

ROYAUME DU MAROC

OFFICE MAROCAIN DE LA PROPRIETE (19)
INDUSTRIELLE ET COMMERCIALE



المملكة المغربية

المكتب المغربي
للملكية الصناعية والتجارية

(12) FASCICULE DE BREVET

(11) N° de publication : **MA 37413 A1** (51) Cl. internationale : **G06F 17/00**

(43) Date de publication :
31.05.2016

(21) N° Dépôt :
37413

(22) Date de Dépôt :
10.10.2014

(71) Demandeur(s) :
UNIVERSITE INTERNATIONALE DE RABAT UIR, PARC TECHNOPOLIS RABAT-SHORE, CAMPUS UNIVERSITAIRE UIR, ROCADE RABAT-SALE, 11100 (MA)

(72) Inventeur(s) :
Younes MOUMEN

(74) Mandataire :
BOUYA MOHSINE

(54) Titre : **SYSTÈME DE MODÉLISATION ET DE TEST D'AUTOMATES FINIS**

(57) Abrégé : Un système mis en oeuvre par ordinateur qui fournit des outils de modélisation d'automates finis. Lesystème intègre des fonctions pour tester les automates modélisés, les minimiser, les rendre déterministes, les compléter. Plusieurs automates peuvent être combinés pour générer un analyseur syntaxique sous forme de librairie qui peut être intégrée dans tout programme informatique exécutable.

Abrégé

Un système mis en œuvre par ordinateur qui fournit des outils de modélisation d'automates finis. Le système intègre des fonctions pour tester les automates modélisés, les minimiser, les rendre déterministes, les compléter. Plusieurs automates peuvent être combinés pour générer un analyseur syntaxique sous forme de bibliothèque qui peut être intégrée dans tout programme informatique exécutable.

31 MAI 2016

Système de modélisation et de test d'automates finis

Description

L'invention est relative aux systèmes mis en œuvre par ordinateur pour la modélisation et le traitement des automates finis.

Nous commencerons par définir les automates finis qui sont traités par notre invention.

Un automate fini ou automate fini non déterministe (AFN) \mathcal{A} sur un alphabet A est un quadruplet $\mathcal{A} = (Q, \mathcal{F}, I, T)$, où :

- Q est un ensemble fini d'états,
- $\mathcal{F} \subset Q \times A \times Q$ est l'ensemble des transitions,
- $I \subseteq Q$ est l'ensemble des états initiaux,
- et $T \subseteq Q$ est un ensemble d'états finaux ou terminaux.

Une transition $f = (p, a, q)$ est composée d'un état de départ p , d'une étiquette a et d'un état d'arrivée q . Un calcul c (on dit aussi un chemin ou une trace) est une suite de transitions consécutives :

$$c = (p_0, a_1, p_1)(p_1, a_2, p_2) \cdots (p_{n-1}, a_n, p_n)$$

Son état de départ est p_0 , son étiquette est le mot $a_1 a_2 \cdots a_n$ et son état d'arrivée est p_n . Un calcul est réussi si son état de départ est l'un des états initiaux, et son état d'arrivée est l'un des états terminaux.

Un mot w est reconnu ou accepté par l'automate s'il est l'étiquette d'un calcul réussi.

Le langage reconnu par l'automate est l'ensemble des mots reconnus. Un langage est reconnaissable s'il est reconnu par un automate fini.

- Un automate est complet si pour tout état q , et pour toute lettre a , il existe au moins une transition partant de q et portant l'étiquette a .
- Un état q est accessible s'il existe un chemin d'un état initial à q .
- Un état q est coaccessible s'il existe un chemin de q à un état final.
- Un automate est accessible (coaccessible) si tous ses états sont accessibles (coaccessibles).

- Un automate est émondé si tous ses états sont à la fois accessibles et coaccessibles.

Un automate fini déterministe (AFD) \mathcal{A} sur un alphabet A est un automate fini qui vérifie les deux conditions suivantes :

- il possède un seul état initial ;
- pour tout état q , et pour toute lettre a , il existe au plus une transition partant de q et portant l'étiquette a .

