



L'air de rien N° 14

Séminaires CSI de juin et juillet

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

Numéro 14, mai 2008

Édito

par *Alexandre Duret-Lutz (Enseignant-Chercheur)*

Les étudiants de l'option CSI présenteront leurs travaux à l'occasion de quatre séminaires. Ce sont les 2009 qui ouvrent le bal les 18 juin² et 25 juin³ 2008, ils seront suivis par les présentations des 2010 les 2 juillet⁴ et 9 juillet⁵ 2008.

D'autre part le 25 juin, Sébastien Hémon exposera un article qu'il a présenté en avril dernier à Paderborn en Allemagne, lors du premier symposium international sur l'algorithmique de la théorie des jeux (SAGT'08).

Ces présentations sont ouvertes à tous et nous vous invitons à venir nombreux vous y faire une idée plus précise de nos thèmes de travail.

18 juin 14h00–16h45 — amphi 4

Transformers	
14h00	SCOOOL : Programmation générique et concepts — <i>Maxime van Noppen</i>
14h30	Vers une application concrète de Transformers — <i>Florian Quèze</i>
Vaucanson	
15h15	Parser d'expressions rationnelles — <i>Vivien Delmon</i>
15h45	FSMXML et son utilisation dans Vaucanson — <i>Florian Lesaint</i>
16h15	Booster les itérateurs de Vaucanson — <i>Jimmy Ma</i>

2 juillet 14h00–16h45 — amphi 4

Olena	
14h00	Distance Transform — <i>Etienne Folio</i>
14h30	Les types de couleur dans Milena — <i>Caroline Vigouroux</i>
Speaker ID	
15h00	Système de discriminants linéaires pour la vérification du locuteur — <i>Antoine Legrand</i>
Spot	
15h45	Traduction d'une LTL étendue en TGBA dans Spot — <i>Damien Lefortier</i>
16h15	Front-end Promela dans Spot — <i>Guillaume Sadegh</i>

25 juin 14h00–17h15 — amphi 4

Olena	
14h00	Ligne de partage des eaux topologique — <i>Alexandre Abraham</i>
14h30	Taxonomie des images de Milena — <i>Nicolas Ballas</i>
15h00	Transformation des courbes de niveau rapide — <i>Matthieu Garrigues</i>
15h30	Recalage d'image rapide — <i>Ugo Jardonnet</i>
Théorie des jeux	
16h15	Étude et implémentation du Fictitious Play alterné — <i>Antoine Leblanc</i>
16h45	Équilibres de Nash approchés dans les jeux multi-joueurs — <i>Sébastien Hémon et al.</i>

9 juillet 14h00–16h45 — salle P10

Transformers	
14h00	Implémentation d'une extension du C++ dans Transformers : <code>class namespace</code> — <i>Vincent Ordy</i>
14h30	Centaur : Une infrastructure générique simplifiant les transformations de C++ — <i>Cédric Raud</i>
15h00	Désambiguïsation des patrons de type C++ avec les Grammaires Attribuées de Transformers — <i>Warren Seine</i>
Vaucanson	
15h45	Interface graphique de Vaucanson — <i>Florent D'Halluin</i>
16h15	Amélioration de la composition des transducteurs dans Vaucanson — <i>Jérôme Galtier</i>

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

²Séminaire CSI du 18 juin 2008, <http://publis.lrde.epita.fr/Seminar-2008-06-18>.

³Séminaire CSI du 25 juin 2008, <http://publis.lrde.epita.fr/Seminar-2008-06-25>.

⁴Séminaire CSI du 2 juillet 2008, <http://publis.lrde.epita.fr/Seminar-2008-07-02>.

⁵Séminaire CSI du 9 juillet 2008, <http://publis.lrde.epita.fr/Seminar-2008-07-09>.

OLENA

Ligne de partage des eaux topologique

par *Alexandre Abraham* ☉ 25 juin, 14h00, amphi 4

Segmenter une image consiste à en extraire les régions d'intérêt, par exemple pour séparer des cellules cancéreuses en imagerie médicale. L'approche par transformation de la ligne de partage des eaux (LPE) ou *Watershed Transform* permet d'obtenir une telle segmentation. Il en existe de nombreuses définitions, ainsi que diverses implémentations, dont certaines sont à la fois performantes et produisent un résultat avec de bonnes propriétés, comme le *Topological Watershed*. Cet exposé présentera l'implémentation d'un algorithme calculant cette LPE au sein de Milena, la bibliothèque C++ générique de traitement d'images de la plate-forme Olena, développée au LRDE. Nous nous intéresserons tout d'abord aux formats d'images "classiques", puis à la généralisation à des formats d'images plus inhabituels (images à support de graphe généraux, etc.).

