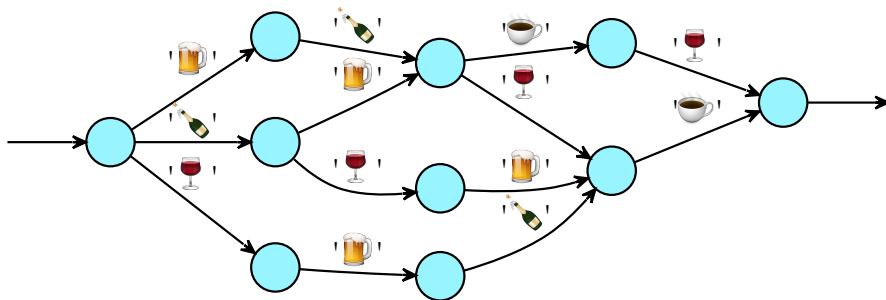


Bien entendu, construire un automate n'est pas suffisant : il faut disposer d'une batterie d'algorithmes pour les manipuler. C'est en utilisant ces constructions, et en les améliorant, que Sébastien Piat, stagiaire épitéen, a réécrit en Vcsn une application de François Yvon, chercheur au LIMSI, qui permet de traduire le langage SMS en "bon" français :

```
sms_to_fr('slt') # => salut
sms_to_fr('bjr') # => bonjour
sms_to_fr('stp') # => s'il te plait
sms_to_fr('cmt c tro bil!')
# => comment c'est trop bien!
```

Cette traduction est bien plus qu'une substitution bête et méchante : il n'y a pas de table qui à chaque "mot sms" lui fasse correspondre un mot "traduit".

```
ctx.expression('"' + 🍷 + ':' + 🍷 + ':' + 🍷 + ':' + 🍷 + '"'
               & ([^]*('🍷'+'🍷')*🍷+[^]*){c}
               & [^]*'🍷'[^]*'🍷'[^]*'🍷'[^]*'🍷').automaton().minimize()
```



Disponibilità

Utiliser Vcsn n'a jamais été aussi simple ! Pour le télécharger, rendez-vous sur sa page⁴. Mais pour

En réalité, à partir d'automates qui modèlisent le français, il s'agit véritablement de calculer la traduction la plus vraisemblable.

De puissantes expressions rationnelles

Vcsn offre une palette d'opérateurs assez rares. Ainsi, tous les mélanges de bière, champagne, vin rouge et café 🍺🍷🍷🍷 tels que la bière passe toujours avant le café ([^]*🍺 [^]*🍷 [^]*) et (&) on ne boit pas ({c}) le vin rouge ou le café (+) immédiatement avant le champagne ([^]*🍷(+🍷🍷🍷 [^]*🍺) sont donnés par l'automate suivant.

le déployer, si docker est installé sur votre machine, il suffit de lancer `docker run -d -p 8888:8888 lr-de/vcsn` ! Enfin, une version en ligne⁵ est disponible, avec des exemples.

En bref

Un nouveau doctorant



Le LRDE est heureux d'accueillir un nouveau doctorant : Ludovic Le Frioux.

Diplômé d'un Master à l'UPMC (Paris VI) en Système et Applications Répartis, il a intégré l'équipe Spot du LRDE en octobre 2015 dans le cadre de sa thèse en collaboration avec l'équipe MoVe du LIP6. Sa thèse dirigée par Fabrice Kordon (LIP6) et Souheib Baarir (LRDE) a pour but de paralléliser un SAT-solveur.

Contactez le LRDE

18, rue Pasteur, Le Kremlin-Bicêtre
Paritalie, bâtiment X, aile Mistral
2e étage, droite droite

Tél. : 01 53 14 59 22

Contact: info@lrde.epita.fr

Les permanents : lrde@lrde.epita.fr

Site Web : <http://www.lrde.epita.fr>

... et surtout, passez nous voir ;
vous serez toujours les bienvenus !