Pour un automate déterministe, la fonction de transition $\delta : Q \times A \rightarrow Q$ est la fonction partielle définie par : $\delta(q, a) = q'$ si (q, a, q') est une transition. Si la fonction de transition est partout définie, l'automate est complet. La fonction de transition δ est étendue en une application (partielle) $Q \times A^* \rightarrow Q$ en posant

- $\delta(q, \varepsilon) = q$ pour tout état q . Ici ε dénote le mot vide.
- $\delta(q, wa) = \delta(\delta(q, w), a)$ pour tout état q , tout mot w et toute lettre a .

Les automates sont des machines abstraites qui sont utilisées dans plusieurs domaines ayant plusieurs applications dans l'industrie. Nous les retrouvons dans la modélisation de processus, le contrôle, les protocoles de communication, la vérification de programmes, la théorie de la calculabilité, dans l'étude des langages formels et en compilation. Ils sont également utilisés dans la recherche des motifs dans un texte.

Il existe plusieurs inventions dans l'état de la technique qui permettent d'extraire des automates finis à partir de bases de données, conceptions de circuits électriques, expressions régulières, etc. L'objectif étant de tester les automates finis, les minimiser, les compléter, etc.

Notre invention a comme objectif la modélisation initiale (sans importation de données préalables) d'automates finis afin de les traiter avec un certain nombre d'algorithmes. Ensuite, l'automate peut être exporté dans un format de fichier persistant comme il peut être utilisé pour générer des expressions régulières ou des analyseurs syntaxiques utilisés par les compilateurs de langages de programmation.

Notre invention est mise en œuvre par ordinateur en fournissant une interface graphique avec laquelle l'utilisateur peut interagir en utilisant la souris, le clavier et un écran qui peut être tactile. L'interface graphique est composée d'un espace de travail (1) et d'un certain nombre de commandes (2) accessibles par des menus déroulants et des raccourcis clavier :

- Nouveau : cette commande réinitialise l'affichage pour commencer la modélisation
- Ouvrir : cette commande ouvre un automate existant

- **Enregistrer** : cette commande enregistre à nouveau l'automate dans le fichier persistant déjà choisi au préalable selon le format déjà choisi. Si l'automate n'a pas encore été enregistré, cette commande se comporte comme la commande Enregistrer sous
- **Enregistrer sous ...** : cette commande affiche une interface pour permettre à l'utilisateur de choisir l'emplacement et le format de fichier à enregistrer
- **Sélectionner** : cette commande permet de basculer en mode sélection. Dans ce mode, l'utilisateur peut sélectionner des états ou des transitions grâce à la souris ou l'écran tactile
- **Ajouter état** : cette commande permet de basculer en mode ajout d'état. Dans ce mode, l'utilisateur peut ajouter un état à l'endroit de son choix grâce à la souris ou l'écran tactile
- **Ajouter transition** : cette commande permet de basculer en mode ajout de transition. Dans ce mode, l'utilisateur peut ajouter une transition entre états de son choix grâce à la souris ou l'écran tactile
- **Supprimer** : cette commande permet de supprimer l'état ou la transition sélectionnée
- **Définir alphabet** : cette commande permet de définir l'alphabet sous forme de succession de caractères
- **Définir point de départ** : cette commande définit l'état sélectionné comme état de départ
- **Propriétés** : cette commande permet de définir les propriétés de l'état ou de la transition sélectionnée
- **Générer** : cette commande permet de générer une expression régulière ou un analyseur syntaxique à partir de l'automate sous forme de librairie qui peut être intégrée dans tout programme informatique exécutable
- **Minimiser** : cette commande permet de minimiser l'automate
- **Tester** : cette commande permet d'entrer une chaîne de caractères à tester sur l'automate. Le résultat étant l'acceptation ou pas de la chaîne.
- **Compléter** : cette commande permet de compléter l'automate
- **Déterminer** : cette commande permet de rendre l'automate déterministe
- **Est complet** : cette commande permet de calculer si l'automate est complet ou pas
- **Est déterministe** : cette commande permet de calculer si l'automate est déterministe ou pas

Nous décrivons dans ce qui suit les instructions pour manipuler notre invention selon un cas d'utilisation :

Pour créer et éditer un automate, il faut définir la liste des entrées de l'automate (alphabet). Pour ce faire, il faut cliquer sur la commande « Définir alphabet ». Une boîte de dialogue s'ouvre contenant une zone d'édition du texte. Ils y seront affichées les anciennes entrées si elles existent. Il faut taper la nouvelle liste des entrées comme suit : la liste {a}, {b}, {c} sera tapée abc. Lorsque la modification est faite, il faut cliquer sur OK afin de la confirmer sinon sur Annuler.

