PUBLICATIONS MATHÉMATIQUES ET INFORMATIQUES DE RENNES

DOMINIQUE PY

La démonstration dans les EIAO de la géométrie

Publications de l'Institut de recherche mathématiques de Rennes, 1998, fascicule S4
« Produire et lire des textes de démonstration », , p. 69-78

<http://www.numdam.org/item?id=PSMIR_1998___S4_69_0>

LA DÉMONSTRATION DANS LES EIAO DE LA GEOMETRIE

Dominique Py

Introduction

Parmi les disciplines qui ont donné lieu à la réalisation d' EIAO (Enseignement Intelligemment Assisté par Ordinateur), la géométrie tient une place de choix. Cependant, si de nombreux logiciels traitent du tracé et de la manipulation de figures ou de la recherche et de l'élaboration d'une démonstration, très peu s'intéressent à la rédaction du texte de la démonstration. Les concepteurs de logiciels privilégient la structure logique de la preuve plutôt que sa représentation textuelle, et ceci pour plusieurs raisons. D'une part, avec les progrès réalisés dans le domaine des interfaces homme-macine, les interfaces basées sur les menus et la souris se sont révélées plus conviviales et plus simples à réaliser que les interfaces en langage naturel. D'autre part, l'analyse et la compréhension automatique d'un texte en langage naturel reste un problème ouvert, d'autant plus ardu si ce texte comporte des erreurs ou des fautes, comme c'est le cas dans une rédaction libre d'un élève. Les auteurs de logiciels ont donc choisi de contourner la difficulté, en limitant les possibilités de représentations textuelles : les phrases produites sont composées d'éléments prédéfinis, les phrases analysées doivent respecter une syntaxe donnée. Deux logiciels accordent une place plus importante à la rédaction : Mentoniezh, qui permet à l'élève de rédiger sa propre démonstration en assemblant des propriétés, des théorèmes et des mots de liaison ; ARRIA, qui propose à l'élève des fragments de démonstration à ordonner de manière logique.

Dans cet article, nous nous intéresserons tout d'abord aux travaux de recherche en informatique concernant la démonstration automatique en géométrie et le traitement du langage naturel. Puis nous aborderons le domaine de l' EIAO, en cherchant dans quelle mesure il hérite des deux précédents. Enfin, nous décrirons plus en détail les logiciels Mentoniezh et ARRIA qui traitent des textes de démonstration dans le contexte scolaire.

Les origines : la recherche en intelligence artificielle

L'intelligence artificielle (IA) est un champ de l'informatique qui vise à développer des programmes basés sur l'utilisation de connaissances. Parmi les domaines d'application qui relèvent de l'IA, deux thèmes concernent plus particulièrement notre sujet : la démonstration automatique et l'analyse du langage naturel.

La démonstration automatique

La démonstration automatique a intéressé très tôt les chercheurs en intelligence artificielle. Le premier programme qualifié de "programme d'intelligence artificielle" est en effet le *Logic Theorist* de Newell et Simon, réalisé en 1956, qui effectue des démons-

trations simples en calcul des prédicats (logique d'ordre 0). Les travaux de Robinson aboutissent en 1965 à la définition d'une méthode générale de résolution, applicable à un ensemble de formules logiques du premier ordre. Cette méthode sert de base au langage de programmation *Prolog*, défini par Colmerauer en 1970.

Du côté de la géométrie, un premier démonstrateur automatique apparaît en 1963 : la *Geometry Machine* de Gelernter [Gelernter 63]. L'objectif de Gelernter est de réduire la complexité de la recherche, afin d'aboutir rapidement à une solution. Il choisit pour cela d'utiliser le chaînage arrière (aller du but vers les hypothèses) et de donner un rôle prépondérant à la figure.

Dans la *Geometry Machine*, la figure est représentée sous forme cartésienne. Un point est défini par un doublet de réels et un segment par un doublet de points. La figure est décrite par les listes des points et des segments mentionnés dans les hypothèses du problème à résoudre. Les propriétés de la figure sont exprimées par des prédicats portant sur des points. Par exemple, une relation entre deux triangles est traduite par une relation entre six points. Les théorèmes sont exprimés sous la forme "Si (hypothèses) alors (conclusion)".

