Spotlight Abstraction of Agents and Areas (Extended Abstract)

Tobe Toben[1], Bernd Westphal[2] and Jan-Hendrik Rakow[3]

[1] OFFIS e.V., 26121 Oldenburg, Germany, toben@offis.de
[2] Albert-Ludwigs-Universität Freiburg, 79085 Freiburg, Germany,
westphal@informatik.uni-freiburg.de
[3] Carl von Ossietzky Universität Oldenburg, 26111 Oldenburg, Germany
jan.rakow@informatik.uni-oldenburg.de

We address the problem of automatically analysing systems that vary dynamically in size and topology. Typical examples of such systems include adhoc networking [1] where a routing infrastructure over a changing set of participants is created and maintained. Similar structures occur in dynamic traffic management systems like the car platooning scenario [2] where physically adjacent cars establish interlinked groups. We treat the inherent unboundedness of the state space by applying a finitary abstraction called *spotlight abstraction* [3,4], which is based on the data-type reduction approach [5]. The abstraction principle is heterogeneous in the sense that the behaviour of a finite number of agents is preserved while the others are only abstractly represented. Other abstraction techniques which address a similar class of systems include counter abstraction [6,7], shape abstraction [8,9], partner abstraction [10,11], indexed predicate abstraction [12] and environment abstraction [13].

Let Id be a set of process identities and $\mathcal{P}$ a set of predicates. A system state is given by an interpretation of the predicates on the set of identities, i.e.

$$\iota : \mathcal{P} \times Id^K \to \mathbb{B} \tag{1}$$

assigns a boolean value to each K-ary predicate and K-tuple of identities. For example, unary predicates can be used to determine the current local states of processes, and binary predicates characterise the actual connection topology among processes. The behaviour of a system is then given in terms of a transition system over interpretations, that is, each state in the transition system corresponds to one interpretation, and the transition relation determines the possible changes in the interpretations over time.

A spotlight is a (typically finite) set of process identities $S \subseteq Id$. The spotlight abstraction of a system state preserves the predicate interpretation of the spotlight processes S and abstracts from the rest. Formally, the abstraction transforms two-values interpretations into three-valued interpretations

$$\alpha_S(\iota)(p, a_1, \dots, a_n) := \begin{cases} \iota(p, a_1, \dots, a_n) & \textit{if } \{a_1, \dots, a_n\} \subseteq S \\ {}^1\!/\!_2 & \textit{else} \end{cases} \tag{2}$$

Dagstuhl Seminar Proceedings 10051
Quantitative and Qualitative Analysis of Network Protocols
http://drops.dagstuhl.de/opus/volltexte/2010/2517

where $1/2$ represents the indefinite value from three-valued boolean logic [14]. For spotlight abstraction, the abstract transition system can typically be obtained by a simple syntactical transformation of the system description [15].

The content of the spotlight is first of all derived from the property specification to be verified. We use first-order variants of temporal logic [16] and employ query reduction [17,15] to obtain a finite set of representative[1] valuations of the quantified variables in the specification. This yields a finite set of finite verification tasks with the spotlight comprising the range of the actual valuation function. Abstraction refinement is done by adding new processes to the spotlight and restricting the behaviour of the non-spotlight part of the abstraction [18,4,19]. Using this methodology, a number case studies have been conducted, for example car platooning [20,4,18], railway systems [19,21], and client/server systems [19].