Taxonomie des images de Milena

par *Nicolas Ballas* ☉ 25 juin, 14h30, amphi 4

Milena est la bibliothèque de traitement d'images générique de la plate-forme Olena. Cette bibliothèque a pour but d'être performante tout en restant simple. L'introduction dans Milena de nouveaux types d'images basés sur des graphes a mis en évidence des problèmes de modélisation qui sont un frein pour sa généralité. Par exemple, nous avons toujours considéré que "les images ont des points". Néanmoins, certains types d'images possèdent des sites qui ne sont pas des points (mais des arrêtes, faces, ou même des ensembles de points). Une autre supposition erronée était de considérer que les sites étaient toujours localisés par un vecteur (cad, (x,y) dans le plan 2D). Cette supposition est fautive lorsque l'on manipule des sites qui ne sont pas "Pointwise". Il était donc nécessaire de modifier les types d'images utilisés Milena et les propriétés qui leur sont associées. Pendant ce séminaire, nous présenterons une nouvelle classification d'images permettant de résoudre ces problèmes.

Transformation des courbes de niveau rapide

par *Matthieu Garrigues* ☉ 25 juin, 15h00, amphi 4

La transformation rapide des courbes de niveau (FLLT) construit une représentation d'une image in-

dépendante du contraste. Cet algorithme construit un arbre suivant les inclusions des formes. Pour un filtre, être invariant suivant le contraste est un plus. Par exemple, en analyse de document, cette représentation a le précieux avantage d'extraire facilement et rapidement les caractères indépendamment du fait qu'ils soient plus clairs ou plus foncés que leur voisinage. Ce document présente l'introduction de l'algorithme dans notre bibliothèque de traitement d'images et montre les résultats de quelques filtres connectés que peut engendrer cette représentation.

Recalage d'image rapide

par *Ugo Jardonnet* ☉ 25 juin, 15h30, amphi 4

Le recalage d'images est une technique classique en traitement d'images. Soit A et B deux images représentant le même objet (par exemple une radiographie et une image à résonance magnétique (IRM)), on calcule une transformation de A telle que le recalage de l'objet dans A soit aligné sur l'objet dans B . Typiquement, cette technique peut permettre la lecture simultanée de deux mesures A et B . Cet exposé discutera des procédés de recalage rapide utilisés dans Milena, la bibliothèque C++ générique de traitement d'images de la plate-forme Olena, développée au LRDE. Certaines améliorations seront présentées.

Distance Transform

par *Etienne Folio* ☉ 2 juillet, 14h00, amphi 4

Une carte de distances est une représentation sous forme d'image d'une fonction distance à un objet. Ces cartes sont utilisées dans de nombreuses applications, en particulier en analyse d'images de documents qui nous serviront d'illustration. Certaines méthodes de calcul de cartes moins génériques que d'autres peuvent s'avérer plus rapides : par exemple, des cartes calculées par propagation de fronts permettent de déterminer des plus courts chemins mais ne fonctionnent que lorsque le support est connu pour être non-convexe. Cette présentation fait un tour d'horizon des différents algorithmes de calculs de cartes de distances, met en évidence leurs atouts et faiblesses et explique les choix retenus.

Les types de couleur dans Milena

par *Caroline Vigouroux* ☉ 2 juillet, 14h30, amphi 4

Le projet Olena fournit une bibliothèque générique pour le traitement d'images, Milena. Nous voulons que cette bibliothèque procure de nombreux types de valeur tels que l'utilisateur puisse toujours choisir le type adapté pour son application. Par

exemple, nous fournissons de nombreux encodages en niveau de gris, de nombreux espaces de couleur, etc.

Nous présentons la manière dont nous mettons en œuvre les types de couleurs dans Milena. Il existe différents espaces de couleur (RGB, HSI, et bien d'autres) et il existe plusieurs encodages possibles pour les mêmes espaces de couleur (`rgb_3x8`, `rgb_f`,

etc.). Nous voulons rendre les choses plus faciles pour l'utilisateur. Donc, notre objectif est de rendre possible l'utilisation des espaces de couleur sans se soucier des mécanismes internes. Par exemple, dans les formules de conversion, on ne veut pas faire apparaître les détails d'implémentation (division par 255).