La méthode de démonstration est basée sur le chaînage arrière qui permet de générer un graphe de preuve. Le problème est résolu lorsque chaque but est déduit des hypothèses et des axiomes. On sait que par chaînage arrière, il est possible de générer des expressions en contradiction avec les hypothèses de départ. Pour éviter ceci, la validité de chaque expression générée est vérifiée à l'aide de la figure, ce qui permet d'élaguer le graphe de preuve et donc de réduire la complexité de la recherche.

Le résultat de la démonstration est un arbre de preuve dont la racine est le but et les feuilles sont les hypothèses. A partir de cet arbre, on produit une preuve en établissant la liste des sous-buts démontrés, dans l'ordre usuel (des hypothèses vers le but). Voici un exemple de preuve généré par *Geometry Machine* :

Hypothèses
Angle ABD = angle DBC
Segment AD perpendiculaire au segment AB
Segment DC perpendiculaire au segment BC

But
Segment AD = segment CD

Solution
Angle ABD = angle DBC (hypothèse)
DAB est un angle droit (définition angle droit)
DCB est un angle droit (définition angle droit)
Angle DAB = angle DCB (angles droits égaux)
DB est un segment (figure)
Segment BD = segment BD (identité)
BCD est un triangle (figure)
BAD est un triangle (figure)
ADB est congruent à CDB (côté - angle - angle)
Segment AD = segment CD (triangles congruents)

L'objectif de l'auteur étant de produire efficacement des preuves, la forme du résultat reste secondaire. On remarque dans cet exemple des lourdeurs dues aux contraintes d'implémentation du démonstrateur, qui auraient été évitées dans un texte rédigé à la main. Par exemple, il faut établir explicitement l'égalité BD = BD pour pouvoir ensuite démontrer que les triangles ADB et CDB sont congruents.

Avec la *Geometry Machine*, Gelernter pose des bases communes aux démonstrateurs géométriques : une représentation des faits à l'aide de prédicats, des règles de déduction de la forme si-alors, l'exploration d'un graphe de recherche. Les variantes étudiées par la suite concernent essentiellement le vocabulaire retenu, qui définit le domaine de problèmes abordé, et les méthodes d'exploration du graphe. Ainsi, Nevins propose en 1975 un démonstrateur qui allie chaînage avant et chaînage arrière au cours de la phase de

recherche [Nevins 75]. Coelho et Pereira réalisent en 1979 une implémentation en Prolog qui exploite la représentation basée sur les prédicats de ce langage [Coelho 79].

Cependant, la recherche de l'efficacité et de la généralité des démonstrateurs conduit actuellement les chercheurs à se tourner vers les méthodes algébriques, et ce au détriment de la lisibilité des solutions. Pour réaliser des démonstrateurs automatiques, le calcul s'avère plus efficace que la déduction.

Le traitement automatique du langage naturel

Le traitement automatique du langage naturel (TALN), aussi appelé informatique linguistique, est un domaine de recherche qui vise à modéliser les processus du langage de manière à automatiser son traitement. Il s'agit de spécifier des algorithmes qui permettent de comprendre et de générer du langage naturel sous sa forme orale ou écrite. Les objectifs sont plus pragmatiques que ceux de la linguistique théorique : on cherche à développer des procédures pour traiter un ensemble utile d'entrées et on accepte des solutions qui couvrent la plupart des phrases intéressantes, même s'il y a quelques échecs.

Les applications du TALN concernent le développement d'interfaces en langage naturel restreint, l'interrogation souple de bases de données, la traduction automatique. Plus récemment, la recherche s'est focalisée sur l'obtention d'informations à partir de textes : indexation, recherche documentaire, résumé automatique.

