Control Engineering Practice 19 (2011) 947–957

Contents lists available at ScienceDirect

Control Engineering Practice

journal homepage: www.elsevier.com/locate/conengprac

Translating Grafcet specifications into Mealy machines for conformance test purposes

Julien Provost *, Jean-Marc Roussel, Jean-Marc Faure

LURPA, ENS Cachan, 61, av du Président Wilson, Cachan F-94230, France

ARTICLE INFO

Article history:
Received 30 November 2009
Accepted 6 October 2010
Available online 15 December 2010

Keywords:
Conformance test
Model-based test
Grafcet
Mealy machine
Logic controllers

ABSTRACT

Conformance test is a black-box test technique aiming at checking whether an implementation conforms to its specification. Numerous results have been already obtained in this field for specifications expressed in a formal language. However, these results cannot be applied for conformance test of industrial logic controllers whose specifications are given in standardized specification languages. To contribute to solve this issue, this paper proposes a method to obtain, from a Grafcet specification, an equivalent Mealy machine, without semantics loss. This method permits to describe explicitly and formally all the states and transitions that are implicitly represented in a Grafcet model.

1. Introduction

Logic controllers are increasingly used in critical systems, like power production and distribution systems or transport systems, even for safety-related functions. To ensure dependability of these systems, it really matters to check, before operation, whether each controller behaves correctly with respect to its specification. This is the aim of conformance test. Conformance test is a black-box test and is experimentally performed (Fig. 1) by sending to the controller an input sequence and comparing the observed output sequence, controller's response to the input sequence, to the expected output sequence so as to build a test verdict (the implemented controller is conform or not). The set of the input sequence and expected output sequence is termed test sequence or test case.

Numerous theoretical results have been published in the domain of conformance test, assuming that the specification is formally described, for instance in the form of a finite state machine (da Silva Simão, Petrenko, & Yevtushenko, 2009; Lee & Yannakakis, 1996), a transition system (Tretmans, 2008) or, more recently, a particular class of Petri net (von Bochmann & Jourdan, 2009). Generally speaking, these results provide a way to build automatically the test sequence from the formal specification model and to deliver a verdict from the observed output sequence.

In industrial practice, the specification of the behavior of logic controllers is not given in such formal models, however, but in tailor-made, (officially or de facto) standardized specification languages, like Grafcet or state-charts. Test cases are then built manually, what is a very tedious, time-consuming and error-prone task. To take benefit of previous works on conformance test based on formal models, it matters to endow the specification languages that are used in industry with a formal semantics and to develop translation methods of models in these languages into formal ones. Several results on model-based conformance test from UML state-charts have been already published (Massink, Latella, & Gnesi, 2006 for instance) but, as far as we know, the issue of conformance test when the specification is given in the form of a Grafcet, a powerful specification language for logic controllers, has never been addressed. The aim of this paper is to fill this gap.

More precisely, this paper proposes a method to translate a Grafcet specification model into an equivalent Mealy machine, without semantics loss (Fig. 2). Mealy machines have been chosen as the formal target model because conformance test of Mealy machines is a mature technique that previously yielded numerous sound results, as surveyed in Lee and Yannakakis (1996). However, this choice implies that only non-timed systems are considered; then, the Grafcet specification model shall not contain any time-dependent element. This limitation is not too strong because the first concern of engineers during conformance test is functional correctness; only the correctness of the non-timed behavior of the controller with regard to the specification is checked. Conformance test for checking time correctness is a second concern, once functional correctness is ensured.

Conformance test is a black-box test: the implementation is seen as a black-box with inputs/outputs. In the case of a logic controller, this means that its internal structure is unknown and its behavior can only be determined by observing its response to an

* Corresponding author. Fax: +33 147402220.
E-mail addresses: julien.provost@lurpa.ens-cachan.fr (J. Provost), jean-marc.roussel@lurpa.ens-cachan.fr (J.-M. Roussel), jean-marc.faure@lurpa.ens-cachan.fr (J.-M. Faure).

0967-0661/$ - see front matter
doi:10.1016/j.conengprac.2010.10.001

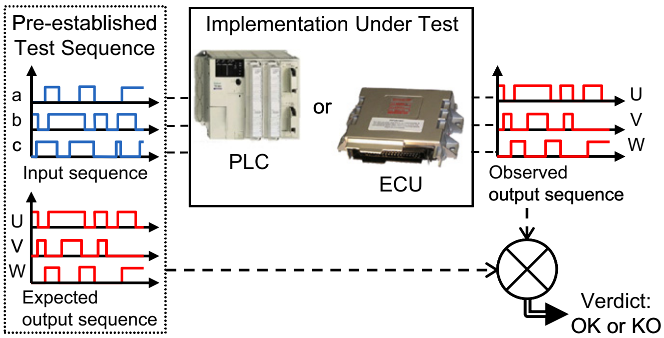

Fig. 1. Conformance test principle.