SPOT

Traduction d'une LTL étendue en TGBA dans Spot

par *Damien Lefortier* © 2 juillet, 15h45, amphi 4

Spot repose sur l'approche automate du *model checking*. La bibliothèque permet de vérifier des propriétés exprimées en logique temporelle à temps linéaire (LTL) sur une modélisation d'un système représentée par un automate de Büchi généralisé basé sur les transitions (TGBA). Spot propose actuellement deux algorithmes de traduction de LTL en TGBA, une des deux étapes principales de l'approche automate. Nous présentons une nouvelle traduction en TGBA d'une logique LTL qui a été étendue en y ajoutant des opérateurs représentés par des automates finis. Cette traduction permet à Spot de vérifier des propriétés qui n'étaient pas exprimables auparavant.

Front-end Promela dans Spot

par *Guillaume Sadegh* © 2 juillet, 16h15, amphi 4

Spot est une bibliothèque de model checking. Pour vérifier des modèles, Spot utilise un format d'entrée représentant des automates de Büchi généralisés basés sur les transitions (TGBA). Ce format est peu pratique pour des utilisateurs, par son manque d'abstraction et par la taille des automates à représenter, souvent composés de millions d'états. Promela (Process Meta-Language) est un langage de spécification de systèmes asynchrones, utilisé par le model checker Spin. Il permet de représenter des systèmes concurrents dans un langage impératif de haut niveau. Nous allons présenter plusieurs approches pour l'ajout d'un front-end Promela dans Spot, qui devront permettre une exploration à la volée du graphe d'états, afin d'éviter de conserver en mémoire tous les états.

SPEAKER ID

Système de discriminants linéaires pour la vérification du locuteur

par *Antoine Legrand* © 2 juillet, 15h00, amphi 4

Dans la reconnaissance du locuteur, les modèles GMM occupent une place très importante dans le développement des systèmes performants. Les mé-

thodes de discrimination linéaire à base de SVM donnent actuellement de meilleurs résultats. On s'intéressera ici à un système de discriminant linéaire (le SVM-GLDS). Celui-ci utilise directement, sans passer par un modèle GMM, des statistiques issues de l'ensemble des paramètres de la parole pour définir le modèle de reconnaissance. On évaluera les performances d'un tel système sur la base de données NIST-SRE en le comparant avec les autres systèmes à base de SVM-GMM.

VAUCANSON

Parser d'expressions rationnelles

par *Vivien Delmon* ☉ 18 juin, 15h15, amphi 4

La bibliothèque Vaucanson permet de manipuler des automates et des transducteurs. Le parser d'expressions rationnelles doit donc lui aussi traiter ces différentes structures. Malheureusement l'ancien parser ne permettait pas de lire des expressions rationnelles décrivant des transducteurs ou même des automates à poids autres que des nombres. Le nouveau parser permet de lire des expressions rationnelles contenant des poids de toutes sortes et des alphabets définis sur des produits de monoïdes. Ces différentes améliorations permettent d'interpréter des expressions rationnelles complexes représentant entre autres des transducteurs.

FSMXML et son utilisation dans Vaucanson

par *Florian Lesaint* ☉ 18 juin, 15h45, amphi 4

Nous avons commencé l'année dernière à travailler sur une nouvelle proposition de format XML de description d'automates, devenu FSMXML. Nous présentons cette année une version aboutie de ce travail sous forme de *rfc*. FSMXML comprend notamment une gestion complète des expressions rationnelles généralisées, il permet de décrire n'importe quel type d'automate et sa gestion est facilitée. Nous avons repensé la structure du parseur XML de VAUCANSON pour s'affranchir d'une mauvaise gestion de dépendances et l'avons mise à jour conformément à la *rfc*.

Booster les itérateurs de Vaucanson

par *Jimmy Ma* ☉ 18 juin, 16h15, amphi 4

Vaucanson est une bibliothèque générique de manipulation d'automates. Le cœur de sa généricité réside dans le support de types d'automates variés mais aussi sa capacité à s'appuyer sur différentes structures de données. Actuellement, nous avons différentes manières de manipuler des transitions. Ce-

pendant, aucune d'entre elles n'est réellement indépendante de la structure de données utilisée. Afin de pallier cela, nous allons nous tourner vers le design pattern Iterator. Nous évaluerons l'impact de ce design pattern sur les performances et sur l'utilisation de la bibliothèque en termes d'écriture d'algorithmes.