L'air de rien
N° 35

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

Numéro 35, Décembre 2015

Édito

par *Olivier Ricou*

Et de deux! notre deuxième épitéen que l'on a pris tout petit au LRDE a soutenu sa thèse après presque sept ans au labo. Edwin est donc maintenant docteur Edwin. Demain, le 11 décembre, c'est Ana qui soutient. Elle est notre première thésarde et donc notre première docteure. Vous trouverez un résumé de leurs thèses dans ce numéro. Comme la nature a horreur du vide, l'espace libéré par ces jeunes docteurs s'est rempli par les quatre nouveaux thésards : Alexandre, Jim et Duy que l'on vous a présentés au numéro précédent² et Ludovic qui est présent dans celui-ci.

En terme de publications, le rythme de croisière

du labo a connu un coup de chauffe puisq'on a produit treize articles dans les six derniers mois alors qu'il s'agit plutôt de notre régime annuel. Comme quoi les records sont faits pour être dépassés. Mais on ne va quand même pas vous infliger la liste de tous ces articles et leurs résumés aussi a-t-on choisi de faire un numéro dédié aux automates avec un article sur la nouvelle version de Vcsn qui vous invite à boire plus que de raisonnable, une sélection d'articles produits par l'équipe Spot et une présentation d'Alexandre, élève-chercheur dans cette équipe. En vous souhaitant automatiquement une bonne lecture.

De la thèse au doctorat

Ana Calarasanu



Le sujet de ma thèse s'inscrit dans le contexte du projet LINX qui a comme but l'assistance des malvoyants dans leur environnement naturel (maison, rue etc.). Notamment, il s'agit d'une application pour mobile capable de détecter le texte dans des images

quelconques et de rendre cette information textuelle au format audio. Les contributions de cette thèse s'inscrivent dans la partie de localisation de texte dans une image. Durant les dernières années il y a eu une forte croissance de tels systèmes qui a amplifié le besoin d'avoir une évaluation précise qui reflète leur performance (afin par exemple d'obtenir un classement correct entre ces systèmes). Malheureusement, peu de protocoles existent et ces protocoles manquent souvent de précision soit parce qu'ils n'arrivent pas à gérer la diversité des résultats produits par les détecteurs soit parce qu'ils proposent des métriques non-représentatives. Dans cette thèse, nous proposons un nouveau protocole d'évaluation qui résout la plupart des problèmes rencontrés dans les méthodes d'évaluation actuelles. Nous montrons

enfin comment nous pouvons analyser un ensemble de résultats de détections, non seulement à travers des mesures et certaines règles mais aussi à travers une représentation visuelle intuitive qui nous permet d'avoir une image claire du comportement d'un détecteur de texte.

Edwin Carlinet



De nombreuses applications issues de la vision par ordinateur et de la reconnaissance des formes requièrent une analyse de l'image multi-échelle basée sur ses régions. De nos jours, personne ne considérerait une approche orientée « pixel » comme une solution viable pour traiter ce genre de problèmes. Pour répondre à cette demande, la Morphologie Mathématique a fourni des représentations hiérarchiques des régions de l'image telle que l'Arbre des Formes (AdF). L'AdF représente l'image par un arbre d'inclusion de ses lignes de niveau ce qui fait de lui une structure bien adaptée au traitement d'images de haut niveau. Néanmoins, il est seulement défini pour les images en niveaux de gris et la

4. Projet Vcsn, <https://vcsn.lrde.epita.fr>.

5. Vcsn en ligne, <http://vcsn-sandbox.lrde.epita.fr>.

1. L'air de rien, http://www.lrde.epita.fr/wiki/L'air_de_rien.

2. toujours en vente sur votre site web préféré, cf l'url ci-dessus.

plupart des tentatives d'extension aux images multivariées (e.g. les images couleurs) ne sont pas satisfaisantes. Nous présentons ainsi une nouvelle approche pour étendre l'AdF scalaire aux images multivariées : l'Arbre des Formes Multivarié (AdFM). Ce dernier est une structure facile à manipuler, polyvalente, et efficace. Pour valider la pertinence de l'approche, nous illustrons l'utilisation de l'AdFM à tra-

vers différentes applications de traitement d'images et de vision par ordinateur telles que la segmentation interactive d'images ou le suivi de documents dans les vidéos. De ces applications, nous montrons ainsi que les méthodes basées sur l'AdFM rivalisent avec les résultats de l'état de l'art, démontrant ainsi le potentiel de notre approche.

