pas d'outil de refactorisation (des interfaces haut niveau pour manipuler des programmes : renommer, déplacer, extraire etc.). Pour attaquer le problème, le projet TRANSFORMERS utilise une technique classique : d'abord lire « grossièrement/en diagonale », puis dans une seconde étape, « désambiguïser », i.e., raffiner la compréhension que l'on a du programme. Dans ce papier, nous comparons trois environnements différents pour la désambiguïsation : l'écriture d'un programme dédié en Stratego, l'écriture d'équations de validation en ASF, et l'utilisation du système de grammaires attribuées du projet TRANSFORMERS.

Publication QUOC, C. L., BELLOT, P., AND DEMAILLE, A.. Towards the world-wide quantum network. In *Proceedings of the 4th Information Security Practice and Experience Conference (ISPEC'08)*, Sydney, Australia

Pour communiquer des informations sensibles à travers des réseaux informatiques, on utilise couramment le chiffrement (« crypter »). Ceci pose deux problèmes. D'abord les hypothèses fondamentales de la cryptologie ne sont que des hypothèses et pas des certitudes prouvées : s'il est globalement admis qu'il est « impossible » de factoriser efficacement de grand nombre, il reste possible que tout le monde ait tort et qu'un jour quelqu'un parvienne à trouver un algorithme qui rendrait immédiatement inutile toute utilisation du chiffrement. Ensuite, puisque le chiffrement repose sur le partage préalable de certaines informations secrètes, il faut avoir communiqué ces « clefs » en parfaite sécurité. . . au dessus d'un réseau informatique.

Les liens quantiques répondent à ces deux problématiques. Ils reposent sur des propriétés de la physique quantique qui garantissent l'échange secret entre deux entités sans aucune compromission. Malheureusement ces liens sont lents, de petite taille, et très coûteux. Dans cet article, nous étudions les *réseaux* quantiques, composés de liens quantiques, les faiblesses qui leur sont propres, et les moyens de les pallier.

Publication HAMEZ, A., THIERRY-MIEG, Y., AND KORDON, F.. Hierarchical set decision diagrams and automatic saturation. In *Petri Nets and Other Models of Concurrency –ICATPN 2008*

Le model-checking est une technique permettant de vérifier des propriétés sur des modélisations de systèmes, en générant leurs « espaces d'états ». Il s'agit en fait tout simplement d'automates décrivant leurs comportement. Cependant, ces espaces d'états deviennent très vite bien trop gros pour être stockés en mémoire.

De ce fait, des techniques à base de *diagrammes de décisions* ont été développées afin de les stocker de manière efficace. Pour les manipuler, il faut appliquer des opérations appelées « homomorphismes », qui sont délicates à écrire de manière optimisée. Une des ces techniques, appelée « saturation » permet de rendre locales les modifications appliquées aux diagrammes de décisions, ce qui est nécessaire pour les exploiter de manière efficace.

Pour faciliter l'utilisation des diagrammes de décisions, nous avons identifié des propriétés structurelles des opérations appliquées afin de les optimiser automatiquement en utilisant la saturation. Il est certes possible d'écrire manuellement ces optimisations, mais elles sont réellement très difficiles à concevoir. Le gain ici n'est donc pas en performance, mais en facilité d'écriture.

Cher *L'air de rien* . . .

L'air de rien écoute et répond à ses lecteurs ! Pour ce faire, rien de plus simple ; faites nous parvenir vos missives à l'adresse électronique suivante :

l-air-de-rien@lrde.epita.fr

Nous publierons les meilleurs courriers, accompagnés de la réponse de votre aléastriel préféré. À vos plumes !

L'air de rien N° 13

L'aléastriel du **L**aboratoire de **R**echerche et de **D**éveloppement de l'**E**PITA[1] Numéro 13, mars 2008

Édito

par *Alexandre Hamez (Doctorant)*

Mine de rien, la loutre littéraire et triskaïdékaphile fait sa treizième apparition pour vous présenter les actualités des entrailles du LRDE. La météo est plutôt bonne, jugez plutôt : entre un gros projet de recherche et développement collaboratif récemment validé, plusieurs articles scientifiques acceptés, le laboratoire peut se targuer d'un bon bilan sur ces dernières semaines.

