We define two normal forms for CSP programs. In the first normal form, each process contains only one I/O repetitive command and all its I/O commands appear as guards of this command. In the second normal form, all guards of this I/O repetitive command are I/O guards. We describe an inductive method that transforms any CSP program into an equivalent program in first or second normal form. The notion of equivalence is discussed. It is shown that no transformation into second normal form can preserve deadlock freedom.

MSC:
68N25 Theory of operating systems

Keywords:
partially interpreted computation; syntactic equivalence; deadlock freedom

Software:
CESAR

Full Text: DOI Link

References:
[2] Apt, K.R.; Clermont, Ph., Two normal form theorems for CSP programs, () · Zbl 0631.68026
[6] Frances, N.; Rodeh, M.; Sintzoff, M., Distributed termination with interval assertion, (), Peniscola, Spain · Zbl 0474.68033

This reference list is based on information provided by the publisher or from digital mathematics libraries. Its items are heuristically matched to zbMATH identifiers and may contain data conversion errors. It attempts to reflect the references listed in the original paper as accurately as possible without claiming the completeness or perfect precision of the matching.