Le model checking selon Spot

Dans le domaine du model checking, on utilise des automates pour représenter des ensembles de séquences d'exécutions infinies représentant les comportements possibles ou les comportements autorisés d'un système. Par exemple un scénario d'évolution d'un carrefour muni de feux tricolores peut être vu comme une séquence d'états donnant la configuration des feux tricolores au fur et à mesure que le temps avance. Le scénario n'est pas unique : il dépend des interactions avec l'environnement (piétons qui demandent à traverser, capteurs de présence des voitures, etc.).

Un automate modélisant le carrefour permet de représenter l'ensemble (infini) des scénarios possibles de façon compacte (et finie). De même, un automate peut être utilisé pour représenter un ensemble (infini) de scénarios interdits : par exemple on pourrait créer l'automate représentant l'ensemble des scénarios du carrefour où deux feux perpendiculaires sont verts. À partir de là, on peut, avec des opérations sur ces automates, vérifier que l'intersection de ces deux automates est vide : c'est-à-dire qu'il n'existe pas de scénarios possibles dans lesquels deux feux perpendiculaires seraient verts.

La bibliothèque Spot permet de manipuler de tels automates pour effectuer ce genre de vérification.

Pour vous aider à comprendre ce qu'est la recherche dans ce domaine, voici quatre articles récents qui s'intéressent chacun à un point précis. Chacun est un petit pas qui fait avancer l'ensemble.

Les automates "bégayants"

Le premier cas est le résultat du stage d'infoSpé de Thibaud Michaud. C'est rare qu'un tel stage aboutisse à un article mais avec un peu de travail autour c'est possible.

Le but est de vérifier efficacement et de plusieurs façons si un automate est "insensible au bégaiement". Un automate a cette propriété si lorsque qu'on considère un scénario qu'il représente (donc une séquence infinie d'états) et que l'on duplique un ou plusieurs de ces états au sein de la séquence, alors on obtient une autre séquence qui était déjà reconnue. Être capable de détecter cette propriété est important car cela autorise plein d'optimisations dans le processus de vérification. Jusqu'à présent les outils utilisent une procédure approchée qui ne permet pas de dé-

tecter tous les automates insensibles au bégaiement. Grâce à ce travail qui a donné lieu à une publication, nous avons maintenant une procédure qui est à la fois précise et facile à implémenter.

MICHAUD, T. AND DURET-LUTZ, A.. Practical stutter-invariance checks for ω -regular languages. In *Proceedings of the 22th International SPIN Symposium on Model Checking of Software (SPIN'15)*, volume 9232 of *Lecture Notes in Computer Science*, pages 84–101. Springer

Les automates "testeurs"

De nombreux types d'automates peuvent être utilisés pour le model checking. Les "automates testeurs" ne sont pas habituellement utilisés bien qu'ils donnent des résultats intéressants. Dans ces automates on n'encode pas des configurations (par exemple la lampe est allumée ou la lampe est éteinte) mais des changements de configuration (la lampe vient de changer d'état). Ces automates sont typiquement utilisables uniquement pour représenter des propriétés insensibles au bégaiement. La mise en place d'une technique qui généralise l'idée au cas de propriétés sensibles au bégaiement a donc constitué un challenge pour la diffusion de ce type d'automates.

BEN SALEM, A. E.. Extending testing automata to all ltl. In *Proceedings of the 35th IFIP International Conference on Formal Techniques for Distributed Objects, Components and Systems (FORTE'15)*, volume 9039 of *Lecture Notes in Computer Science*, Grenoble, France. Springer

Un format d'échange !

Dans la communauté du model checking, de nombreux outils existent chacun manipulant des types différents d'automates. Afin de pouvoir comparer les outils et les techniques mises en place, il est nécessaire de pouvoir s'échanger simplement des automates. La mise en place d'un format d'échange a donc constitué l'aboutissement d'un consensus entre des développeurs de plusieurs outils pour rendre ces outils interopérables. Un point important du format est qu'il présente la "condition d'acceptation" de l'automate, c'est-à-dire les règles qui indiquent quels chemins infinis de l'automate doivent être pris en

compte ou ignorés, d'une façon qui non seulement unifie les différentes conditions d'acceptation classique utilisées jusqu'à présent, mais aussi les généralise.