L'analyse du langage naturel se fait à plusieurs niveaux, qui ne correspondent pas nécessairement à une réalité psychologique mais plutôt au découpage fonctionnel utilisé dans les systèmes. La *morphologie* rend compte de la forme des mots dans une phrase : conjugaison des verbes, pluriel des noms communs, accord des adjectifs. La *syntaxe* est l'ensemble des règles qui décrivent la distribution des mots dans les phrases en termes de combinaisons admises de classes de mots. La *sémantique* détermine la signification des phrases. Enfin, la *pragmatique* vise à interpréter les phrases en les replaçant dans leur contexte d'énonciation.

Langage naturel vs langage formel

Par rapport aux langages formels, les langages naturels présentent des caractéristiques qui rendent leur analyse d'une grande complexité formelle et technique, même lorsqu'il s'agit de ne traiter qu'un sous-ensemble bien délimité de la langue. Ces difficultés font apparaître les limites actuelles de l'analyse automatique du langage naturel.

Un langage naturel comporte un ensemble a priori non déterminable de phrases bien formées : on ne connaît pas totalement le vocabulaire d'une langue, ni l'ensemble des structures grammaticales, ni l'ensemble des sens attribuables à chaque mot. Une seconde difficulté tient à la notion de compréhension : une phrase en langage naturel peut être incorrecte du point de vue grammatical et cependant compréhensible par l'interlocuteur. Il existe un seuil de distorsion syntaxique au-delà duquel la phrase devient incompréhensible, mais ce seuil varie en fonction de la signification de la phrase et de son contexte d'énonciation. Enfin, le langage naturel contient des ambiguïtés (ellipses, métaphores, litotes...) dont l'analyse pose des problèmes considérables car elle nécessite une représentation riche du contexte.

Grammaires formelles

Une grammaire formelle est un ensemble de règles de dérivation qui définit l'ensemble des structures grammaticalement bien formées. C'est un outil qui permet de produire ou d'analyser des phrases d'un point de vue purement syntaxique. Considérons l'exemple suivant :

P —> **GN GV**
GN —> **Article Nom commun**
GN —> **Nom propre**
GV —> **Verbe GN**

Analyser une phrase avec cette grammaire revient à construire l'arbre syntaxique dont la racine est **P** et les feuilles sont les mots de la phrase. Par exemple, pour la phrase "Marie voit le chat", on obtient :

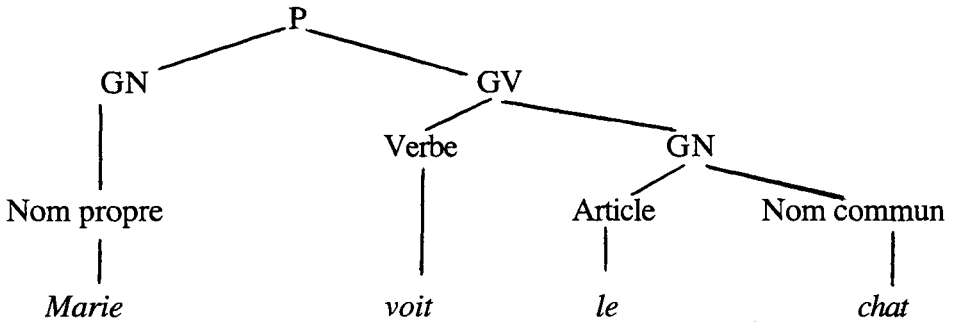

FIG. 1 - *Arbre syntaxique*

Des phrases syntaxiquement formées peuvent n'avoir aucun sens. Pour contrôler la validité sémantique d'une phrase, il faut ajouter des contraintes sémantiques à la grammaire. Chaque mot du lexique se voit attribuer des traits sémantiques, puis on établit les relations exprimant que les traits sémantiques d'un mot dans une position syntaxique donnée (sujet, complément) doivent vérifier certaines propriétés par rapport aux traits sémantiques d'autres mots de la phrase.

