



L'air de rien

N° 7

L'aléastriel du Laboratoire de Recherche et de Développement de l'EPITA¹

Numéro 7, Novembre 2006

Édito

par Akim Demaille (Enseignant-Chercheur)

Peu de feuilles sur les branches, bientôt la désinté, déjà presque plus d'étudiants en classe, c'est décidément la fin de l'année. Dans cette dernière ligne

droite, les CSI 2007 vous présentent la conclusion de leur passage au LRDE, ou un sujet qui leur tient à cœur. Les doctorants du labo feront le reste de l'actualité.

Un nouveau doctorant : Alexandre Hamez



Alexandre Hamez commence une thèse intitulée *Model checking réparti* sous la direction de Fabrice Kordon (LIP6/MoVe) et des co-encadrants Akim Demaille (EPITA/LRDE) et Yann Thierry Mieg (LIP6/MoVe), après avoir obtenu un master

Systèmes et Applications Distribués à l'université Pierre et Marie Curie (Paris VI).

Le model checking est une technique de vérification formelle permettant de prouver que le comportement d'un système est correct. Cependant, elle est très souvent sujette à une utilisation prohibitive du temps de calcul, ceci à cause du problème dit de l'ex-

plosion de l'espace d'états.

La répartition du model checking est apparue récemment comme étant une solution possible à ce problème en utilisant les avantages d'un réseau de machines, augmentant de ce fait la taille mémoire et la puissance de calcul disponible. Le but de ce travail est de répartir le processus de model checking, en se focalisant sur le cas particulier des clusters, lesquels offrent un environnement dédié au calcul réparti. Nous avons aussi comme objectif de fournir un framework facilitant la répartition de model checkers existants. De plus, nous souhaitons y intégrer des techniques de réduction d'espace d'états, tels que les diagrammes de décision.

Comment joindre l'utile à l'agréable



par Sébastien Hémon
Thésard LRI-Orsay - Équipe algorithmique et complexité

S'il n'était venu à l'esprit de Von Neumann de sortir la théorie des jeux de leur cadre d'étude, peut-être n'aurions-nous aucune réponse à cette question. L'explosion de la théorie des

jeux provient de son incroyable faculté à se faufiler partout : la bourse, les réseaux, les ventes aux enchères, les jeux (qui l'eut cru ?) et même l'évolution du langage et le règne animal ! Comment ?

Bien que tirée des jeux de cartes et de dés, la modélisation mathématique permet d'employer ce qui décrit un jeu à d'autres choses. La base est simple : on liste des stratégies possibles, on les relie entre elles en pondérant avec des gains et on octroie telle ou

¹L'air de rien, <http://publis.lrde.epita.fr/LrdeBulletin>.

telle en fonction des choix des autres (ou même des siens), voire même de la puissance de calcul à disposition et c'est parti. Il est dans ce petit monde un étrange phénomène appelé équilibre de Nash. Une position qui, à la manière d'un trou noir, fait que si l'on s'y engouffre, il n'y a plus moyen d'en sortir (en fait si, mais alors on est sûr de perdre). John Nash a montré dans les années cinquante que tout jeu possède au moins un tel point d'équilibre. Alors maintenant quoi ? Et bien, il faut les calculer ces positions. Posséder cette donnée vous permet de vous préparer d'emblée au jeu et même de savoir si ce dernier vous est, à la longue, favorable ou non. Et le problème est là : le calcul est complexe. Très complexe. Alors tout est bon : analyse, géométrie, algèbre, programmation dynamique et simplexe.

Bien sûr, la théorie des jeux s'intéresse à d'autres aspects : les stratégies gagnantes. Là, il ne s'agit plus de connaître vos gains en fonction de vos actes mais de savoir comment procéder pour gagner tout simplement. Il y a donc l'aspect quantitatif (« combien ça rapporte ? ») et qualitatif (« ma stratégie, c'est la meilleure »). Et là, surprenant : dans tout jeu à deux joueurs sans possibilité de nul, il existe une stratégie gagnante. Un seul des deux joueurs la possède alors. Aux dames, c'est fini : on a même fabriqué la partie de *Pensée profonde* qui joue aux dames et qui donne actuellement des leçons aux grands maîtres centenaires des dames. Dans les jeux résolus, vous trouverez le très célèbre puissance 4, Babylone (un jeu avec des petites tablettes à empiler) ou le célèbre jeu des allumettes.

