

Projet

Algorithmique des systèmes parallèles et distribués

Dominique Méry
Telecom Nancy, Université de Lorraine

22 février 2025

Table des matières

1	Objectifs du projet	2
1.1	Algorithmes work1, work2, work3 et work4	2
1.2	Dossier à analyser	3
2	Dossier à rendre	3

1 Objectifs du projet

Le projet vise à modéliser, à analyser et à programmer des algorithmes parallèles ou répartis en utilisant le langage TLA/TLA⁺ [8, 9] et le langage PlusCal. Vous pouvez traduire directement un algorithme en TLA/TLA⁺ ou utiliser PlusCal.

Deux types de travaux sont demandés :

- Modélisation et analyse avec la toolbox et PlusCal des algorithmes (work1 ou work3) et (work2 ou work4).
- Lire un article et rédiger un résumé de moins de quatre pages qui explique ce que fait l'algorithme décrit; il faudra donner les propriétés requises (correction partielle, correction totale, propriétés de sûreté, de vivacité etc). Vous essaieriez de donner une description algorithmique de haut niveau afin de décrire l'algorithme aussi simplement que possible.

1.1 Algorithmes work1, work2, work3 et work4

work1 Un ensemble de processus P_1, \dots, P_n partage une ressource R qui n'est utilisable que par au plus deux processus à la fois. Pour utiliser cette ressource, un serveur Q gère la ressource et son utilisation selon des règles données. Les processus P_1, \dots, P_n communiquent par envoi de messages à Q et Q peut communiquer des messages à chaque processus P_i . Les communications sont fiables. On suppose que tout message envoyé est finalement reçu. On ne donne pas a priori de règles de communication trop strictes comme FIFO ou autre chose et cela est laissé à votre choix. Il vous appartiendra de justifier chaque choix fait quand cela est possible. Proposer une technique qui organise la gestion équitable et fiable de cette ressource par Q . Vous pouvez utiliser des primitives comme `send(mes,Q)` pour envoyer le message `mes` à Q et `receive(mes,cond(mes),Q)` pour attendre de recevoir un message `mes` satisfaisant `cond(mes)`. Pour écrire les algorithmes, vous utiliserez les constructions classiques *if, while, ...* Proposer un module TLA⁺ ou PlusCal qui développe votre solution et qui la vérifie.

work2 Vous devez modéliser un des algorithmes de ramasse-miettes ou garbage collection de Ben-Ari [2] qui sont écrits dans un langage algorithmique de type ADA. Ces algorithmes sont des algorithmes concurrents partageant une même mémoire.

work3 Vous devez modéliser l'algorithme de Ricart et Agrawala en PlusCal. Ricart et Agrawal [13] décrivent leur solution et la solution proposée est déjà sous une forme algorithmique

work4 Vous devez modéliser l'algorithme de Carvalho et Roucairol en PlusCal. Carvalho et Roucairol[3] décrivent leur solution et la solution proposée est déjà sous une forme algorithmique

Dans les quatre cas, il est demandé de rédiger un document qui explique les choix et aussi les difficultés rencontrées. Il faudra aussi montrer avec l'outil en

quoi les solutions retenues sont correctes et répondent au cahier des charges. Il est donc important de lister les charges et ensuite de montrer avec l'outil ce qui est vérifié.

1.2 Dossier à analyser

Nous proposons d'analyser l'un des dossiers suivants :

analysis1 *Distributed Algorithms for Sensor Networks* [10]

analysis2 *A Square Root N Algorithm for Mutual Exclusion in Decentralized Systems* [11]

analysis3 *An Efficient Message Passing Election Algorithm based on Mazurkiewicz's Algorithm* [4]

analysis4 *A Distributed Algorithm for Minimum-Weight Spanning Trees* [6]

analysis5 *A Quorum-Based Group Mutual Exclusion Algorithm for a Distributed System with Dynamic Group Set* [1]

analysis6 *Self-Stabilizing Distributed Algorithm for Strong Matching in a System Graph* [7].

analyse7 *Fully distributed algorithms for convex optimization problems* [12]

analyse 8 *Reliable broadcast protocols* [?] [5]

Le travail demandé est de rédiger une synthèse qui explique les éléments importants de l'algorithme présenté. Il faudra aussi indiquer les problèmes qui pourraient se poser pour sa mise en œuvre. Si vous êtes assez efficaces, proposez un algorithme PlusCal . . .