Par exemple, "Marie" aura pour trait sémantique "humain", "Médor" aura pour trait sémantique "animal" et le nom commun "rose" aura pour trait sémantique "plante". De plus, le verbe "cueillir" devra avoir comme sujet un nom de type "humain" et comme complément un nom de type "plante". Ces contraintes amènent à accepter la phrase "Marie cueille une rose" et à rejeter la phrase "Médor cueille une rose", dont la construction syntaxique est valide mais qui ne respecte pas la concordance des traits sémantiques.

Bilan

Démonstration automatique et TALN sont, dans les faits, deux domaines bien distincts, aux finalités différentes. La démonstration automatique, à partir d'un énoncé déjà formalisé, cherche à produire efficacement une démonstration dont la forme finale importe peu. Le TALN vise à analyse et comprendre des phrases dans le cadre d'un dialogue homme-machine.

Un rapprochement de ces deux domaines est pourtant envisageable lorsque la démonstration est faite **pour** ou **par** un élève, car la forme de la preuve devient alors un point crucial de la communication entre l'humain et la machine.

Informatique et enseignement de la géométrie

La rencontre de l'informatique, de l'enseignement et de la géométrie a donné lieu à plusieurs familles de réalisations :

- des résolveurs "pédagogiques" qui cherchent à produire des démonstrations en restant proches des méthodes de résolution des experts humains,
- des environnements d'apprentissage, de type tuteur ou micro-monde, qui permettent à un élève de s'entraîner à résoudre des problèmes de démonstration.

Résolution pédagogique en géométrie

Les solutions produites par les démonstrateurs classiques ne sont pas des solutions "naturelles", c'est pourquoi ils ne pourraient pas être utilisés dans un cadre pédagogique. Partant de ce constat, des chercheurs ont voulu réaliser des démonstrateurs dont le comportement soit plus proche de celui des experts humains. Sur le domaine de la géométrie, on peut citer :

GEOMUS, qui se base sur un mécanisme d'enrichissement et d'étiquetage du problème, afin de déclencher les règles les plus pertinentes [Bazin 93] ;

UBL, qui analyse ses propres résolutions afin d'apprendre de nouvelles règles et méthodes, et s'auto-améliore [Pintado 94].

Si les méthodes de résolution employées par ces systèmes sont plus naturelles, la forme des preuves produites reste stéréotypée. Voici, par exemple, une preuve fournie par UBL et "mise en forme" par le système après élimination des étapes inutiles.

Hypothèses
ABCD est un parallélogramme
M est le milieu de [AB]
N est le milieu de [CD]

But
AMCN est un parallélogramme

Solution
(0) ABCD est un parallélogramme (hypothèse)
(13) M est le milieu de [AB] (hypothèse)
(14) M est dans [AB] (automatisme [13])
(15) N est le milieu de [CD] (hypothèse)
(16) N est dans [CD] (automatisme [15])
(73) (AB) // (CD) (définition [0])
(74) AMCN est non croisé (théorème [0 , 13 , 15])
(97) CN = DN (définition [15])
(98) (BC) // (AD) (définition [0])
(100) (AD) // (MN) (théorème [13 , 15 , 98])
(101) ADNM est un parallélogramme (définition [14 , 16 , 73 , 100])
(104) AM = DN (définition [14 , 16 , 101])
(105) AM = CN (transitivité [97 , 104])
(106) AMCN est un parallélogramme (théorème [14 , 16 , 73 , 74 , 105])

Plus d'une centaine de faits ont été démontrés, mais UBL ne présente que ceux qui sont nécessaires à la compréhension de la preuve, en indiquant la manière dont ils ont été obtenus.

EIAO de la démonstration en géométrie

Les premiers environnements d'apprentissage de la géométrie sont apparus dans le milieu des années 80. Ils concernent pour la plupart le domaine de la géométrie euclidienne plane, telle qu'elle est enseignée au collège [Anderson 85] [Bernat 94] [Desmoulins 94] [Giorgiutti 91] [Laborde 95] [Nicolas 89]. Ces logiciels ont bénéficié des progrès réalisés dans le domaine des interfaces graphiques, qui ont permis le développement d' EIAOs de la géométrie dans l'espace.

