



ELSEVIER

Contents lists available at ScienceDirect

Information Sciences

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

Soundness-preserving reduction rules for reset workflow nets

M.T. Wynn^{a,*}, H.M.W. Verbeek^b, W.M.P. van der Aalst^{a,b}, A.H.M. ter Hofstede^a, D. Edmond^a^a Business Process Management Group, Queensland University of Technology, GPO Box 2434, Brisbane QLD 4001, Australia^b Department of Mathematics and Computer Science, Eindhoven University of Technology, P.O. Box 513, NL-5600 MB Eindhoven, The Netherlands

ARTICLE INFO

Article history:

Received 13 July 2007

Received in revised form 10 September 2008

Accepted 26 October 2008

Keywords:

Reset nets

Reduction rules

Workflow verification

Soundness

ABSTRACT

The application of reduction rules to any Petri net may assist in its analysis as its reduced version may be significantly smaller while still retaining the original net's essential properties. Reset nets extend Petri nets with the concept of a reset arc, allowing one to remove all tokens from a certain place. Such nets have a natural application in business process modelling where possible cancellation of activities need to be modelled explicitly and in workflow management where such process models with cancellation behaviours should be enacted correctly. As cancelling the entire workflow or even cancelling certain activities in a workflow has serious implications during execution (for instance, a workflow can deadlock because of cancellation), such workflows should be thoroughly tested before deployment. However, verification of large workflows with cancellation behaviour is time consuming and can become intractable due to the state space explosion problem. One way of speeding up verification of workflows based on reset nets is to apply reduction rules. Even though reduction rules exist for Petri nets and some of its subclasses and extensions, there are no documented reduction rules for reset nets. This paper systematically presents such reduction rules. Because we want to apply the results to the workflow domain, this paper focusses on reset workflow nets (RWF-nets), i.e. a subclass tailored to the modelling of workflows. The approach has been implemented in the context of the workflow system YAWL.

Crown Copyright © 2008 Published by Elsevier Inc. All rights reserved.

1. Introduction

The analysis of a non-trivial concurrent process is a complicated task. There are a number of different approaches to deal with this complexity. Reducing a specification, while preserving its essential properties with respect to a particular analysis problem, is one such approach. There exists a body of research that addresses the concept of reduction in the area of Petri nets (see, e.g., [4,13]) and its various subclasses (see, e.g., free-choice nets [6]) and extensions (see, e.g., timed petri nets [14]).

Reset nets [5,8–11] extend Petri nets with the concept of a *reset arc*, a type of arc that connects a place and a transition and which semantics is to remove all tokens from that place when the transition fires. The complexity introduced by a reset arc (when compared with Petri nets in general) is threefold: (1) as the transition removes *all* tokens and not just one, place invariants do not hold for such nets, (2) the reset action can be *ineffective* if a place does not contain any tokens at the exact time when the transition fires and the reset action is carried out, and (3) a reset arc can affect any place in the entire net (i.e., its effect is global), unlike normal arcs of a transition which can only influence their input and output places (i.e., their effect is local). As a result, the notion of reachability is undecidable for reset nets with more than two reset arcs [9].

* Corresponding author.

E-mail addresses: m.wynn@qut.edu.au (M.T. Wynn), h.m.w.verbeek@tue.nl (H.M.W. Verbeek), w.m.p.v.d.aalst@tue.nl (W.M.P. van der Aalst), a.terhofstede@qut.edu.au (A.H.M. ter Hofstede), d.edmond@qut.edu.au (D. Edmond).

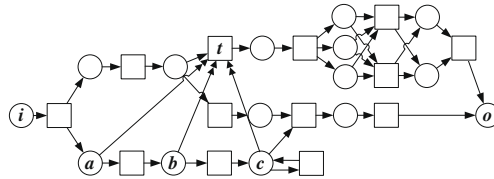


Fig. 1. An example RWF-net.

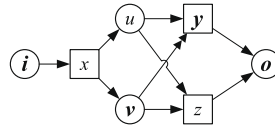


Fig. 2. Reduced RWF-net for the net in Fig. 1.

Reset nets form a natural foundation for workflow languages with explicit support for cancellation. Cancellation is an important concept in workflow management (see, e.g., [3]) where the execution of some activities may lead to the termination of other activities *in certain circumstances*. Cancellation can be triggered by either a customer request (e.g., a customer wishes to withdraw a credit card application) or by exceptions (e.g., an order cannot be processed due to insufficient stock level). In general, cancellation results in one of two outcomes: disabling some scheduled activities or stopping currently running activities. The complicating factor is that due to concurrency issues, the cancellation action may or may not result in cancelling certain activities, i.e., the process may be in a state before or after the part that is supposed to be cancelled. This can introduce deadlocks (the state where a business process is stuck and cannot proceed). As it is important to detect errors in workflows before their deployment, it is desirable to speed up the analysis of workflows if possible. When this analysis is performed on reset nets, corresponding to the workflows involved, this boils down to being able to perform efficient reset net analysis. While there are potentially a number of avenues that could be explored in order to achieve this, one approach is the application of reduction rules for reset nets.

When reducing a net it is imperative that certain essential properties are preserved. In the area of workflow verification, soundness is such a property. Soundness is a correctness notion of workflows that requires that a workflow can always terminate, that when termination is signalled no other tokens remain elsewhere in the net, and that it does not contain any dead tasks [15]. A reset workflow net (RWF-net) is a reset net with three structural restrictions: there is exactly one source node, one sink node and every node in the graph is on a directed path from the source node to the sink node. See Fig. 1 for an example of a sound RWF-net. In Fig. 1, transition t can remove tokens from places a , b , and c when it fires (denoted by double-headed arcs). By applying reduction rules, it is possible to reduce the net while preserving the soundness property of the net as shown in Fig. 2, where transition y corresponds to transition t and its successors in the original net and place v replaces all three places a , b , and c .

In this paper a number of soundness-preserving reduction rules for RWF-nets are documented and proven correct. They are inspired by reduction rules provided for Petri nets in [4,13] and for free-choice Petri nets provided in [6]. As mentioned before, there are no documented reduction rules for reset nets that preserve soundness in the current literature, and as it turns out, one has to be careful in determining the conditions under which certain types of reductions can be applied. For example, because of the nature of reset arcs removing *all* tokens when a transition fires, it is possible that an incorrect net that does not satisfy the proper completion criterion (i.e. tokens can be left in the net when it reaches the end) becomes sound when there is a reset arc to remove the leftover tokens before completion. Another interesting finding is that strict conditions are necessary for reset net reduction rules (for instance, two places can only be reduced to one if and only if both places are reset by the same transition). Hence, the presence of reset arcs does not offer new possibilities for reduction, rather it limits them. Also, a reset arc can never be abstracted entirely from a reset net. That is, if a net contains reset arcs, it is not possible to obtain a reduced net *without* any reset arc. Hence, the goal of reduction is to reduce the number of reset arcs when these arcs are connected to more than one place or transition and as a result, to reduce the complexity introduced by multiple reset arcs.

The contributions of the paper are as follows:

- The set of reset net reduction rules represents a practical contribution to achieve a more efficient verification of complex workflows with cancellation behaviours. The rules have been implemented as part of the verification functionality of the workflow language YAWL.¹

¹ www.yawl-system.com.

Download English Version:

<https://daneshyari.com/en/article/394843>

Download Persian Version:

<https://daneshyari.com/article/394843>

[Daneshyari.com](https://daneshyari.com)