Mais pour atteindre cette excellence intellectuelle en informatique, il faut d'abord savoir ce que contient un octet. Ce n'est pas aussi évident qu'il n'y paraît, car, comme le rapporte Akim Demaille, à en croire moult étudiants, il y a au moins 2^8 manières différentes de comprendre ce petit espace mémoire. . .

Peut-être auraient-ils dû se débrouiller pour atteindre un équilibre de Nash leur permettant de mettre au point une stratégie gagnante pour l'examen. . . Raison de plus pour en lire l'explication de Sébastien Hémon.

SCRIBO financé à hauteur de 2 M€ par l'État

par *Roland Levillain et Olivier Ricou (Enseignants-Chercheurs)*

Dans le cadre du 5e appel à projets du Fonds Interministériel de Soutien aux Projets de Recherche & Développement Collaboratifs des Pôles de Compétitivité, le projet SCRIBO[2], auquel participe le LRDE, a été soutenu par le pôle de compétitivité System@tic Paris-Région[3], et retenu pour financement par l'État. SCRIBO s'inscrit dans le nouveau Groupe Thématique Logiciel Libre de ce pôle.

System@tic Paris-Région a pour objectif de faire de l'Île-de-France un territoire visible au niveau mondial sur le thème de la conception, de la réalisation et de la maîtrise des systèmes complexes. Le Groupe Thématique Logiciel Libre (GTLL) a pour vocation de traiter des sujets de R&D spécifiques au logiciel libre, de contribuer à la structuration et au développement de l'offre en la matière et d'accompagner l'organisation de l'innovation qui en est issue.

SCRIBO (*Semi-automatic and Collaborative Retrieval of Information Based on Ontologies*) a pour but la mise au point d'algorithmes et d'outils collaboratifs pour l'extraction de connaissances à partir de textes et d'images. Il se distingue de l'état de l'art par son ambition de combiner les approches sémantiques et statistiques dans le traitement du langage naturel, par la prise en compte d'une dimension collaborative dans la définition et le paramétrage de règles d'extraction semi-automatique de structures, et par l'accent mis sur l'élaboration (ou le perfectionnement lorsqu'ils existent) de standards de traitement du langage naturel. SCRIBO a l'objectif de concevoir des services d'extraction d'ontologies (systèmes de représentation des connaissances) à partir de corpus de documents, d'extraction de structures dans des documents numériques et d'acquisition de connaissances en mettant en œuvre des ontologies.

Le LRDE contribuera principalement au sous-projet d'extraction de structures dans des documents numérisés. Les autres partenaires du projet, piloté

[1] L'air de rien, `http://publis.lrde.epita.fr/LrdeBulletin`.
[2] SCRIBO, `http://scribo.xwiki.com/`.
[3] System@tic Paris-Région, `http://www.systematic-paris-region.org/`.

par XPertNet, sont l'AFP, le CEA-LIST, l'INRIA, Mandriva, Nuxeo, Proxem, Tagmatica et Xwiki.

Le LRDE a déjà été amené à travailler sur des problématiques de gestion électronique de documents (GED) et de « dématérialisation » de documents papier au cours d'une collaboration de plusieurs années avec l'éditeur de logiciels EMC Captiva (ex-SWT). Le projet SCRIBO permettra d'enrichir la plateforme de traitement d'images libre Olena[4] (sous licence GNU GPL) et d'intégrer de nouveaux services dans les outils de GED du LRDE.