Bien qu'il apparaisse que les jeux soient des outils que l'on implémente informatiquement, tels les très chers cercles vicieux des logiciens, on peut utiliser la théorie des jeux pour modéliser des problèmes informatiques (que l'on va donc résoudre par l'informatique). Ainsi, l'immensité du Web peut être vue comme un jeu dans lequel chaque joueur veut tirer profit de l'Internet. Rappelez-vous comment on se retrouvait facilement à télécharger à une vitesse de 3 Ko/s avec une connexion ADSL sur les réseaux de *peer-to-peer* lors de leurs fastidieux débuts. Citons des jeux algorithmiques très connus, à l'usage des informaticiens : le jeu de la vie et la fourmi de Langton (directement reliés au concept d'automate cellulaire).

C'est aussi grâce à la théorie des jeux que l'on voit apparaître une importante distinction entre deux entités souvent confondues : le hasard et le chaos. En informatique, on utilise du chaos pour simuler le ha-

sard. Le jeu de la vie et la fourmi de Langton sont typiquement des systèmes chaotiques, c'est à dire initialement déterministes mais dont l'évolution est suffisamment complexe pour que, si l'on ignore ce qu'il s'est passé à différentes étapes, on n'est plus en mesure d'expliquer ce que l'on observe. Ceci est lié à la classe des jeux dits à *information incomplète*. En revanche, le hasard est le fruit d'événements aléatoires, soit de phénomènes dont on ne connaît aucune explication autre que celle apportée par l'observation empirique d'une loi de probabilités (loto, dés et mécanique quantique).

Parmi les jeux chaotiques, il est aussi un petit jeu étrange et simple aux multiples appellations dont la véritable est « jeu de Gale ». La découverte de la stratégie gagnante générale pour ce jeu rapporte un prix coquet de quelques milliers de dollars. Ses règles sont simples : sur un quadrillage $n \times k$ muni d'une origine dans un coin, chaque joueur choisit un carreau à tour de rôle. De ce choix résulte la disparition de tous les carreaux situés dans la partie opposée au coin. Ainsi, si votre coin est bas-gauche, tout carreau plus haut et plus à droite disparaît. Le but ? Ne pas prendre le dernier carreau. Si ce jeu a des allures de jeu d'allumettes, son mécanisme en est très loin. Il appartient à une catégorie de jeu identifiée sous le nom de Poset Game (pour Partially Ordered Set) là où le jeu des allumettes est un jeu à ordre linéaire. Une partie de mon travail est consacrée à l'étude de ce jeu, dont la découverte d'une stratégie gagnante rapporte un prix de plusieurs milliers de dollars.

Parallèlement, mes travaux ont consisté à établir une nouvelle approche géométrique de l'équilibre de Nash ainsi qu'un nouveau concept d'approximation : ne reste plus qu'à déterminer leurs caractéristiques en terme de complexité (i.e. fournir un algorithme associé, efficace) pour savoir si demain, quelqu'un pourra faire main basse sur la bourse juste avec son portable connecté en Wi-Fi.

Pour aller plus loin

- jeu de Gale²
- fourmi de Langton³

Pour s'amuser un peu

- jeu de la vie⁴
- et autres automates cellulaires⁵
- fourmi de Langton⁶
- jeu de Gale (ou *Chomp Game*), à deux joueurs⁷ ou contre un ordinateur⁸

²jeu de Gale, <http://www.win.tue.nl/~aeb/games/chomp.html>.

³fourmi de Langton, <http://www.sdv.fr/pages/casa/html/langton.html>.

⁴jeu de la vie, <http://perso.orange.fr/jean-paul.davalan/divers/jeuvie/index.html>.

⁵autres automates cellulaires, http://perso.orange.fr/therese.eveilleau/pages/truc_mat/textes/conway.htm.

⁶fourmi de Langton, http://mypage.bluewin.ch/Ysewijn/french_ant.htm.

⁷jeu de Gale, deux joueurs, <http://www.duke.edu/~1jw6/ooga/chompgame.html>.