Il faut ensuite entrer les états. Cliquer sur la commande « Ajouter état ». Le programme attendra que l'utilisateur spécifie la position de l'état avec un clique de souris gauche dans la zone d'espace de travail. Une boîte de dialogue s'ouvrira contenant un champ pour le nom où l'utilisateur devra entrer le nom du prochain état et activer une case à cocher si le nouvel état est acceptant. Appuyer sur OK pour confirmer. Le premier état ajouté est par défaut l'état de départ.

Une fois tous les états ajoutés. L'utilisateur procède à l'ajout des transitions. Il clique sur la commande « Ajouter transition ». Il clique ensuite sur l'état d'où partira la transition, puis sur un endroit où sera affichée l'entrée correspondante et enfin, il clique sur l'état où aboutira la transition. Une fois toutes ces opérations faites, une boîte de dialogue lui demandera de choisir une entrée pour cette transition.

Lorsque l'utilisateur crée tous les états et les transitions nécessaires il est possible de modifier l'automate. Pour sélectionner un état ou une transition, il suffit de cliquer avec le bouton droit de la souris (ou utiliser la commande « Sélectionner ») puis cliquer avec le bouton gauche de la souris sur l'état ou la transition désirée. Une fois un état ou une transition sélectionnée, l'utilisateur peut double cliquer dessus (ou utiliser la commande « Propriétés ») afin de modifier ses propriétés. Aussi, il peut le déplacer en maintenant le bouton gauche de la souris enfoncé et glisser l'objet dans une nouvelle position. Sinon la touche supprimer du clavier (ou la commande « supprimer ») permet de supprimer l'état ou la transition en question. Sauf que lorsque l'utilisateur supprime un état, il supprime également toutes les transitions qui en sortent et celles qui y aboutissent.

Lorsque l'utilisateur termine la modélisation de l'automate, il peut l'enregistrer et commencer à le tester avec des chaînes d'entrées. Pour le tester il n'a qu'à appuyer sur la touche Entrer du clavier (ou la commande « Tester »). Une boîte de dialogue sera ouverte pour demander la chaîne de caractères à tester. Aussi, il peut minimiser, compléter ou déterminer (entre autres) l'automate. Il peut également calculer si l'automate est complet ou s'il est déterministe. Tout ceci se trouve dans le menu des commandes.

L'invention peut (entre autres) être programmée en langage C++ à cause de sa puissance et des avantages de l'héritage qu'il présente. Cette particularité peut être exploitée par l'usage de la hiérarchie de classes MFC (Microsoft Foundation Classes) qui est d'une grande utilité dans les fins de la programmation événementielle sous le système d'exploitation Windows.

Voici l'ensemble des classes utilisées dans l'invention :

CAutomata : qui manipule les structures ARC, VERTEX et PVERTEX. C'est cette classe qui contient toutes les données en rapport avec l'automate. Les structures contiennent les informations et CAutomata les manipule à l'aide de listes doublement chaînées pour permettre une implémentation dynamique.

CAutomataApp : Cette classe est dérivée de la classe CWinApp des MFCs. Cette classe contrôle le noyau de l'application (gestion des Threads et de la relation entre les différents composants du programme).

CMainFrame : est dérivée de CFrameWnd des MFCs. Elle gère exclusivement la fenêtre principale et ses éléments comme le menu.

CAutomataDoc : Se charge de la manipulation des documents (ouverture, enregistrement ...) à l'aide de sa classe mère CDocument des MFCs.

CAutomataView : Répond aux commandes de l'utilisateur et les lie avec la classe CAutomata. Cette classe se charge aussi de l'affichage de la zone cliente. Elle est dérivée de la classe CView des MFCs.

CAboutDlg, CDlgArc, CDlgVertex, CDlgInputs et CDlgTest : Des classes dérivées de CDialog des MFCs pour la manipulation des boîtes de dialogue.

La figure 1 donne une vue de l'interface graphique de l'invention.

La figure 2 représente un état simple de l'automate.

La figure 3 représente un état acceptant de l'automate.