2 Dossier à rendre

Le rapport à rendre devra expliquer clairement ce que vous avez fait. Ce rapport sera synthétique (10 pages) et mettra en avant les difficultés rencontrées et donnera des possibilités d'extension de cet outil. Le dossier final sera une archive contenant les différents éléments concernant la vérification avec chaque outil, les sources, tous les éléments expliquant ce que vous avez fait et le rapport final en pdf. Vous enverrez une archive avec le rapport et les fichiers TLA. La date de remise des dossiers est le 15 mai 2025 et chaque groupe présentera son travail le 20 mai 2025 entre 14h00 et 20h00 à Telecom Nancy salle Telecom Nancy. Un dossier est une archive dont le nom est de la forme nom1-nom2-nom3- projet.zip (utilisez zip pour compresser).

Références

- [1] Ranganath Atreya, Neeraj Mittal, and Sathya Peri. A quorum-based group mutual exclusion algorithm for a distributed system with dynamic group set. *IEEE Trans. Parallel Distrib. Syst.*, 18(10) :1345–1360, 2007.

- [2] Mordechai Ben-Ari. Algorithms for on-the-fly garbage collection. *ACM Trans. Program. Lang. Syst.*, 6(3) :333–344, July 1984.
- [3] Osvaldo Carvalho and Gérard Roucairol. On the distribution of an assertion. In Robert L. Probert, Michael J. Fischer, and Nicola Santoro, editors, *ACM SIGACT-SIGOPS Symposium on Principles of Distributed Computing, Ottawa, Canada August 18-20, 1982*, pages 121–131. ACM, 1982.
- [4] Jérémie Chalopin and Yves Métivier. An efficient message passing election algorithm based on mazurkiewicz’s algorithm. *Fundam. Inform.*, 80(1-3) :221–246, 2007.
- [5] K. M. Chandy and J. Misra. The drinking philosophers problem. *ACM Trans. Program. Lang. Syst.*, 6(4) :632–646, October 1984.
- [6] Robert G. Gallager, Pierre A. Humblet, and Philip M. Spira. A distributed algorithm for minimum-weight spanning trees. *ACM Trans. Program. Lang. Syst.*, 5(1) :66–77, 1983.
- [7] Wayne Goddard, Stephen Hedetniemi, David Jacobs, and Pradip Srimani. Self-stabilizing distributed algorithm for strong matching in a system graph. pages 66–73, 06 2003.
- [8] Leslie Lamport. The temporal logic of actions. *ACM Trans. Program. Lang. Syst.*, 16(3) :872–923, 1994.
- [9] Leslie Lamport. *Specifying Systems, The TLA+ Language and Tools for Hardware and Software Engineers*. Addison-Wesley, 2002.
- [10] Christoph Lenzen and Roger Wattenhofer. Distributed algorithms for sensor networks. School of Engineering and Computer Science, Hebrew University of Jerusalem Edmond Safra Campus, Givat Ram, 91904 Jerusalem, Israel and Computer Engineering and Networks Laboratory, ET.
- [11] Mamoru Maekawa. A square root N algorithm for mutual exclusion in decentralized systems. *ACM Trans. Comput. Syst.*, 3(2) :145–159, 1985.
- [12] Damon Mosk-Aoyama, Tim Roughgarden, and Devavrat Shah. Fully distributed algorithms for convex optimization problems. *SIAM Journal on Optimization*, 20(6) :3260–3279, 2010.
- [13] Glenn Ricart and Ashok K. Agrawala. An optimal algorithm for mutual exclusion in computer networks. *Commun. ACM*, 24(1) :9–17, January 1981.