Qu'est-ce qu'un octet ?

par *Akim Demaille (Enseignant-Chercheur)*

Bit, Octet

Les ordinateurs sont de gigantesques machines à traiter l'information, comme le dit le mot d'« informatique » en français. Ils sont composés de vastes mémoires très différentes (optiques pour les CD/DVD, magnétiques pour la mémoire vive ou les disques durs, autrefois on utilisait des tubes de mercure, des lampes à vide etc.) dont la plus petite unité est le « bit ». Le bit code deux valeurs différentes, le nom que vous leur donnez n'importe pas (faux/vrai, vrai/faux, 0/1, bleu/rouge, up/down, ta/tech etc.) ce qui compte c'est de distinguer deux valeurs différentes. Parce que c'est plus pratique de voir ces deux valeurs comme des chiffres, on les représente généralement par 0/1. Le mot « bit » en tire son étymologie : « BInary digiT », soit « chiffre binaire » en français. C'est également un jeu de mot avec « bit of information », « un *peu* d'information ».

En règle générale, les ordinateurs ne peuvent pas accéder directement à un bit de leur mémoire : les bits sont regroupés en mémoire par paquets de taille fixe, et pour lire ou écrire un bit, en réalité on lit/écrit le paquet de bits qui contient celui qui nous intéresse. La taille de ce paquet est une caractéristique physique de la machine, et si historiquement les ordinateurs ont essayé plusieurs tailles différentes, en règle générale aujourd'hui les paquets sont de taille 8. Ils sont nommés « octets » en français.

Combien de valeurs différentes peut coder un octet ? Autrement dit, combien existe-t-il de possibilités de donner 8 fois la valeur 0 ou 1 ? Pour le premier chiffre, deux choix : 0 ou 1. Pour le second chiffre, deux choix à nouveau : 0 ou 1. Par conséquent $4 = 2 \times 2$ choix pour deux chiffres : 00, 01, 10 ou 11. Pour le troisième chiffre, 2 choix, soit $8 = 2 \times 2 \times 2$ valeurs différentes. Et ainsi de suite : avec 8 choix à faire parmi 2, on a $\underbrace{2 \times \cdots \times 2}_{n \text{ fois}}$, ce que l'on note 2^8, et qui vaut 256. Tout bon informaticien le sait par cœur : un octet code 256 valeurs différentes. Tous ?

Best of

La question a été (re)posée en partiel cette année. Un étudiant sur quatre ne connaît pas la réponse ! Mais parfois, ils sont créatifs. L'orthographe est d'origine.

- Une valeur [*Cette réponse est juste : à un instant donné un octet ne détient qu'une seule valeur.*]
- 2
- Il y a 8 valeurs différentes pour coder un octet.
- Il faut huit valeurs (car "octo" en Grec veut dire huit).
- 16 : de 0 à F(=15)
- $2^7 = 64$
- 64
- 128
- 1 octet = 8 bits $\Rightarrow 2^7 = 128$ valeurs différentes.
- De 0 à 128, donc 129 valeurs possibles.
- $2^8 - 1$
- 255 bien évidemment !
- Un octet = 8 bit soit $2^8 - 1$ possibilité = 256.
- Tout dépend du point de vue :
  - En prenant un octet sans y appliqué d'algorithme $\rightarrow$ 256.
  - En prenant le bit de poid fort comme bit de signe $\rightarrow$ 256 aussi (-128,128).
  - En y appliquant des algorithmes (loi A, loi y) on peut obtenir un panel plus important.
- $2^9 = 256$
- 258.
- $2^8 - 1 = 511$.
- $2^9 - 1$
- 2^9
- 1024
- 8^8

[4] Olena, `http://olena.lrde.epita.fr`.

Équilibre de Nash approché dans les jeux à plusieurs joueurs

par *Sébastien Hémon (Doctorant)*

Bon, que se cache donc sous ce nom barbare et anglophone (non ce n'est pas un pléonasme) ? Les équilibres de Nash sont les positions, existantes dans tout jeu à nombre fini de joueurs et de décisions, pour lesquelles aucun joueur n'a intérêt à dévier car s'il était le seul à le faire, il ne pourrait qu'y perdre. Alors après, la notion d'approximation, bien connue lorsque l'on passe sur machine, c'est pour dire qu'on tolère qu'un joueur puisse tout de même espérer gagner un ε mais après tout, il ne va pas nous embêter pour cela. La seconde partie du titre, c'est pour rappeler que l'on ne s'est pas contenté de le faire pour deux ou trois joueurs, mais bien pour autant que l'on souhaite (du moment qu'ils soient en nombre fini).