La figure 4 représente l'état de départ de l'automate.

La figure 5 représente une transition partant de l'état 2 vers l'état 3 avec l'entrée a.

La figure 6 représente une boucle sur l'état 2 avec l'entrée b.

La figure 7 représente l'automate correspondant à l'expression régulière $(a + b)^3 (a^* b^*)^*$

Revendications

1- Un système mis en œuvre par ordinateur caractérisé par une interface graphique contenant un espace de travail et des commandes de modélisation, d'importation, d'exportation, de test, de génération d'expression régulière et de minimisation d'un automate fini, ainsi que des commandes pour le rendre complet ou déterministe et des commandes pour tester s'il est complet et déterministe. L'interface graphique contient également une commande pour générer une expression régulière ou un analyseur syntaxique à partir de l'automate fini.

Dessins

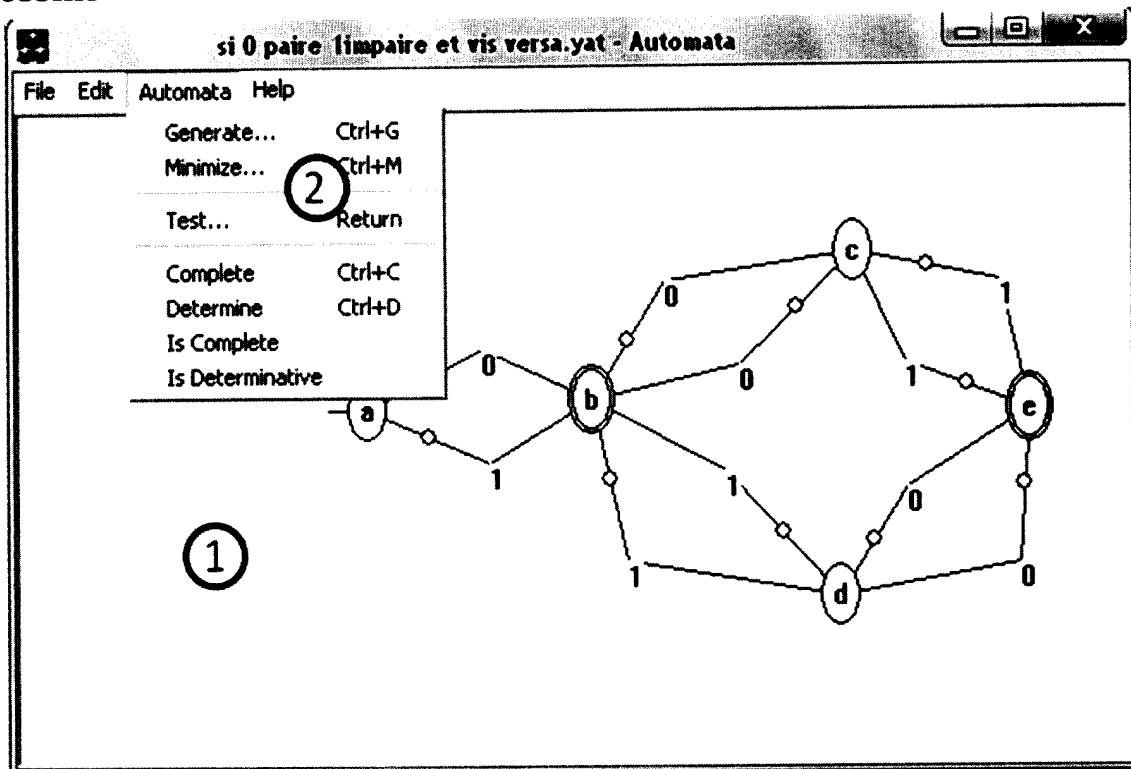


Figure 1



Figure 2



Figure 3



Figure 4

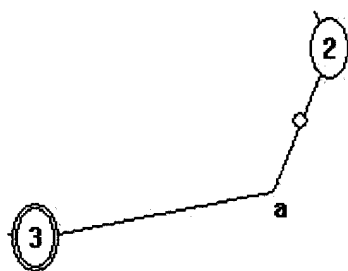


Figure 5

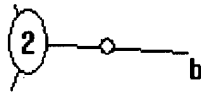


Figure 6

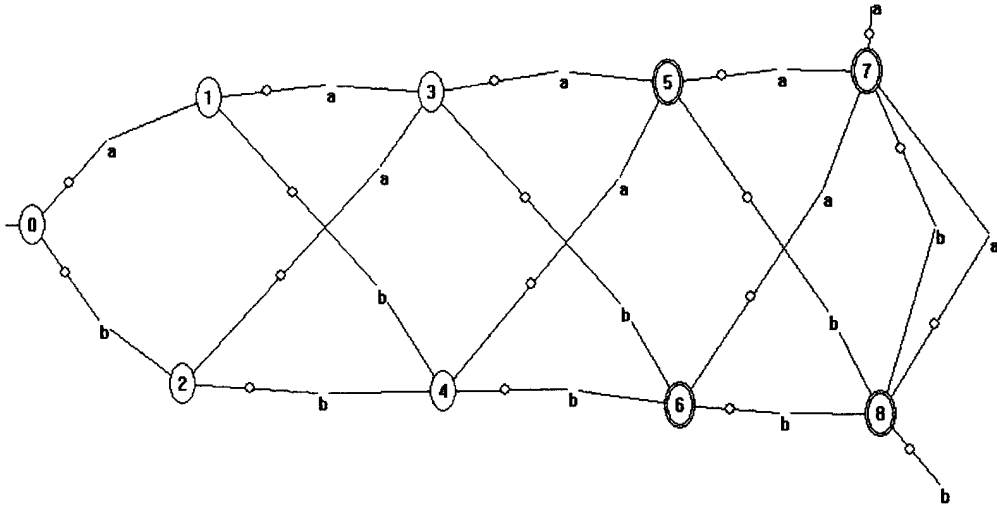
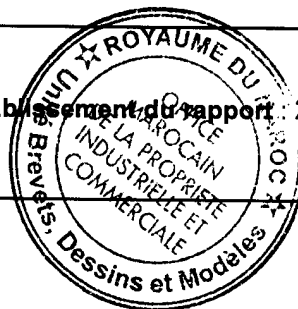


Figure 7



**RAPPORT DE RECHERCHE PRELIMINAIRE AVEC
OPINION SUR LA BREVETABILITE**

Renseignements relatifs à la demande	
N° de la demande : 37413	Date de dépôt : 10/10/2014
Déposant : UNIVERSITE INTERNATIONALE DE RABAT PRIVEE UIR	Date de Priorité :
Intitulé de l'invention : SYSTÈME DE MODÉLISATION ET DE TEST D'AUTOMATES FINIS	
