this kind of work has been accompanied by parallel work aiming at capturing the general principles behind those algorithms. In [Apt99a], the author studied constraint propagation through fair iterations of functions; see Section 4 in [Apt99a] for a complete overview of similar approaches. Later, in [Apt00a], the hypothesis of fairness was abandoned and the update operator took its place in a general algorithm schema for function iterations. We extended this in [Gen00] to explain the AC-4 algorithm of Mohr and Henderson [1986] and the AC-5 one of Van Hentenryck et al. [1992], which itself generalises several arc consistency algorithms. Our extension also explains the HAC-4 algorithm of Mohr and Masini [1988], i.e. the generalisation of AC-4 to n-ary constraints. In [Gen02], we provided another extension of the original schema to explain the PC-4 path consistency algorithm of Han and Lee [1988] and the KS algorithm of Cooper [1989], that can achieve either k-consistency or strong k-consistency. We also extended the theory of function iterations to soft constraints in the joint papers [BGR00, BGR02]. Part I of this thesis presents a unifying framework for all those generalisations, thus it explains all those algorithms and also the basic strong relational consistency algorithm of Dechter and van Beek [1997].

In Part II, the work in Chapter 8 is based on a joint paper [AGHdR00] with Areces, Heguiaberehere and de Rijke. This refined the standard translation of van Benthem [1983]. Our refinement there is motivated by semantic intuitions. These same intuitions trigger the work in Chapter 9: the design of a constraint solver for modal logics, based on the propositional solver of [GS00].

Looking Ahead

As the above section highlights, a number of questions follow up from this thesis. We briefly discuss some of them as below, starting with those related to CSPs and then passing to modal logics.
Chapter 10. Conclusions and Questions

Constraint Propagation

In the above section, we conclude that the SGI schema appears to be sufficiently general to abstract the common features of constraint propagation algorithms. Besides and foremost, the SGI schema is sufficiently expressive and ductile to allow us to distinguish each of those algorithms according to its specific strategy in iterating functions, so to speak. Certainly, a proof theoretic view of constraint propagation algorithms would be more general than the SGI based one. It seems thus natural to investigate whether such a proof theoretic view can be as expressive as the SGI based approach.

In the conclusions to Chapter 6, we suggest that it would be interesting to compare the given characterisation of constraint algorithms in terms of functions to the database relational model: in particular, this could be useful for optimisation tasks as our discussion on p. 122 pinpoints. We also suggest how this view can be extended to soft CSPs; we spotlight the problems that we face in this extension, and how these can be tackled.

In the conclusions to Chapters 4 and 5 we also suggest how the approach via functions, adopted in this thesis, could be useful in devising new algorithms, or new notions of constraint propagation; the basic algorithm schema can be SGI or a variation of it, the iterated functions should be polynomial-time computable. In what follows, we touch on this issue in the context of modal logics.

Diamond Satisfaction

A question emerges from the second part of this thesis: how efficiently constraint based algorithms can be applied to modal reasoning problems. The question can be investigated by working further on both the following issues:

- mapping modal formulas into CSPs,
- mapping inferences on modal formulas into constraint propagation and satisfaction steps.

For instance, efficient reasoning in this setting can be achieved by refining the encodings of modal formulas into CSPs, as proposed in Chapter 9; just like we do in Chapter 8 where we refine the standard translation of modal formulas into first-order formulas. In turn, these refined encodings could provide more information to constraint-based propagation and satisfaction algorithms, and this information could result in more efficient constraint-based algorithms.

The area of automated theorem proving for modal logics could also constitute a test-bed for soft constraint algorithms, as suggested in Section 9.7. In real time systems, modal formulas are used to express properties of the systems; in this context, the user is often interested in retrieving an optimal solution, that prioritises certain properties. Hence soft CSPs and algorithms may be employed in such situations, or existing algorithms may be optimised for these tasks.