Abstract. We provide a Petri net semantics for a subset of CCSP, the union of Milner's CCS and Hoare's CSP. It assigns to each process term in the subset a labelled, one-safe place/transition net. As opposed to many other approaches to Petri net semantics, our definition is operational as it is based on Plotkin-style transition rules. These rules are inspired by work of Degano, DeNicola and Montanari, but differ in the way they model the interplay of the central concepts in CCSP: concurrency, nondeterminism and recursion. To discuss these differences, we propose criteria for a good Petri net semantics for CCSP.

1. Introduction

Attractive methods of describing concurrent processes are Petri nets [Re2], Milner's CCS (Calculus of Communicating Systems) [Mi] and Hoare's CSP (Communicating Sequential Processes) [Ho]. Whereas Petri nets concentrate on a proper representation of concurrency, both conceptually and graphically, CCS and CSP provide insights into the aspects of structure and abstraction (cf. [012]). In fact, CCS and CSP have converged to one theory of processes which - following a suggestion of M. Nielsen - we shall call here "CCSP". It offers an intriguing variety of structural operators on processes and of equivalences which abstract from internal process behaviour and thus facilitate process verification. On the other hand, true concurrency is neglected in favour of a simple interleaving model of parallelism.

Clearly, any satisfactory description of processes must integrate all three aspects: concurrency, structure and abstraction. A step
towards such an integration is giving a Petri net semantics for CCSP, with the benefit of clearly distinguishing between concurrency and nondeterminism. In fact, a large body of research has been devoted to this aim (cf. e.g. [By, Cz, DMFS, GV, Gz, GL, GM, Lo, Po, Re1, Ta, Wil, 2] and [Be, LC] for related work on COSY, a prerunner of CCS and CSP). Surprisingly, all these papers can be seen as providing a denotational Petri net semantics \( \mathcal{N} \) for (subsets of) CCSP.

Denotational means that for each \( n \)-ary CCSP operator \( \text{op} \) a corresponding Petri net operator \( \text{op}^{\mathcal{N}} \) is defined, satisfying the equation

\[
\mathcal{N}[\text{op}(P_1, \ldots, P_n)] = \text{op}^{\mathcal{N}} (\mathcal{N}[P_1], \ldots, \mathcal{N}[P_n]).
\]

Some of these Petri net operators \( \text{op}^{\mathcal{N}} \) have a lengthy definition. Maybe simpler definitions can be given, but certainly these complications make it worthwhile to try an alternative approach to Petri net semantics which has been highly successful in providing a lucid interleaving semantics for CCSP: structured operational semantics as advocated by Plotkin [PlI, 2] and used by many others [Ap, BHR, HP, Mi, OH].

Operational means that only the states and transitions of an abstract machine are described. The interleaving semantics of CCSP takes as machine a nondeterministic automaton in the sense of classical automata theory [RS], also known as labelled transition system [Ke]. To obtain a Petri net semantics, the automaton has to be replaced by a Petri net. The question is how to do this in the structured style of Plotkin's operational semantics where the distributed states and transitions of a Petri net should be represented by syntactic expressions. An answer to this question has recently been given by Degano, DeNicola and Montnari who are the first to provide such an operational Petri net semantics for Milner's CCS [DDM].

While their semantics models well the concurrency present in the parallel composition of CCS processes, it - in our view - fails to master the subtleties of the interplay of concurrency with nondeterminism and with recursion. Here the semantics of [DDM] exhibits less concurrency than seems natural.

This is why in our paper we develop a new operational Petri net semantics for (a subset of) CCSP. It follows closely the work of [DDM], but differs in the way nondeterminism and recursion is modelled. To evade any subjective judgements of one semantics being "better" than another one, we propose four criteria that a good Petri net semantics for (that subset of) CCSP should satisfy. While these criteria themselves can be debated, it can now be clearly stated whether or not a semantics satisfies them. For example, the semantics of [DDM] satisfies only two of them while our semantics satisfies all four.