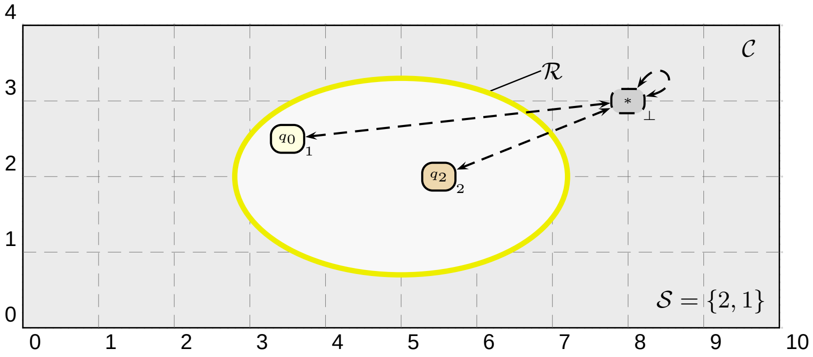

Fig. 1. Spotlight Abstraction of the area $\mathcal{R}$ on space $\mathcal{C}$. Only the information for identities 1 and 2 are preserved, the other processes are collapsed into one summary process $\perp$. Still, process $\perp$ may communication with the spotlight processes and new processes may appear in the spotlight area.

As a *new way* to determine spotlight candidates, we propose to investigate systems where the physical position of agents is relevant. Let $\mathcal{C}$ be a characterisation of some metric space (e.g. a subset of $\mathbb{R}^2$), we now assume that an interpretation ι in particular determines the actual position of an agent a, denoted by $\mathsf{pos}_\iota(a) \in \mathcal{C}$. We propose to analyse the behaviour of physically adjacent agents by letting the spotlight comprise agents within a certain area $\mathcal{R} \subseteq \mathcal{C}$. So instead of a fixed set of spotlight processes, we obtain a spotlight

$$S_{\mathcal{R},\iota} = \{a \in Id \mid \mathsf{pos}_\iota(a) \in \mathcal{R}\} \tag{3}$$

that depends on the area and varies with the interpretation. Note that the abstraction principle as given in equation (2) can be reused without modifications.

[1] Given that the transition system is symmetric in identities [15]

The basic principle of area spotlight abstraction is visualised in Fig. 1. For choosing the area $\mathcal{R}$ we observe different possibilities:

1. Use a fixed area that focusses on critical zones of the underlying motion space, e.g. junctions or highway exits. This establishes properties of the kind “for all agents $a \in \mathit{Id}$ there is no collision in the considered area $\mathcal{R}$”.
2. For a symmetric space one may be able to determine a *representative* area $\mathcal{R}$. Then, if both the behaviour of the agents is independent of its identity, and the focussed area $\mathcal{R}$ is representative for the overall space $\mathcal{C}$, one may establish properties like “for all agents $a \in \mathit{Id}$ there is no collision on $\mathcal{C}$”.
3. Use a varying area depending on the positions of some ego agents $\mathit{Ego} \subset \mathit{Id}$

$$\mathcal{R}_{\mathit{Ego},\iota,r} = \{x \in \mathcal{C} \mid \exists\, a \in \mathit{Ego} : d(\mathsf{pos}_\iota(a), x) \leq r\} \tag{4}$$

where d is the metric on $\mathcal{C}$ and r is a suitable characterisation of a radius. Intuitively, each ego agent determines a certain spotlight area with the agent being at the center. Whenever an ego agent moves, the spotlight area moves accordingly. Note that while the set of ego agents is fixed, other agents may enter the spotlight part and become spotlight processes.

With equation (3), the usage of a moving area according to (4) immediately yields

$$\mathit{Ego} \subseteq S_{\mathcal{R}_{\mathit{Ego},\iota,r},\iota} \tag{5}$$

for any ego agents Ego, interpretation ι and radius r. That is, the ego agents are always in the spotlight.

Future work will transfer the notion of spotlight abstraction refinement to area abstraction as described above. We observe two basic possibilities of spurious interactions, namely (i) communication interference and (ii) materialisation. The first issue relates to spurious messages from the abstracted part of the system to the concrete part. This problem has already been addressed in [22,20], however the information concerning the physical position of agents may improve the existing solutions. Materialisation corresponds to the fact that new spotlight processes may appear dynamically when they enter the focussed area $\mathcal{R}$. It will be necessary to derive suitable constraints from the underlying dynamics of the agents in order to obtain a meaningful and sound notion of materialisation.

