Chapter 8

Conclusions and future work

In this concluding chapter, we recapitulate the contributions of the thesis and indicate several directions for further research.

8.1 Summary of contributions

The work described in this thesis concerns the (design) verification of concurrent systems, and of communication protocols in particular. Communication protocols, which constitute an important class of concurrent systems, were assumed to be specified explicitly in the well-established CFSM (communicating finite state machine) model [Boc78, Wes78, WZ78, ZW80, BZ83] as networks of finite-state machines that communicate asynchronously by sending and receiving messages over error-free FIFO queues. Concurrent systems at large were viewed more abstract as collections of concurrent, interacting sequential processes whose joint behavior can be formalized as a (labeled) transition system. Indeed, state transitions of concurrent systems in general may represent system events other than just message transmissions and receptions and, furthermore, processes may communicate not only by asynchronous message passing but also by synchronous “hand shaking”.

The focus of our research has been on improving strategies proposed earlier to relieve the state explosion problem which arises during the verification of concurrent systems and protocols by state space exploration, or reachability analysis. This was motivated by the awareness that, despite the merit of existing relief strategies, pursuing further performance improvements in verification remains utterly important. Indeed, concurrent systems are inherently complex and this complexity is here to stay (and grow). Based on a thorough review of the state of the art in verification, it became our particular objective to seek improvements of relief strategies that are based on state exploration. Such relief strategies inherit the simplicity of conventional reachability analysis (CRA), but reduce its complexity by examining only a fraction of the state space of a system. In effect, they enable the
verification of properties of concurrent systems without exploring all possible interleaving orders of concurrent events/transitions, which is one of the foremost causes of state explosion. Existing state exploration based relief strategies differ in the classes of systems they can handle, the types of properties they can verify, and the savings in space and time they can yield. The most advanced and promising state exploration based relief strategies in these respects are fair reachability analysis, simultaneous reachability analysis and partial-order reduction methods. In this thesis, we have proposed an incremental improvement of each of these three relief strategies so as to broaden their applicability to yet more complex and larger concurrent systems and protocols. The potential practical impact of our contributions is that more realistic industrial-strength systems may become amenable to automated verification.

Fair reachability analysis (FRA) is a relief strategy which was first proposed for the verification of logical correctness properties of two-process protocols specified in the CFSM model [RW82, GH85]. It was recently generalized to cyclic protocols, in which two or more processes form a unidirectional ring [LM94, LM96]. We have further generalized the technique of FRA to so-called multi-cyclic protocols (Chapter 4) [SU95a, LM96]. A multi-cyclic protocol consists of any number of unidirectional rings, or component cyclic protocols, which are interconnected such that no two rings share more than one process. The class of multi-cyclic protocols has an interestingly wide applicability in practical protocol modeling. It captures not only protocols with a multi-ring topology (of which the two-process and cyclic protocols studied in [RW82, GH85, LM94, LM96] are special cases), but also protocols with other regular network topologies like a daisy-chain, a star or a tree. Moreover, any combination of these elementary topologies is allowed as long as no two rings in the resulting protocol have more than one process in common.

As for cyclic protocols [RW82, GH85, LM94, LM96], FRA in its basic form is effective and efficient for the detection of deadlocks in multi-cyclic protocols. The fair reachable global state space of a multi-cyclic protocol explored by FRA generally constitutes only a very small fraction of the complete state space of the protocol explored by CRA (cf. Table 6.4). Indeed, through proper formulation we established that it entails just those reachable global states of a multi-cyclic protocol in which for each ring all channels in the ring are of equal length. This so-called ring-wise equal channel length property captures in particular all deadlock states. As a result, FRA decides the deadlock detection problem for every multi-cyclic protocol whose fair reachable global state space is finite. We also determined two sufficient conditions for finiteness that relate to the boundedness aspect of channels (see propositions 4.39 and 4.41). Both these conditions allow the presence of unbounded channels, which indicates that FRA is important not only as a relief strategy but also as a state exploration technique capable of handling various unbounded (multi-cyclic) protocols. The ring-wise equal channel length property and the boundedness conditions for multi-cyclic protocols generalize the equal channel length property and the respective boundedness conditions given in
[LM94, LM96] for cyclic protocols. This generalization proved necessary because multi-cyclic protocols that are not cyclic (i.e. those that are composed of multiple rings) have “connector” processes with more than one incoming and one outgoing channel. A next step along this line would be an extension of FRA to protocols with yet more complex and perhaps even arbitrary communication topologies. However, we established that FRA is in fact infeasible beyond multi-cyclic protocols. More accurately, its effectiveness is limited to protocols that are fair-formed (see Definition 4.49). This negative result stems from the principle characteristics of FRA, forcing progress of at least two processes at each step during state exploration while preserving a global channel invariant.