⁸jeu de Gale, contre un ordinateur, <http://perso.orange.fr/jean-paul.davalan/jeux/nim/chomp/index.html>.

Séminaire CSI de décembre 2006

Les étudiants du LRDE présenteront leurs travaux lors des séminaires ayant lieu à l'EPITA en décembre 2006. Au programme, des exposés sur la Parallélisation de calculs, le projet Vaucanson, les Diagrammes de décision binaires, la Théorie des jeux et le projet Olena. Venez nombreux !

Parallélisme appliqué au traitement d'images

par *Alexandre Borghi*

Le parallélisme permet d'effectuer plusieurs opérations simultanément tout en tirant parti d'une mémoire partagée. Aujourd'hui, l'informatique grand public permet d'utiliser cette capacité autrefois réservée aux professionnels du fait de leurs grands besoins en puissance de calcul.

Ce travail vise à explorer certaines techniques permettant la conception de bibliothèques parallélisées. L'objectif est de faciliter l'écriture d'algorithmes parallèles tout en cachant un maximum de détails implémentatoires superflus sans perdre en fonctionnalité.

Le traitement d'images a été choisi comme cadre de travail et plusieurs algorithmes ont été sélectionnés pour illustrer les concepts rencontrés.

Présentation du TAF-Kit

par *Robert Bigaignon*

La plate-forme de manipulation d'automates finis [Vaucanson](http://vaucanson.lrde.epita.fr)⁹ propose depuis peu un ensemble d'outils permettant la manipulation d'automates au travers de simples programmes Unix, à la façon de [FSM](http://www.research.att.com/~fsmttools/fsm/)¹⁰. Ces programmes sont réunis dans le Typed Automata Functions Kit (TAF-Kit) et s'appuient directement sur la bibliothèque C++, cœur du projet Vaucanson. Ils fournissent ainsi à l'utilisateur un sous-ensemble des services déjà présents dans Vaucanson (algorithmes et entrées-sorties) tout en lui épargnant l'écriture de programmes C++.

Nous présenterons les fonctionnalités du TAF-Kit ainsi que les outils dont il dispose pour mesurer ses performances.

État des lieux des propositions XML pour les automates

par *Florent Terrones*

Les utilisateurs de logiciels de manipulation d'automates ont exprimé depuis quelques années le

besoin de disposer d'un format de fichier universel de description d'automates.

L'équipe Vaucanson a présenté dès 2004 pour la conférence CIAA'04 une proposition de format XML pour la description d'automates à états finis. Cette dernière a subi de nombreuses modifications et suivi différentes lignes directrices pour en arriver à la version actuelle.

Le but de cet exposé est de faire un état des lieux de cette proposition et des avantages qu'elle apporte. Une description des autres propositions ainsi que leurs différentes caractéristiques seront également présentées.

BDD distribués, Java, Threads, et optimisations

par *Guillaume Guirado*

Lors d'une précédente présentation, les BDD (diagrammes de décision binaires) distribués ont été étudiés. Leur implantation, en Java, a montré ses limites lors des tests de performances.

Une des idées proposées afin d'accélérer les calculs était de mieux gérer les processus, ou threads. Premièrement, limiter le nombre de processus simultanés permet d'éviter de nombreuses restaurations des registres du processeur. Second point, une file d'attente des processus avec un contrôle intelligent des processus prioritaires ou non pourrait augmenter la parallélisation de l'application. Nous verrons ce qu'il en est dans les faits.

Enfin, le monde Java dispose de nombreux compilateurs ou optimiseurs susceptibles d'améliorer les performances. Nous testerons un certain nombre de ces outils et verrons, si oui ou non, les résultats en sont améliorés.

Les systèmes de recommandations

par *Thomas Largillier*

La multiplication des sites de ventes sur le Net a poussé à une plus grande fidélisation des clients. Ceci passe par une offre plus ciblée correspondant mieux à chaque utilisateur. On assiste donc à une émergence des systèmes de recommandations. Ces systèmes se basent sur les achats déjà effectués et la ressemblance de comportements entre les utilisateurs pour proposer des produits toujours plus attractifs. Nous ferons ici une présentation de plusieurs d'entre eux ainsi qu'une comparaison.