BABIAK, T., BLAHOUEK, F., DURET-LUTZ, A., KLEIN, J., KRÉTINSKÝ, J., MÜLLER, D., PARKER, D., AND STREJČEK, J.. The hanoi omega-automata format. In *Proceedings of the 27th International Conference on Computer Aided Verification (CAV'15)*, volume 9206 of *Lecture Notes in Computer Science*, pages 479–486. Springer

Et de la minimisation !

La généralisation de la condition d'acceptation introduite par le format décrit ci-dessus a permis d'envisager de nouvelles façons de minimiser les automates. Cette minimisation est très importante pour le model checking car elle permet de traiter plus efficacement certains problèmes, et donc a fortiori des problèmes plus complexes. Cette nouvelle minimisation utilise un SAT-solveur et pourra donc bénéficier des avancées qui seront réalisées dans le cadre de la thèse de Ludovic (présenté plus loin).

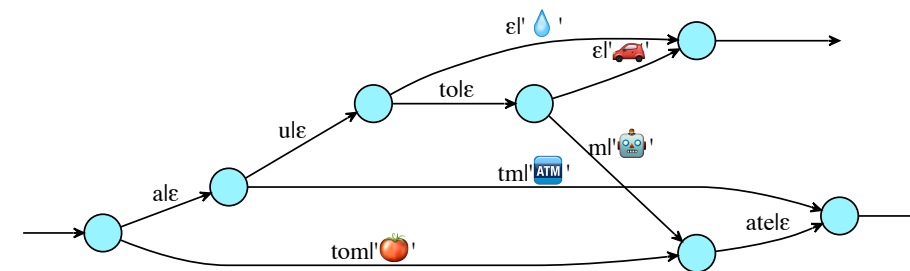
BAARIR, S. AND DURET-LUTZ, A.. SAT-based minimization of deterministic ω -automata. In *Proceedings of the 20th International Conference on Logic for Programming, Artificial Intelligence, and Reasoning (LPAR'15)*, volume ???, pages ??–??? Springer

Vcsn 2.1

par Akim Demaille



Descendant de Vaucanson, Vcsn est une plate-forme libre (GNU GPLv3+) dédiée aux automates et aux expressions rationnelles. Son efficacité est assurée par une couche C++, dont l'utilisation des templates assure la généralité. L'environnement graphique et interactif exploitant IPython/Jupyter en fait un outil de choix pour l'enseignement et l'expérimentation.



3. Vcsn 2.1, <http://vcsn.lrde.epita.fr/Vcsn2.1>.

Retour d'un étudiant-chercheur

par Alexandre Lewkowicz



Lors de mon stage d'infoSpé, j'ai eu la chance de découvrir le LRDE en participant au développement du projet Vcsn. L'implémentation de 12 algorithmes à partir de définitions mathématiques et l'ambiance du laboratoire m'ont donné goût au monde de la recherche. De plus, voir son code tourner sur un projet open source était assez gratifiant. L'année suivante, j'ai décidé d'intégrer le LRDE en tant qu'étudiant chercheur (CSI) sur le projet de Spot. Pendant mes deux années CSI, j'ai pu travailler sur des problématiques liées au model checking comme la conversion d'automates de Büchi non-déterministes en automates déterministes. L'étude de nombreux articles et le travail en collaboration avec Alexandre Duret-Lutz (enseignant chercheur au LRDE) m'ont permis d'élaborer une nouvelle méthode qui produit des automates déterministes avec peu d'états. Suite à ce travail collectif, je vais pouvoir commencer la rédaction de mon premier article qui sera soumis à une conférence.

Des transducteurs !

Vcsn 2.1³, publié en octobre dernier, apporte des changements importants dans le support de *transducteurs*, i.e., des automates qui "traduisent". Ils sont désormais représentés comme le veut la tradition : 'foo|bar' s'interprète comme "si on lit foo en entrée, produire bar en sortie", et ε désigne le mot vide. Les linguistes sont particulièrement friands des transducteurs, et bien sûr, il leur faut également un alphabet bien plus riche que le vénérable ASCII. Nous pouvons désormais construire des automates comme celui qui suit, qui traduit certains mots (dont un avec une belle faute d'orthographe !) en symbole Unicode.