<p>Le présent document est le rapport de recherche préliminaire avec opinion écrite sur la brevetabilité établi par l'OMPIC conformément à l'article 43 et notifié au déposant conformément à l'article 43.1 de la loi 17/97 relative à la protection de la propriété industrielle.</p> <ul style="list-style-type: none"> - Le présent rapport est constitué de pages (la présente page incluse) - Les documents cités par l'examinateur dans la partie Rapport de recherche sont joints au présent document 	
<p>Le présent rapport contient des indications relatives aux éléments suivants :</p> <p>Partie 1 : Considérations générales</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Cadre 1 : Base du présent rapport <input type="checkbox"/> Cadre 2 : Priorité <input type="checkbox"/> Cadre 3 : Titre et/ou Abrégé tel qu'ils sont définitivement arrêtés <p>Partie 2 : Rapport de recherche</p> <p>Partie 3 : Opinion sur la brevetabilité</p> <ul style="list-style-type: none"> <input checked="" type="checkbox"/> Cadre 4 : Remarques de clarté <input checked="" type="checkbox"/> Cadre 5 : Déclaration motivée quand à la Nouveauté, l'Activité Inventive et l'Application Industrielle <input type="checkbox"/> Cadre 6 : Observations à propos de certaines revendications dont aucune recherche significative n'a pu être effectuée <input type="checkbox"/> Cadre 7 : Défaut d'unité d'invention 	
Examinateur: N KARTIT	Date d'établissement du rapport : 29/09/2015
Téléphone: 212 5 22 58 64 14/00	
Email : kartit@ompic.ma	



Partie 1 : Considérations générales

Cadre 1 : base du présent rapport

Les pièces suivantes de la demande servent de base à l'établissement du présent rapport :