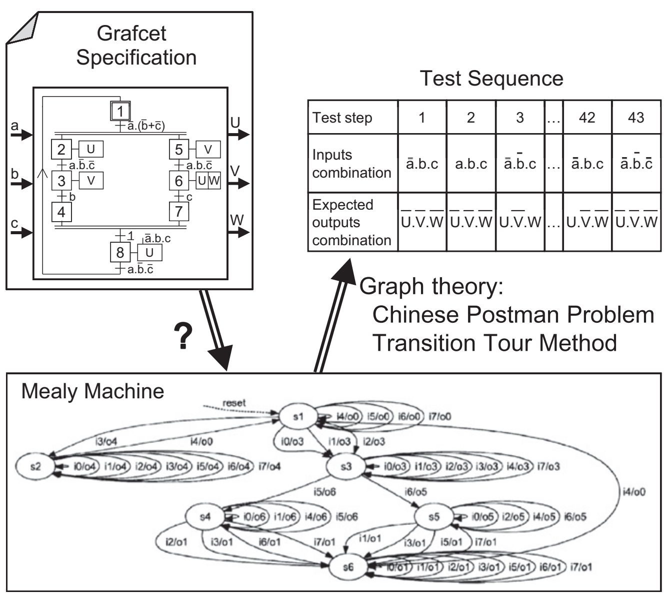

Fig. 2. Objective of the work.

input sequence. Moreover, to provide reliable results for controllers of highly critical systems, this test must be:

- *Non-invasive*: No probe or piece of code can be introduced within the controller. It is therefore impossible to obtain the values of its internal variables.
- *Exhaustive*: The whole state space of the specification model, a Grafcet model in this work, must be explored. In the rest of this paper, it will be supposed that the size of this state space is small enough to avoid combinatorial explosion. This assumption is quite reasonable for safety/security functions of critical systems. Indeed, since these functions must be very reactive (the response time to any change of their inputs must be very short), they do not perform complex treatments and the state space of the specification of such a function is tractable. For this reason, scalability of the test method will be not more addressed in what follows.

The model obtained by the translation method shall permit to satisfy these two test constraints.

The outline of the paper is the following. The background of this work—Grafcet syntax and standardized evolution rules as well as conformance test of Mealy machines—is reminded in the next section. An overview of the translation method is given in Section 3. The two phases of this method are then detailed and illustrated on a small but non-trivial example, respectively in Sections 4 and 5. Then, Section 6 focuses on test sequence generation from the final formal model, while perspectives for future works to extend this contribution are given in the conclusion.

2. Background

2.1. Grafcet specification language

Grafcet is a standardized graphical specification language (IEC 60848, 2002) to describe the behavior of logic sequential systems. This language is widely used in several industrial domains, like railway transport, electrical power production, manufacturing industry, environment, to specify the expected behavior of a logic control system which is connected to a physical system (plant) that sends logic signals to the control system and receives the logic signals which are generated in response. Grafcet was first standardized in France at the beginning of the 1980s, and at the international level in 1988. Since this date, several extensions have been proposed to enhance the modeling possibilities; they are included in the last version of the standard (IEC 60848, 2002). A good scientific presentation of the main features of the previous and current versions of the Grafcet standard can be found respectively in David (1996) and Guéguen and Bouteille (2001). Last, the reader is warned that the specification language described in the IEC 60848 standard differs from the SFC (sequential function chart) proposed by the IEC 61131-3 standard (IEC 61131-3, 2003), even if both are often named SFC in English and if models in these two languages may look similar; the differences stand both in syntax and semantics. The main differences between those two languages will be discussed in subsection 'Differences between Grafcet and SFC'. To avoid misunderstandings, only the term Grafcet will be kept in the sequel of this paper for the specification language.

Grafcet has been developed from the results of the Petri nets community and in particular from those on Interpreted Petri Nets. A specific syntax and semantics have been defined however, to take into account the specific needs of engineers when specifying complex sequential systems. The key features of Grafcet syntax and semantics are briefly recalled as follows.

2.1.1. Grafcet syntax

A Grafcet model describes the expected behavior of a logic controller which receives logic input signals and generates logic output signals; then, the input and output variables of a Grafcet are both logic variables. A Grafcet (Fig. 3) comprises steps, graphically represented by squares, and transitions, represented by horizontal lines; a step can only be linked to transitions and a transition only linked to steps. The links from steps to transitions and from transitions to steps are oriented links. The default orientation is from top to bottom and it is not necessary in this case to put an arrow on the link. An arrow must be put on a link if this link goes from bottom to top or may be put on any link to ease understanding. A step defines a partial state of the system and can be active or inactive; hence, a Boolean variable, named step activity variable can be defined for each step. Actions may be associated to a step; an action associated to a step is performed only when this step is active and then acts upon an output variable. A transition condition must be associated to each transition; this condition is a Boolean expression which may include input variables, steps activity variables and conditions on time. As only non-timed systems are considered in this work, only the Grafcets whose transition conditions are built from input variables and steps activity variables are dealt with.