Like FRA, simultaneous reachability analysis (SRA) employs the concept of executing multiple transitions in a single atomic step to reduce the number of global states and transitions explored, and thereby the space and time needed for verification. However, unlike FRA, SRA can be used to verify logical correctness properties of protocols specified in the CFSM model with totally arbitrary communication topologies [ÖU95, Özd95]. In essence, this generality was achieved by allowing processes in a protocol to progress concurrently (or simultaneously) in a more flexible way than FRA. Notwithstanding its novelty, we have proposed an incremental improvement of SRA which enables further savings in space and time for protocol verification (Chapter 5) [SU96b, SU98a]. This improvement, named leaping reachability analysis (LRA), governs the execution of sets of concurrent transitions (i.e. leap sets) similar as SRA to verify the absence of non-progress states (including deadlocks), non-executable transitions, unspecified receptions and buffer overflows in a protocol. Through an analytical comparison we established that, for every protocol, the fraction of the protocol state space explored by LRA is contained in the fraction of the state space explored by SRA, for each of these four logical correctness properties. That is, LRA never explores more global states and transitions than SRA. Moreover, LRA never incurs more run-time overhead and even eliminates the need for a protocol augmentation as required by SRA for detecting unspecified receptions. LRA is thus an absolutely risk-free improvement of SRA, i.e. using LRA instead of SRA for verifying logical correctness properties of protocols in the CFSM model is at no cost whatsoever, neither in space nor in time. This is a notable result by itself, since all too often one is confronted with a trade-off between space and time. Especially when attempting to improve the performance of an already efficient verification technique, extreme care must be taken to ensure that potential extra savings in space are not attended by unacceptable expenses in time.

The increased efficiency in space and time of LRA over SRA stems mainly from the fact that LRA selects fewer leap sets (called selected simultaneously executable sets in [ÖU95, Özd95]) than SRA for execution at any given global state $G$. The leap sets in $G$ that are executed by LRA are conditioned to contain at most one executable transition from among the processes with potentially executable transitions at $G$ (i.e. transitions that are not executable at $G$ but may become executable
later at a global state reached from $G$), while those executed by SRA are allowed to have multiple executable transitions from such processes. Simple combinatorics attest that the number of extra leap sets executed in $G$ by SRA is exponential in the number of processes with both executable and potentially executable transitions at $G$ (see Section 5.6.1). In general, LRA can therefore be expected to employ (i.e. compute and execute) a significantly smaller number of leap sets during state exploration than SRA, especially for protocols whose state spaces manifest a relatively wide distribution of potentially executable transitions. We have complemented this rational finding with an empirical comparison of the performance of LRA and SRA on a large number of protocols (Chapter 6). Based on their implementation in the research tool package RELIEF [Özd95, Ngu97], LRA and SRA were tested on a set of 400 impartial sample protocols constructed with the automatic protocol synthesiser in RELIEF [ÖU95, Öz95], and on three real protocols from the literature. The experiments revealed that, overall, LRA is able to yield important incremental extra savings over SRA in space and especially in time. Remark that these extra savings should indeed be expected to be “only” incremental, since SRA itself can already yield significant savings. For the detection of non-progress states in a protocol, both the space and time savings by LRA over SRA can be quite substantial (cf. Table 6.2 and Table 6.3). For the detection of non-executable transitions, unspecified receptions and buffer overflows, the extra space savings by LRA turn out to be rather modest, but the extra time savings can still be very good (cf. Table 6.5-6.7). Add thereto that we also offered an optional refinement of LRA which exploits the special characteristics of a depth-first search (see Section 5.5). This refinement (called LRA2 in Chapter 6) enables more discrete space savings over SRA for the detection of non-executable transitions and unspecified receptions, and with just little extra computational overhead (cf. Table 6.5 and Table 6.6). Based on our analytical and empirical comparisons, and in view of the notorious state explosion problem, we may certainly conclude that LRA is a worthwhile improvement of SRA as a relief strategy for protocol verification.

