

Résumé

Ce travail entre dans le cadre de la vérification formelle de processus parallèles pris dans un environnement d'interleaving (entrelacement), et présentant un nombre *infini* d'états.

Le formalisme que nous avons utilisé pour décrire cette classe de processus étant *les grammaires à contexte libre*. Selon l'interprétation que l'on donne à la concaténation des symboles non terminaux de la grammaire, nous parlons de *processus à contexte libre* et de *processus à contexte libre concurrents*, la première classe considère une composition *séquentielle* des non terminaux de la grammaire et coïncide avec les processus de l'algèbre *BPA* (BPA processes), la seconde considère une composition *parallèle* de ces mêmes symboles et se confond avec les processus de l'algèbre *BPP* (BPP processes). Nous définissons aussi une troisième classe de processus admettant les deux formes de composition: il s'agit des *processus concurrents* qui ne sont autres que les processus de l'algèbre *PA* (PA processes).

Nous nous intéressons à la vérification comportementale, et plus précisément aux relations basées sur la sémantique de *bisimulation*, essentiellement la *bisimulation forte*.

Nous reprenons d'abord les deux algorithmes proposés pour la vérification de la bisimulation (forte) s'appliquant aux processus à contexte libre *normés* et aux processus à contexte libre concurrents *normés*.

Notre contribution consiste d'une part, à faire abstraction de ce caractère normé des processus, ainsi nous procédons à une généralisation de ces algorithmes pour des processus à contexte libre *non normés* et des processus à contexte libre concurrents *non normés*.

D'autre part, nous proposons un algorithme de décision de la bisimulation (forte) pour la classe des processus concurrents (PA processes) normés.

Nous orientons notre étude vers des relations moins fines que la bisimulation (forte), ainsi pour des processus à contexte libre normés, nous proposons un algorithme de décision de la *congruence observationnelle*. Dans certains cas, cet algorithme peut être utilisé pour décider de la *τ -bisimulation*, la *branching-bisimulation*, la *η -bisimulation* et la *delay-bisimulation*.