Une fois le titre décrypté (déchiffré ?), non pouvons aborder le contenu. La grande question est alors *quels sont les résultats nouveaux ?* Beaucoup de ce qui est dit est une généralisation de travaux déjà effectués dans le cas plus simple »avec deux joueurs. Nous y trouvons le fait qu'il est impossible d'espérer trouver une bonne approximation d'un équilibre en construisant une stratégie utilisant un nombre au plus logarithmique de décisions possibles. Pour simplifier, nous dirons ici que chaque joueur dispose d'un ensemble de décisions possibles et que, combinées, elles forment une stratégie. Dans la littérature, on parle en fait de stratégies et stratégies mixtes. Le problème, c'est qu'utiliser un nombre logarithmique de décisions, c'est déjà trop. En effet, la donnée d'un jeu est exponentielle : n décisions et m joueurs se représentent avec pas moins de mn^m entrées ! Même l'approximation semble hors de portée. La contrepartie positive, c'est l'élaboration d'une procédure inductive qui étend des résultats sur 2 joueurs à n joueurs. Ainsi, on peut maintenant se contenter de raisonner sur des jeux à deux joueurs, la procédure d'extension se charge du reste.

Enfin, nous montrons qu'une belle propriété sur les jeux à deux joueurs n'existe plus dès trois joueurs. Cela s'est déjà souvent révélé vrai sur d'autres, mais dans le cas présent, on n'y croyait pas vraiment. Sans rentrer dans les détails, des bornes de calcul coïncidaient dans le cas 2 joueurs, à savoir le nombre minimum de décisions à utiliser pour réaliser un ε quelconque et celui pour passer un facteur constant déterminé. Et là, les bornes ne coïncident plus : il y a ce que l'on appelle un *gap*. Brouillard total entre les deux. Bien sûr, pour $m = 2$, les valeurs coïncident (la généralisation doit conserver le cas particulier).

Les krachs boursiers ont encore de l'avenir devant eux et vous pouvez encore prétendre battre des machines à vos jeux préférés, surtout si le nombre de joueurs est élevé !

En bref

Roland Levillain a démarré depuis janvier 2008 une thèse de doctorat en informatique à l'Université Paris-Est (Marne-la-Vallée) co-encadrée par Laurent Najman, professeur à l'ESIEE (laboratoire A2SI, Institut Gaspard Monge) et par Thierry Géraud, enseignant-chercheur à l'EPITA (LRDE) sur le sujet « *Exploitation de bibliothèques C++ génériques de calcul dans des environnements dynamiques* ».

Publication DARBON, J.. Global optimization for first order Markov random fields with submodular priors. In *Proceedings of the twelfth International Workshop on Combinatorial Image Analysis (IWCIA'08)*, Buffalo, New York, USA

Ce papier décrit un algorithme d'optimisation globale d'énergies (utilisées par exemple en traitement des images, vision par ordinateur et physique statistique) qui repose sur une approche combinatoire. Nous montrons que cet algorithme calcule un minimiseur dans un cadre beaucoup plus général qu'auparavant. En outre, nous donnons des conditions nécessaires et suffisantes pour l'applicabilité de cette approche. En d'autres termes, la classe d'énergies optimisables par cette approche est complètement caractérisée.

Publication DEMAILLE, A., DURLIN, R., PIERRON, N., AND SIGOURE, B.. Semantics driven disambiguation : A comparison of different approaches. In *Proceedings of the 8th workshop on Language Descriptions, Tools and Applications (LDTA'08)*

L'analyse syntaxique (autrement la « lecture » par un ordinateur) de certains langages est extrêmement difficile. Le langage C++ compte parmi les pires cauchemars des auteurs de compilateurs. La difficulté de l'analyse du C++ explique aussi pourquoi il n'existe