Interface graphique de Vaucanson

par *Florent D'Halluin* ☉ 9 juillet, 15h45, salle P10

Vaucanson est une plateforme de manipulation d'automates finis. Débuté en 2002, le projet attire de plus en plus d'utilisateurs. De ce fait, une interface utilisateur efficace est nécessaire.

Pour l'utilisateur non expert, la manipulation d'automates peut s'effectuer via *taf-kit*, une suite d'outils accessible en ligne de commande. Une première interface graphique avait été esquissée en 2005, mais son fonctionnement était lent et compliqué car elle s'appuyait sur *taf-kit* pour réaliser chaque opération.

Cette nouvelle interface graphique, branchée directement sur le cœur de la bibliothèque pour plus d'efficacité, simplifie la manipulation d'automates et rend accessible les algorithmes génériques de Vaucanson.

Amélioration de la composition des transducteurs dans Vaucanson

par *Jérôme Galtier* ☉ 9 juillet, 16h15, salle P10

Vaucanson est une bibliothèque dont un des buts est de permettre un accès facilité à des automates et aux algorithmes qui leur sont associés. Elle met donc à notre disposition plusieurs algorithmes standard (et d'autres moins conventionnels) tels que la détermination, le calcul des états accessibles etc. L'un de ces algorithmes est la composition de transducteurs. Celui-ci n'est pas d'une nature aisée à aborder et son implémentation dans Vaucanson est perfectible. Améliorer l'implémentation d'un tel algorithme est alors un bon moyen de mettre à l'épreuve certains choix de conception dans Vaucanson.

TRANSFORMERS

Vers une application concrète de Transformers

par Florian Quèze ☉ 18 juin, 14h30, amphi 4

Transformers est un ensemble d'outils permettant la manipulation de programmes C++ en Stratego/XT. Pendant le développement de Transformers, la plupart des tests sont effectués sur de toutes petites portions de code en C++ (des tests unitaires). Sur des exemples de tailles plus conséquentes, certains outils sont lents, inefficaces, voire totalement inutilisables.

Le but de ce travail est d'analyser les difficultés qui apparaissent lorsque les fichiers traités sont volumineux et d'y remédier. Cette étude portera à la fois sur les comportements incorrects de Transformers dans cette situation et sur les adaptations nécessaires des outils permettant aux développeurs de visualiser les étapes du traitement.

Une preuve de succès serait de parvenir à réaliser une analyse statique ou une transformation assez simple sur quelques fichiers du code en C++ d'un logiciel libre largement diffusé.

SCOOL : Programmation générique et concepts

par Maxime van Noppen ☉ 18 juin, 14h00, amphi 4

SCOOL est un langage statique orienté objet qui a été créé afin de pouvoir utiliser toute la puissance du C++ statique de manière plus aisée grâce à une syntaxe plus expressive et agréable. Il n'a pas pour but d'être directement compilé mais d'être traduit en C++. Cette année le travail revêt une importance particulière. En effet, SCOOL est développé en étroite collaboration avec l'équipe de développement de la bibliothèque de traitement d'images MILENA de la plate-forme OLENA ; l'an passé a été pour elle le cadre de grands changements internes. Un des axes majeurs du développement de SCOOL va donc être de s'adapter aux nouveaux paradigmes et aux nouveaux besoins de la bibliothèque. Le second axe essentiel de travail est la poursuite du développement du langage. Cette année le travail va être concentré sur la programmation par concepts qui est une approche permettant de formaliser facilement des contraintes sur la programmation générique.

Implémentation d'une extension du C++ dans Transformers : class namespace

par Vincent Ordy ☉ 9 juillet, 14h00, salle P10

Les classes en C++ sont fermées, c'est-à-dire qu'on ne peut rien leur ajouter une fois leur définition terminée. Or, la plupart du temps, les programmeurs séparent la définition de l'implémentation, ce qui oblige à utiliser une syntaxe répétitive, en particulier dans le cas de patrons de classes ou de classes imbriquées. On se propose donc de faire une extension de la grammaire du C++ permettant via une syntaxe proche de celle des namespaces de définir plus aisément des méthodes ou attributs statiques déjà déclarés dans la définition de la classe. Dans ce but, nous utiliserons la grammaire du C++ implémentée dans Transformers, et des transformations écrites en Stratego.