Partial-order reduction methods [God90, Val90, HGP92, KP92a, Val92, Val93, GW93, GW94, HP95, God96, Pel96] are a collection of cognate state exploration techniques set to relieve the state explosion problem for the verification of (finite-state) concurrent systems in general. That is, these methods are largely independent of the model used for specifying concurrent systems. They are pertinent in principle to all specification models whose semantics induce labeled transitions systems [HP95, God96], including the CFSM model. Partial-order reduction methods have proved effective and efficient for verifying local and termination properties (e.g. deadlock-freedom and freedom non-executable transitions) and, moreover, for verifying linear-time temporal logic (LTL) properties of concurrent systems. This is known as LTL model-checking, and captures arbitrary (temporal) safety and liveness properties. To achieve space and time reduction for verification, partial-order
reduction methods aim at exploring just one fixed order out of all possible interleaving orders of concurrent independent transitions at a given global state, by executing at each step during state exploration only a selective subset of the transitions executable at the current state, rather than all of them (as is done in CRA). We have shown how to combine this idea with the concepts underlying LRA to yield an approach which enables further reductions in space and time for LTL model-checking in the general context of finite-state concurrent systems that can be formalized as labeled transition systems (Chapter 7) [Sch97, SU98b]. In particular, we have proposed an enhancement of the partial-order reduction method based on ample sets as described in [HP95, Pel96]. This method, which we referred to as POVAS (Partial Order Verification with Ample Sets), was chosen because it is generic in the sense that it can be readily adapted to capture other partial-order reduction methods (i.e. those based on persistent sets or stubborn sets), and because it is advocated as the most advanced in terms of the properties it can check, the way it deals with fairness, and the low overhead and high overall performance of its implementation [HP95, Pel96]. In essence, instead of exploring some fixed interleaving order among concurrent independent transitions at global states, as does POVAS or any other partial-order reduction method on the basis of ample sets, persistent sets or stubborn sets, our enhanced approach abstains whenever possible from any order altogether by executing leap sets that mimic truly concurrent executions of such transitions.

We have further shown how to realize POVAS and its proposed enhancement specifically in the context of the CFSM model. In particular, we determined that the general notion of dependency among transitions can be captured efficiently within the CFSM model in terms of the notion of potentially executable transitions. This made it possible to incorporate our enhancement of POVAS for LTL model-checking into the formulation of LRA, thereby establishing LRA as a uniform relief strategy for the verification of both logical (or syntactic) and functional (or semantic) correctness properties of protocols specified in the CFSM model. That is, given a protocol $\mathcal{P}$, exploring its state space by LRA is based uniformly on the notion of “leaping” and varies only with the property to be verified. The checked property induces the designated subset of leap sets in leap($G$) to be executed at each global state $G$ encountered during state exploration, i.e. pleap($G$) for verifying indefinite progress and xpleap($G, J, K, V$) for verifying freedom of non-executable transitions, unspecified receptions wrt $J$, buffer overflows wrt $K$, and any LTL formula $f$ with $\text{vis}_T(\mathcal{P}) \models V$ (see Section 7.4). The resulting savings in space and time are commensurate with the complexity of the property. The experiments performed with POVAS and LRA for (off-line) LTL model-checking in the CFSM model, based on their implementation in the research tool package RELIEF, indicated that LRA can indeed yield considerable extra savings in space and time over POVAS. Hence, it widens the applicability of LTL model-checking to more complex and larger concurrent systems and protocols.
8.2 Future work

Naturally, several open problems arise from our investigations. These problems are discussed below, together with some other suggestions for future research.

FRA for multi-cyclic protocols beyond deadlock detection

As explained in Section 4.5, FRA in its basic form is inadequate for the detection of logical errors other than deadlocks, mainly because it does not ensure the exposure of all reachable process states of the different processes in a multi-cyclic protocol. At least a finite extension of the fair reachable global state space of a multi-cyclic protocol is thus needed to provide a more comprehensive logical error coverage. Such an extension is already in force for cyclic protocols. Specifically, for the class of cyclic protocols with a finite fair reachable global state space, a procedure has been proposed in [LM94b, LM96] that finitely augments this reduced state space for deciding (un)boundedness, freedom of non-executable transitions and freedom of unspecified receptions. In [LM95], three more reachability problems were solved in a similar way, viz. global state reachability, abstract state reachability and execution cycle reachability. We have described the extension procedure for cyclic protocols in detail, and argued that it is unfit for generalization to multi-cyclic protocols due to the possible interaction dependencies that may arise among processes in different rings in these protocols. Unfortunately, we have not succeeded in devising an alternative extension procedure for the class of multi-cyclic protocols with a finite fair reachable global state space. We did sketch an argument suggesting that logical correctness properties other than deadlock-freedom are in fact undecidable in general for this class by FRA plus finite extension, but this argument is informal by all means. Strictly speaking, it is therefore still open whether FRA can be used to achieve the same logical error coverage for multi-cyclic protocols as for cyclic protocols. We are left with either finding a finite extension procedure suitable for multi-cyclic protocols, or proving formally that such a procedure cannot exist.