⁹Vaucanson, <http://vaucanson.lrde.epita.fr>.

¹⁰FSM, <http://www.research.att.com/~fsmttools/fsm/>.

Extraction automatique d'artères à partir d'images médicales 3D

par *Nicolas Widynski*

Depuis le début du 20^{ème} siècle, les techniques d'acquisition employées en imagerie médicale sont en constante évolution. Rayons X, IRM, ultra-sons et imagerie nucléaire sont des outils physiques permettant l'acquisition de différentes données telles que le scanner, l'échographie, la radiographie... Face à l'amélioration de ces techniques d'acquisition et au nombre croissant des données à traiter, le rôle du traicteur d'images médicales est d'extraire le maximum de caractéristiques et d'agencer au mieux les données, dans le but d'aider le spécialiste médical à effectuer son diagnostic.

Plus spécialement, le scanner peut nous permettre d'acquérir des données 3D. Notre rôle consiste ici à en extraire les artères coronaires. Pour ce faire, nous emploierons la méthode décrite par Thierry Géraud et Jean-Baptiste Mouret dans *Fast Road Network Extraction in Satellite Images Using Mathematical Morphology and Markov Random Field* (EURASIP 2004), en divisant notre chaîne de traitement en deux étapes : l'extraction d'une surface à partir de données 3D, puis l'extraction d'une ligne, représentant notre artère, à partir de cette surface. Nous développerons également les outils nécessaires à cette segmentation, à savoir la convolution gaussienne, la transformation de la ligne de partage des eaux, les graphes d'adjacence de régions et les champs de Markov.

Extractions d'objets : l'algorithme Fast Level Lines Transform (FLLT)

par *Christophe Berger*

Il existe de nombreux algorithmes de segmentation. Nous présenterons ici la FLLT qui se base sur le principe de composantes connexes séparées par les lignes de niveaux. En effet, il existe sur une image, comme sur une carte géographique, des lignes formées par les différences entre les zones. Dans une image ces différences sont matérialisées par la teinte ou le contraste.

Nous verrons tout d'abord les spécificités de cet algorithme et des différentes structures utilisées afin de détecter les lignes de niveaux, nous proposerons

également une comparaison avec le max-tree et présenterons les différences de ces deux méthodes. Finalement nous verrons en quoi la FLLT est intéressante dans la pratique grâce à son implémentation dans *Olena*¹¹.

Poker AI

par *Johan Oudinet*

Une des stratégies les plus utilisées pour créer une intelligence artificielle (IA) dans un jeu où les joueurs jouent à tour de rôle, consiste à construire l'arbre associé : chaque branche d'un nœud correspond à une stratégie possible du joueur qui doit jouer à ce nœud.

Le problème avec le poker, ou tout autre jeu à information imparfaite, c'est qu'on ne connaît pas les stratégies accessibles à l'adversaire (elles dépendent des cartes qu'il a reçues). Ainsi, à chaque décision prise par l'adversaire, on doit se référer à son modèle pour améliorer la qualité des décisions. Un excellent joueur de poker arrive à deviner avec précision la main de son adversaire, c'est l'objectif que doit atteindre le modèle.

Au cours de cette présentation, nous nous intéresserons au Texas Hold'em qui est une des variantes les plus complexes du poker et la plus populaire actuellement. Nous présenterons une IA utilisant un modèle d'adversaire pertinent.

Recherche d'un ϵ -Nash de support de taille 2

par *Michaël Cadilhac*

Dans le cadre de la théorie des jeux, la recherche d'un équilibre tel qu'aucun joueur ne puisse augmenter ses gains en changeant de stratégie est primordiale. De récents résultats montrent que l'obtention algorithmique de cet équilibre, dit de Nash, n'est pas abordable en temps raisonnable.

Depuis 2003 et les résultats de Lipton, on s'est rendu compte que chercher un équilibre approximatif, à un ϵ près, amenait à des algorithmes de plus faible complexité et un contexte plus humainement parlant.

Nous étudierons des algorithmes permettant la recherche d'un équilibre de Nash approximé dans un cadre où les joueurs choisissent de manière probabiliste entre deux stratégies parmi une multitude.

¹¹*Olena*, <http://olena.lrde.epita.fr>.