- Description
5 Pages
- Revendications
1

Partie 2 : Rapport de recherche

Classement de l'objet de la demande :

CIB : G06F 11/36

CPC :

Bases de données électroniques consultées au cours de la recherche :

EPOQUE, Espacenet, Orbit

Catégorie*	Documents cités avec, le cas échéant, l'indication des passages pertinents	N° des revendications visées
X	WO1999028820 A1 10 juin 1999 Christophe Broult, Centre Nat Rech Scient, Samuel Dellacherie, Samuel Devulder, France Telecom, Lambert Jean Luc, Univ Caen,	1
A	http://ehess.modelisationsavoirs.fr/marc/ens/langages/2006/AutChapitre2.pdf	1

***Catégories spéciales de documents cités :**

-« X » document particulièrement pertinent ; l'invention revendiquée ne peut être considérée comme nouvelle ou comme impliquant une activité inventive par rapport au document considéré isolément
 -« Y » document particulièrement pertinent ; l'invention revendiquée ne peut être considérée comme impliquant une activité inventive lorsque le document est associé à un ou plusieurs autres documents de même nature, cette combinaison étant évidente pour une personne du métier
 -« A » document définissant l'état général de la technique, non considéré comme particulièrement pertinent
 -« P » documents intercalaires ; Les documents dont la date de publication est située entre la date de dépôt de la demande examinée et la date de priorité revendiquée ou la priorité la plus ancienne s'il y en a plusieurs
 -« E » Éventuelles demandes de brevet interférentes. Tout document de brevet ayant une date de dépôt ou de priorité antérieure à la date de dépôt de la demande faisant l'objet de la recherche (et non à la date de priorité), mais publié postérieurement à cette date et dont le contenu constituerait un état de la technique pertinent pour la nouveauté

Partie 3 : Opinion sur la brevetabilité*Cadre 4 : Remarques de clarté*

4.1 Conformément aux exigences de l'article 34 de la loi 17/97, la description de l'invention doit être suffisamment claire et complète en divulguant des informations suffisantes permettant à un homme de métier, sans expérience excessive, d'exécuter l'invention.

4.2 La revendication 1 ne satisfait pas aux exigences de l'art. 35 de la loi 17/97 modifiée et complétée par la loi 23/13 car l'objet de la protection demandée n'est pas défini. La revendication tente de définir l'objet par le résultat recherché. En tout état de cause, cette formulation n'est pas acceptable en l'espèce, puisqu'il semble possible de définir l'objet en des termes plus concrets, c'est-à-dire en exposant comment l'effet peut être obtenu.

Cadre 5 : Déclaration motivée quant à la Nouveauté, l'Activité Inventive et l'Application Industrielle

Nouveauté (N)	Revendications 1 Revendications aucune	Oui Non
Activité inventive (AI)	Revendications aucune Revendications 1	Oui Non
Possibilité d'application Industrielle (PAI)	Revendications 1 Revendications aucune	Oui Non

Il est fait référence aux documents suivants. Les numéros d'ordre qui leur sont attribués ci après seront utilisés dans toute la suite de la procédure

D1 : WO1999028820

1. Nouveauté (N) :

Aucun des documents cités ci-dessus ne divulgue l'ensemble des caractéristiques techniques énoncées dans la revendication 1.

D'où, l'objet des revendications 1 est nouveau au sens de l'art. 26 de la loi 17/97 telle que modifiée et complétée par la loi 23/13.

2. Activité inventive (AI) :

Le problème adressé par la présente demande consiste à réaliser un système de modélisation et de test d'automates finis. Le système comprend une interface graphique contenant des commandes de modélisation, d'importation, d'exportation, de test.

Les techniques informatiques décrites dans la revendication 1 sont considérées être universellement connues, largement répandues et accessibles à tous à la date de dépôt de la demande ci-présente. La notoriété d'un tel art antérieur ne peut être raisonnablement contestée. Aucune preuve écrite n'est donc considérée nécessaire.

Par conséquent, cette solution consiste pour la personne du métier une démarche technique normale qui n'implique pas une activité inventive.

3. Possibilité d'application industrielle (PAI) :

L'objet de la présente invention présente une utilité spécifique, substantielle et crédible, au sens de l'art. 29 de la loi 17/97 modifiée et complétée par la loi 23/13.