Of importance in this respect is also the characterization established in [LM94b, LM96] of the logical correctness of a cyclic protocol: a cyclic protocol is free from logical errors if and only if its fair reachable global state space is free from logical errors. In other words, every logical error within the reachable global state space of a cyclic protocol implies the existence of a logical error (not necessarily the same) within the fair reachable global state space of the protocol. This offers the advantage of iterative verification: one can verify a cyclic protocol correct by repeatedly applying FRA, fixing errors after each single run, until no more logical errors are found. The time required for verification may of course increase substantially, but it can still be a practical way to circumvent
memory shortage. Establishing whether or not a similar result holds for multi-cyclic protocols is clearly beneficial.

Identifying classes of unbounded protocols with decidable properties

It is certainly of theoretical interest to identify classes of infinite-state systems with decidable properties (cf. Section 2.5.1). Both FRA and LRA turned out to be capable of deciding logical correctness properties for various such unbounded protocols in the CFSM model, but a complete characterization of these protocols remains to be determined. Regarding FRA, it seems that the existence of such a characterization for unbounded multi-cyclic protocols goes hand in hand with the ability to use FRA beyond deadlock detection. Indeed, in relation to the discussion above, the notion of weak boundedness (see Definition 4.40) is a necessary and sufficient condition for cyclic protocols to have a finite fair reachable global state space [LM94a, LM96], but not so for multi-cyclic protocols in general (see Figure 4.5). We speculate that certain structural conditions on the process graphs of individual processes may ultimately provide a complete characterization of the classes of unbounded (multi-cyclic) protocols amenable to FRA and LRA, respectively, but this requires further investigation.

More state reduction

The amount of state reduction obtained by LRA (and likewise by SRA and partial-order reduction methods) relies on the number of dependencies between the transitions (i.e. the coupling among the processes) of a concurrent system or protocol. The more dependencies, the more LRA degrades to conventional reachability analysis, thus yielding less reduction. Since the exact dependency relation between transitions is generally too hard to determine, in practice one must employ some upper approximation of this relation that can be easily computed. Such approximated dependency relation may then still be refined to obtain fewer dependencies and thus more state reduction at the expense of extra computational overhead (see e.g. [KP92b, Val92, GP93, God96]). For protocols in the CFSM model we implicitly used an approximated dependency relation, by way of the notion of potentially executable transitions. This notion is currently defined on the basis of global state information only, but it seems possible to refine it by exploiting the structural characteristics of individual processes. For instance, in FRA, certain potentially executable transitions are further classified as enabled transitions. Enabled transitions provide the means to look one step ahead during state exploration. FRA indeed benefits from this, as witnessed by Table 6.4: for various multi-cyclic protocols it can yield much better space reductions than LRA, without incurring totally unacceptable time penalties.
Alternative ways to improve the efficiency of LRA, and state exploration techniques in general, may be found by tackling other causes of the state explosion problem (i.e. other than the modeling of concurrency by interleaving), such as variables whose values range over a large and possibly infinite domain. For instance, studying the possibility of combining symbolic verification techniques [BC+90, Bry92, HD93, McM93], and structural or functional decomposition techniques [VC82, CM83, LS84, CGL85, CM86] with LRA is certainly worthwhile. On a different note, implementing LRA in conjunction with the bit-state hashing [Hol88, Hol90, Hol91] and/or state space caching [Hol85, Hol87, JJ91, GHP92] disciplines will further increase its efficiency as well, as already discussed in Section 3.6.

**Verifying other properties with LRA**

So far, we have developed LRA as a(n) (improved) relief strategy for verifying logical correctness properties, and for model-checking LTL properties of concurrent systems and protocols, including arbitrary safety and liveness properties. This covers many of the properties that will ever be verified in practice. Still, it would be commendatory to extend the scope of LRA to other types of properties, like properties expressed in branching-time temporal logics [Eme90, BVW94]. Several results on adapting partial-order reduction methods for branching-time temporal logic model-checking have already been reported [GK+95, WW96]. The work in [GK+95] follows POVAS very closely, and an improvement of this work in terms of an extension of LRA for branching-time temporal logic model-checking is therefore imminent. Along the same line of thought, another important direction for further research is the verification of “real-time” and “probabilistic” properties of systems specified in models that involve a quantitative notion of time. A large body of work is currently underway to develop efficient techniques for this purpose (see e.g. [PRO98]).

**Other applications**

The state explosion problem is a limiting factor not only in verification, but also in areas such as protocol synthesis [PS91] and protocol conversion [CL90], as well as in many other applications of computer science and engineering. Any technique that tackles the state explosion problem in a systematic manner may therefore be useful beyond verification. In particular, we think that the leaping concept underlying LRA can be set to work for any problem that can be reduced to a state exploration (or search) problem and that exhibits some form of concurrency.