References

1. Frodigh, M., Johansson, P., Larsson, P.: Wireless Ad Hoc Networking: The Art of Networking without a Network. Ericsson Review **4** (2000)
2. Varaiya, P.: Smart cars on smart roads: problems of control. IEEE Transactions on Automatic Control **38** (1993) 195–207
3. Wachter, B., Westphal, B.: The Spotlight Principle. In Cook, B., Podelski, A., eds.: VMCAI. Volume 4349 of LNCS., Springer (2007) 182–198

4. Toben, T.: Counterexample Guided Spotlight Abstraction Refinement. In Suzuki, K., Higashino, T., Yasumoto, K., El-Fakih, K., eds.: FORTE. Volume 5048 of LNCS., Springer (2008) 21–36
5. McMillan, K.L.: Verification of infinite state systems by compositional model checking. In Pierre, L., Kropf, T., eds.: CHARME. Volume 1703 of LNCS., Springer (1999) 219–234
6. Lubachevsky, B.D.: An Approach to Automating the Verification of Compact Parallel Coordination Programs I. Acta Inf. **21** (1984) 125–169
7. Pnueli, A., Xu, J., Zuck, L.D.: Liveness with (0, 1, ∞)-Counter Abstraction. In Brinksma, E., Larsen, K.G., eds.: CAV. Volume 2404 of LNCS., Springer (2002) 107–122
8. Sagiv, S., Reps, T.W., Wilhelm, R.: Parametric shape analysis via 3-valued logic. ACM Trans. Program. Lang. Syst. **24** (2002) 217–298
9. Wies, T., Kuncak, V., Zee, K., Podelski, A., Rinard, M.C.: On Verifying Complex Properties using Symbolic Shape Analysis. CoRR **abs/cs/0609104** (2006)
10. Bauer, J.: Analysis of Communication Topologies by Partner Abstraction. PhD thesis, Universität des Saarlandes (2006)
11. Bauer, J., Wilhelm, R.: Static Analysis of Dynamic Communication Systems by Partner Abstraction. In Nielson, H.R., Filé, G., eds.: SAS. Volume 4634 of LNCS., Springer (2007) 249–264
12. Lahiri, S.K., Bryant, R.E.: Constructing Quantified Invariants via Predicate Abstraction. In Steffen, B., Levi, G., eds.: VMCAI. Volume 2937 of LNCS., Springer (2004) 267–281
13. Clarke, E.M., Talupur, M., Veith, H.: Environment Abstraction for Parameterized Verification. In Emerson, E.A., Namjoshi, K.S., eds.: VMCAI. Volume 3855 of LNCS., Springer (2006) 126–141
14. Kleene, S.C.: Introduction to metamathematics. Bibliotheca Mathematica. North-Holland, Amsterdam (1952)
15. Westphal, B.: Specification and Verification of Dynamic Topology Systems. PhD thesis, Carl von Ossietzky Universität Oldenburg, Germany (2008)
16. Pnueli, A.: The Temporal Logic of Programs. In: Proceedings of the 18th Annual Symposium on Foundations of Computer Science, Providence, Rhode Island, USA, IEEE (1977) 46–57
17. Ip, C.N., Dill, D.L.: Better verification through symmetry. Formal Methods in System Design **9** (1996) 41–75
18. Bauer, J., Toben, T., Westphal, B.: Mind the Shapes: Abstraction Refinement Via Topology Invariants. In Namjoshi, K.S., Yoneda, T., Higashino, T., Okamura, Y., eds.: ATVA. Volume 4762 of LNCS., Springer (2007) 35–50
19. Toben, T.: Analysis of Dynamic Evolution Systems by Spotlight Abstraction Refinement. PhD thesis, Carl von Ossietzky Universität Oldenburg, Germany (2009)
20. Toben, T.: Non-Interference Properties for Data-Type Reduction of Communicating Systems. In Davies, J., Gibbons, J., eds.: IFM. Volume 4591 of LNCS., Springer (2007) 619–638
21. Westphal, B.: LSC Verification for UML Models with Unbounded Creation and Destruction. Electr. Notes Theor. Comput. Sci. **144** (2006) 133–145
22. Damm, W., Westphal, B.: Live and let die: LSC based verification of UML models. Sci. Comput. Program. **55** (2005) 117–159