行政院國家科學委員會專題研究計畫　成果報告

模組化設計之合成與參數化驗證分析(I)

計畫類別：整合型計畫
計畫編號：NSC93-2213-E-002-080-
執行期間：93 年 08 月 01 日至 94 年 07 月 31 日
執行單位：國立臺灣大學電機工程學系暨研究所

計畫主持人：顏嗣鈞

計畫參與人員：盧紀憲　陳建良 王紹中 余宗恩

報告類型：精簡報告

處理方式：本計畫可公開查詢

中　華　民　國 94 年 11 月 1 日

行政院國家科學委員會專題研究計劃成果報告

模組化設計之合成與參數化驗證分析

計畫編號: 93-2213-E-002-080
執行期間:93 年 8 月 1 日至 94 年 7 月 31 日
計畫主持人: 顏嗣鈞 教授 國立台灣大學電機工程系

一、摘要

本年度之研究計畫中，我們研究無限狀態系統(infinite-state systems) 之「自穩定性問題」(the self-stabilization problem)。 一般而言，此問題為不可解。我們探討三類可解之情形: lossy vector addition systems with states, one-counter machines, and conflict-free Petri nets。我們並發展新的驗證技術來解決以上問題。

關鍵字：無限狀態系統，自穩定性問題，驗證

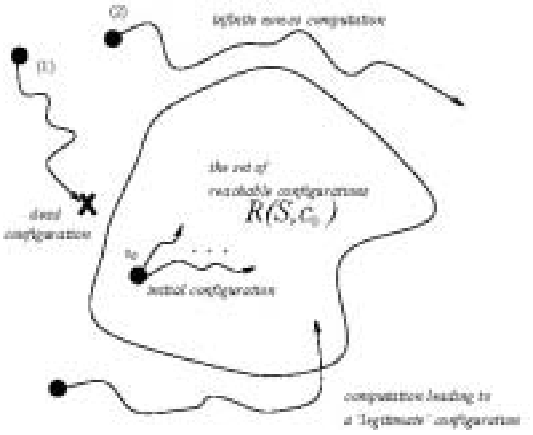
For a variety of infinite-state systems, the problem of deciding whether a given system is self-stabilizing or not is investigated from the decidability viewpoint. We develop a unified strategy through which checking self-stabilization is shown to be decidable for lossy vector addition systems with states, one-counter machines, and conflict-free Petri nets. For lossy counter machines and lossy channel systems, in contrast, the self-stabilization problem is shown to be undecidable.

Keyword: infinite-state system, self-stabilization, verification

二、計畫緣由與目的

The notion of *self-stabilization* was introduced by Dijkstra to describe a system having the behavior that regardless of its starting configuration, the system would return to a `legitimate' configuration eventually (by a legitimate configuration, we mean a configuration which is reachable from the initial configuration of the system). The motivation behind self-stabilization is that a self-stabilizing system has the ability to `correct' itself even in the presence of certain unpredictable errors that force the system to reach an `illegitimate' configuration during the course of its operations. In this sense, self-stabilizing systems exhibit fault-tolerant behaviors to a certain degree. With the increased interest in designing fault-tolerant systems, the study of self-stabilization has been gaining increasing popularity in the computer science community lately.

Intuitively speaking, a system is said to be *self-stabilizing* (*ss*, for short) if, regardless of its starting configuration, the system would eventually return to a `legitimate' configuration which is reachable from the initial configuration of the system. That is, a self-stabilizing system has the ability to `correct' itself even in the presence of certain unpredictable errors that force the system to reach an `illegitimate' configuration during the course of its operations. Let S be a (finite or infinite) system with c0 as its initial configuration. A system is said to be self-stabilizing if for each configuration c, none of the computations emanating from c is non-ss. The self-stabilization problem is to determine, for a given (finite or infinite) system, whether the system is self-stabilizing. In this paper, our main concern is to investigate the decidability issue of the self-stabilization problem for a variety of infinite-state systems.



三、計畫方法

A system S is not self-stabilizing iff either (i) a finite computation ends up with a dead configuration being not reachable from the initial configuration, or (ii) an infinite non-ss computation exists. (i) is relatively easy to check as long as certain properties of S are decidable. The idea behind our approach of checking (ii) is built upon showing that to demonstrate the non-ss nature of a system, it is sufficient to search among only non-ss computations of periodic behaviors, and hopefully, the confinement to such computations admits a decidable checking of (ii). By non-ss periodic computations we mean those non-ss computations of the form s $\xrightarrow{\pi\pi\pi\ldots}$, i.e., repeating π from s infinitely many times witnesses non-ss.