Centaur : Une infrastructure générique simplifiant les transformations de C++

par Cédric Raud ☉ 9 juillet, 14h30, salle P10

La grammaire du standard du C++ n'ayant pas été conçue pour être aisément analysable, son utilisation dans le cadre de la manipulation de programme est comparable à la complexité de l'AST généré par celle-ci. Le rôle de Centaur au sein de Transformers est ainsi de fournir une infrastructure générique permettant de manipuler et de synthétiser cet AST : les transformations de programmes sont simplifiées grâce à un accès plus aisé aux informations contenues dans l'arbre syntaxique et ses annotations. Grâce à cette bibliothèque, les tâches répétitives et souvent génératrices d'erreurs, comme l'énumération des éléments d'un conteneur ou la recherche des classes parentes d'une classe, seront factorisées par un ensemble de fonctions correspondant à un modèle modulaire et extensible.

Désambiguïsation des patrons de type C++ avec les Grammaires Attribuées de Transformers

par Warren Seine ☉ 9 juillet, 15h00, salle P10

Malgré sa sensibilité au contexte, le C++ est analysable avec une grammaire hors-contexte mais ambiguë. La désambiguïsation est ensuite nécessaire pour sélectionner le seul arbre syntaxique sémantiquement valide. Transformers est une collection

d'outils pour la transformation de programmes C++ qui utilise les grammaires attribuées pour réaliser cette étape.

Une des plus difficiles ambiguïtés dans le langage concerne la méta-programmation. Puisque du code est généré à l'instanciation, tous les types ne sont pas nécessairement connus à la déclaration. La vérifica-

tion des types est donc obligatoire pour traiter totalement le cas des patrons, ce qui pose un véritable défi.

Ce rapport se concentre sur la désambiguïssation des patrons de type et détaille les problèmes et leur méthode de résolution, afin de fournir une meilleure plateforme de manipulation de sources.

THÉORIE DES JEUX

Étude et implémentation du Fictitious Play alterné

par *Antoine Leblanc* © 25 juin, 16h15, amphi 4

Le calcul d'un équilibre de Nash dans un jeu fini est un problème démontré PPAD-complet, ce qui signifie qu'il paraît impossible de trouver une méthode de calcul efficace ; la complexité en pire cas des algorithmes usuels est $2^{O(n)}$ pour un jeu de taille n . La recherche en ce domaine s'oriente donc vers le calcul d'équilibres approchés, à savoir des situations vérifiant les conditions d'un équilibre de Nash à ε près.

L'algorithme du *Fictitious Play* s'inscrit dans cette démarche de recherche. Son principe est simple : à chaque itération, chacun des joueurs "renforce" celle de ses stratégies pures qui est la plus efficace face à ses adversaires. Pour certains jeux, cet algorithme converge vers un équilibre de Nash, fournissant ainsi un algorithme d'approximation efficace. La convergence ne peut toutefois être prouvée que pour un nombre limité de cas.

Pour cette raison, il est intéressant d'étudier d'autres algorithmes basés sur le *Fictitious Play*, afin de trouver d'autres cas de convergence. Nous allons étudier ici le *Fictitious Play* alterné, dans lequel seul le joueur le plus "éloigné" de son gain optimal renforce sa stratégie la plus efficace.

Équilibres de Nash approchés dans les jeux multi-joueurs

par *Sébastien Hémon et al.* © 25 juin, 16h45, amphi 4

Les équilibres de Nash sont des positions-clés de tout jeu admettant une représentation finie : en effet, quel que soit le nombre de joueurs et de stratégies, une telle position existe toujours. Lorsqu'elle est atteinte, elle dissuade tout joueur de vouloir se détourner de sa stratégie actuelle, d'où la notion d'équilibre. De nombreux problèmes y font appel mais calculer de façon effective l'équilibre demeure un problème difficile. En effet, le meilleur algorithme connu pour, dans le cas général, calculer un équilibre est exponentiel en le nombre de stratégies.

Nous présenterons ici la notion d'équilibres approchés, et donnerons des résultats concernant leur calcul. Nous montrerons qu'il ne saurait exister d'algorithmes pouvant calculer un équilibre, même approché, sans utiliser au moins, pour un joueur, un nombre logarithmique de stratégies. Nous montrerons comment calculer un équilibre approché en temps sub-exponentiel $n^{O(\frac{\ln n}{\varepsilon^2})}$, ce qui demeure actuellement, pour le cas général, la meilleure complexité en pire cas.

Enfin, nous présenterons une approche inductive de transfert d'approximation d'une position d'un jeu à deux joueurs en une approximation pour un jeu à r joueurs, ce qui confère des résultats novateurs dans le domaine.