A method to compare the descriptive power of different types of Petri nets

Kurt Jensen
Computer Science Department
Aarhus University
Ny Munkegade
DK-8000 Aarhus C
Denmark

1. INTRODUCTION

The purpose of this paper is to show how the descriptive power of different types of Petri nets can be compared, without the use of Petri net languages. Moreover, the paper proposes an extension of condition/event-nets and it is shown that this extension has the same descriptive power as condition/event-nets.

In many applications the basic Petri net formalism is augmented in different ways. Each of these extensions corresponds to a subclass of the very general "transition systems" defined in [Keller 76]. The descriptive power of such subclasses can be compared by use of formal language theory, where each transition is given a name and the set of possible firing sequences is considered, [Hack 76]. In the present paper it is, however, proposed to compare the descriptive power in a more direct way, which is closely connected to the idea of simulation.

In section 2 we define condition/event-nets and an extension of them called "testing Petri nets". Testing Petri nets were introduced in [Jensen 78], (there they were called "extended Petri nets"), and they have been used to define a formal semantics for a system description language in [Jensen, Kyng & Madsen 79]. Similar primitives are described in [Zuse 79]. The new primitives allow a transition to test some of its conditions without altering their markings.

In section 3 we give a transformation mapping each testing Petri net into a condition/event-net. Moreover we construct a function mapping markings for a testing Petri net into markings for the corresponding condition/event-net. We prove that the transformation satisfies three equations, all of them dealing with reachability.

In section 4 we define "transition systems" (from [Keller 76]) and "simulations" between them. The definition of simulation is directly inspired by the three equations proved for the transformation in section 3. We then prove a close connection between simulation and "strict reduction" (from [Kwong 77]) and this allows us to translate results, obtained for strict reduction by Kwong, to our situation where testing Petri nets are simulated by condition/event-nets. These results show that properties such as existence of a home state, Church-Rosser, non-haltingness and determinacy are preserved by simulation (i.e. the simulated system has the properties iff the simulating system has).
In section 5 we replace transition systems by "named transition systems" (from [Keller 76]) and simulation by "simulation induced by consistent homomorphisms". The latter definition is inspired by "strict reduction induced by homomorphisms" (from [Kwong 77]) and by "consistent" homomorphisms (from [Roucairol & Valk 79]). This allows us to translate a result obtained by Roucairol and Valk concerning liveness.

In section 6 we conclude that the method, used in sections 3, 4 and 5 to compare testing Petri nets and condition/event-nets, can be used to compare other types of Petri nets. We define "equivalence with respect to descriptive power". We discuss how to find the transformations necessary to compare two types of Petri nets, and we give references to papers where such transformations have been sketched.

2. CONDITION/EVENT-NETS AND TESTING PETRI NETS

Definition

A condition/event-net is a 5-tuple CEN = (P, T, PRE, POST, m₀) where

1) P is a set of places
2) T is a set of transitions
3) P ∩ T = ∅, P ∪ T ≠ ∅
4) PRE, POST ∈ [T → IP(P)] where [... and IP denote total functions
and powersets respectively
5) ∀ i ∈ T [PRE(i) ∩ POST(i) = ∅]
6) m₀ ∈ [P → {0, 1}] is the initial marking.

A marking is a function m ∈ [P → {0, 1}]. A place p is marked iff m(p) = 1 and unmarked iff m(p) = 0. A place p is a condition for a transition t iff p ∈ COND(t) = PRE(t) ∪ POST(t). It is a precondition iff p ∈ PRE(t) and a postcondition iff p ∈ POST(t).

Functions defined on P or T will in this and the following sections be extended to IP(P) or IP(T) in the usual way. As an example PRE(X) = \bigcup_{t ∈ X} PRE(t), for all X ⊆ T.

Condition/event-nets can be represented as directed graphs with two kinds of nodes. Circles represent places, while squares represent transitions. Each transition has ingoing arcs from its preconditions and outgoing arcs to its postconditions. The initial marking is represented by tokens (solid dots) on the marked places.

Two transitions are independent iff their conditions are disjoint. A nonempty set of mutually independent transitions X has concession (and may fire) in a marking m iff all places in PRE(X) are marked and all places in POST(X) are unmarked. If X fires, a new marking m' is reached where all places in PRE(X) are unmarked and all places in POST(X) are marked. We then say that m' is directly reachable from m, which we write as m → m' or m \xrightarrow{X_1 \cdot X_2 \cdot \ldots \cdot X_n} m', where \{X_i | i ∈ 1..n\} with