1. (Strongly periodic:) Every finite computation c $\xrightarrow{\pi}$ c' with c' ≥ c and c ∉ R(S, c0) ensures non-ss of its infinite repetition (i.e., c $\xrightarrow{\pi\pi\pi\ldots}$ is non-ss).

2. (Weakly periodic:) If infinite non-ss computations exist, there must be one of the form c $\xrightarrow{\pi\pi\pi\ldots}$.

The disparity between strongly and weakly periodic non-ss computations is that in the former, a finite computation of the form `c $\xrightarrow{\pi}$ c' with c' ≥ c and c ∉ R(S, c0)' is guaranteed to conclude the system's non-ss behavior, whereas in the latter, however, not *every* finite computation of the above form is extendible to render ss; nevertheless, there does *exist* a periodic witness.

As far as we know, very little is known regarding the complexity of the self-stabilization problem (i.e., the problem of deciding whether a system is self-stabilizing or not) for infinite-state systems. At this moment, the best bounds of the problem for general Petri nets are a lower bound of exponential space and an upper bound of Π_2 (the second level of the arithmetic hierarchy), whereas it is Π_2-complete for Turing machines. Therefore, it is of interest and importance to take a closer look at the problem for other infinite-state systems, in the hope of recognizing the key characteristics which govern the decidability/undecidability nature of the problem. An equally important goal is to, perhaps, come up with a unified framework through which decidability/undecidability of self-stabilization can be obtained. In this paper, we investigate, from the decidability viewpoint, the problem of deciding whether a given system is self-stabilizing for a wide range of infinite-state systems, including lossy vector addition systems with states, one-counter machines, conflict-free Petri nets, lossy counter machines, and lossy channel systems. As it turns out, the decidability of self-stabilization for lossy vector addition systems with states, one-counter machines, and conflict-free Petri nets can be established in a unified setting, taking advantage of the existence of a periodic witness for non-self-stabilizing infinite computations. For lossy counter machines and lossy channel systems, however, the self-stabilization problem turns out to be undecidable.

四、結論與未來展望

We have studied, from the decidability viewpoint, the problem of determining whether a given system is self-stabilizing or not for a variety of infinite-state systems. To this end, we have developed a unified strategy through which the problems for one-counter machines, conflict-free Petri nets, and lossy vector addition systems with states were shown to be decidable. The key behind our strategy is that for each of the three classes of systems, the reachability set from an arbitrary configuration is effectively semilinear, and it suffices to consider infinite non-self-stabilizing computations with periodic behaviors.

It is interesting to see whether our strategy can be applied to a wider class of infinite-state systems, or a similar strategy (without relying on the reachability set being semilinear) can be extended to systems not enjoying the semilinearity property. An equally important direction of future research is to find out the exact complexity of the self-stabilization problem for the above classes of systems. Finally, self-stabilization for real-time systems deserves further investigation, as many real-world systems are of real-time nature. For the model of timed automata, the region graph technique can easily be applied to showing the decidability (in fact, in PSPACE) of self-stabilization (in the sense defined in this paper). However, to cope with the nature of real-time systems, one might have to tailor the notion of self-stabilization to better capture the intuitive concept of `self-stabilizing in a certain amount of time,' as opposed to `reaching a legitimate configuration eventually' as defined in the conventional sense.

五、參考文獻

[1] Bouajjani, A. and Mayr, R., Model checking lossy vector addition systems, *Proc. of STACS'99*, LNCS 1563, 323-333, 1999.

[2] Dijkstra, E. Self-stabilizing systems in spite of distributed control, *C. ACM*, 17:643-644, 1974.

[3] Herman, T., A comprehensive bibliography on self-stabilization, *Chicago Journal of Theoretical Computer Science*, (http://www.cs.uiowa.edu/ftp/selfstab/bibliography), 1998.

[4] Mayr, R., Undecidable problems in unreliable computations, *Theoret. Comput. Sci*. 1-3(297):337-354, 2003.

[5] Presburger, M., Uber die vollstandigkeit eines gewissen systems der arithmetik ..., Comptes rendues du premier Congres des Math. des Pays Slaves, Warsaw, 92-101, 395, 1929.

[6] Yen, H., Wang, B. and Yang, M., Deciding a class of path formulas for conflict-free Petri nets, *Theory of Computing Systems*, 30(5):475-494, 1997.