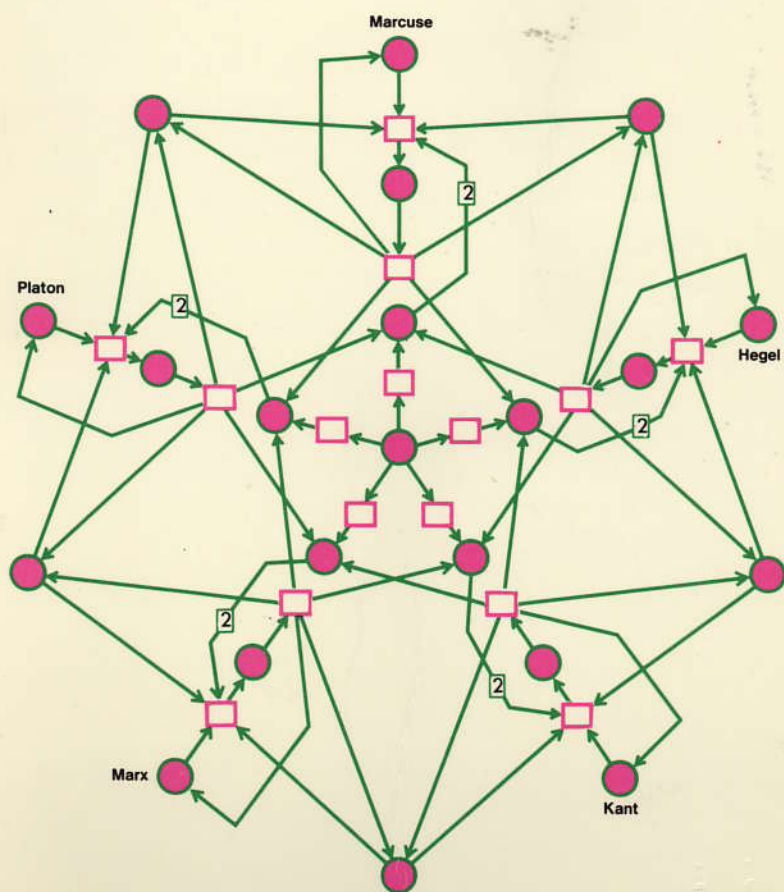


parallélisme, communication et synchronisation

édité par
J.P. VERJUS-G. ROUCAIROL



Éditions du CNRS



Parallélisme, Communication et Synchronisation

SOMMAIRE

Édité par

J.P. VERJUS

Professeur à l'Université de Rennes

*Directeur de l'IRISA (Laboratoire Associé CNRS n° 227
et Centre INRIA de Rennes)*

G. ROUCAIROL

Directeur de la Recherche en Architecture et Logiciel

BULL (Louveciennes)

Partie 1 :	
Méthodes	
1. Spécification, Analyse et Comparaison des Systèmes Synchronisés	3
A. Arnold	
2. Le Calcul MEHE	23
G. Baudet	
3. Équivalence	47
Directeur de la Recherche en Architecture et Logiciel	
BULL (Louveciennes)	
Partie 2 :	
Méthodes et Outils de Preuve de la Synchronisation	
4. PROLOG Interprète de Règles de Petri	67
Applications aux Protocoles de Communication.	
P. Azam	
5. Conception Certifiée de Systèmes Distribués : un exemple	91
P. Caspi, N. Halbwachs	
6. Principe des Méthodes de Preuve de Propriétés d'Invariance et de Fatalité des Programmes Parallèles	129
P. Lacombe, R. Cousot	
7. Vérification des Propriétés des Systèmes de Transition	151
ÉDITIONS DU CENTRE NATIONAL DE LA RECHERCHE SCIENTIFIQUE	
15, quai Anatole France - 75700 PARIS	
1985	

Principe des Méthodes de Preuve de Propriétés d'Invariance et de Fatalité des Programmes Parallèles

P. Cousot, R. Cousot

1. - INTRODUCTION

Les programmes, en particulier parallèles, doivent faire l'objet d'analyses détaillées avant d'être considérés comme valides. Nous nous intéressons à des analyses qualitatives qui consistent à comparer les comportements possibles d'un programme parallèle à une spécification des comportements souhaitables. Dans ce rapport, nous ne traitons que des méthodes de preuve. Dans un but simplificateur, nous supposons résolus les problèmes de description d'un programme parallèle, de ses comportements possibles et de sa spécification. Nous utiliserons donc des modèles simples mais très généraux des programmes (représentés par une relation de transition sur des états), de leurs comportements (définis par des ensembles de traces complètes) et de leur spécification (à l'aide de propriétés d'invariance ou de fatalité). Reste notre problème d'intérêt à savoir l'étude des méthodes de preuve permettant de démontrer qu'un programme parallèle se comporte conformément à une spécification. Dans ce rapport, nous nous limiterons au cas des méthodes de Floyd (dite des assertions inva-

riantes) et de Burstall (dites des assertions intermittentes) et à quelques-unes de leurs généralisations.

Nous essayons tout d'abord d'exprimer l'essence d'une méthode de preuve à l'aide de principes d'induction. Quand c'est fait, nous en démontrons la correction et la complétude sémantique. Nous cherchons ensuite à établir des comparaisons, en particulier à démontrer l'équivalence forte c'est-à-dire qu'une preuve par une méthode peut se réécrire en une preuve par une autre méthode. Nous nous efforçons également de généraliser ces principes d'induction, en particulier au cas de comportements possibles correspondant à des ensembles de traces non clos. C'est le cas des hypothèses d'exécution faiblement équitables pour les programmes parallèles que nous prendrons en exemple dans ce rapport. Finalement, nous étudions comment adapter une méthode de preuve à un langage particulier en partant d'une sémantique opérationnelle de ce langage et d'un principe d'induction formalisant la méthode de preuve.

2. - SYSTEMES DE TRANSITION : MODELES DES PROGRAMMES

PARALLELES

Nous considérerons, de manière simple mais générale, qu'un programme parallèle est un système de transition (S, t, ε) où S est un ensemble non vide d'états, $t \in (S \times S \rightarrow \{tt, ff\})$ une relation (non-déterministe) de transition entre un état et ses successeurs possibles et $\varepsilon \in (S \rightarrow \{tt, ff\})$ caractérise les états initiaux. Par commodité t et ε sont considérés comme des fonctions à valeurs de vérité (tt vrai, ff faux).

Il sera quelquefois plus aisé de considérer que la relation de transition t est partitionnée : $t = \bigvee_k t_k$ où chaque t_k correspond à un processus du programme parallèle. De même, l'ensemble S des états pourra être défini par un recouvrement $S = \bigcup_{\ell} S_{\ell}$.