Les tâches proposées par ces logiciels sont variées et recouvrent la plupart des activités de l'élève : tracé de la figure, exploration du problème, démonstration, rédaction. Les possibilités d'interaction et les interfaces graphiques ont de plus permis l'apparition d'une nouvelle activité : la manipulation de la figure, qui permet à l'élève de déformer la figure tracée à l'écran tout en conservant ses propriétés.

Nous allons décrire les tâches de démonstration et de rédaction telles qu'elles sont proposées à l'élève dans ces logiciels, en nous intéressant plus particulièrement au traitement du texte de la démonstration, puis nous étudierons deux logiciels traitant la phase de rédaction de la preuve.

Les tâches de démonstration et de rédaction

Dans la plupart des logiciels, la tâche de démonstration est organisée globalement selon le même schéma. L'élève construit sa preuve pas à pas, chaque pas étant formé d'un ensemble d'hypothèses, d'un théorème et d'une conclusion. Le système valide le pas proposé ou le rejette avec un message d'erreur.

En ce qui concerne l'interface, la saisie se fait par sélection dans des listes prédéfinies de théorèmes ou de propriétés, ces propriétés étant ensuite complétées par les noms des objets (points, droites, cercles) saisis au clavier. L'état de la démonstration est visualisé soit par un graphe, qui met en évidence la structure de la preuve, soit par des listes de propriétés classées selon leur statut (donnée, conclusion, résultat intermédiaire...).

Ces choix de conception facilitent la saisie et simplifient les contrôles d'erreur portant sur la syntaxe des expressions ou le typage. Ils ont l'avantage de mettre l'accent sur la structure logique de la preuve. En revanche, la forme textuelle de la preuve n'apparaît pas au cours de cette étape.

La tâche de rédaction n'est généralement pas prise en charge par la machine : on considère que l'élève rédige sur papier à partir de la démonstration produite avec la machine. Deux logiciels font exception : ARRIA et Mentoniezh [Bruillard 91] [Py 96].

Mentoniezh traite de manière successive les étapes de démonstration et de rédaction. L'élève rédige donc la preuve qu'il vient de réaliser. Le domaine considéré est celui de la géométrie plane du collège.

ARRIA traite uniquement la phase de rédaction. L'élève compose une rédaction à partir de fragments préparés par le professeur. ARRIA peut donc traiter n'importe quel domaine.

La rédaction dans Mentoniezh

Le logiciel Mentoniezh comporte trois phases distinctes et successives : l'analyse de l'énoncé, la démonstration et la rédaction. Lorsque l'élève arrive à la phase de rédaction, il a déjà réalisé une preuve, sous la forme d'une liste de pas composés chacun d'une ou plusieurs hypothèses, d'un théorème et d'une conclusion. La rédaction se fait pas à pas, chaque pas donnant lieu à la rédaction d'une phrase. Rédiger une phrase consiste à composer une séquence de termes pris parmi les éléments du pas ou parmi une liste de connecteurs :

on sait que
et
or
donc
parce que
, (virgule)
. (point)

Pendant la saisie, aucun contrôle n'est effectué sur la validité de la phrase. Le contrôle n'intervient que lorsque l'élève pense avoir composé une phrase correcte et demande sa validation.

Le diagnostic s'effectue selon deux critères. La validité logique est assurée si tous les éléments du pas (hypothèses, théorème, conclusion) sont présents exactement une fois. La validité syntaxique est contrôlée au moyen d'une grammaire formelle qui définit les tournures considérées comme correctes. Une phrase doit être composée d'un bloc hypothèses, un bloc théorème et un bloc conclusion, dans n'importe quel ordre, mais reliés par des connecteurs adaptés, par exemple :

Phrase —> "on sait que" Bloc-H "or" Bloc-T "donc" Bloc-C "."
Phrase —> Bloc-C "car" Bloc-T "et" Bloc-T "."
...

De même, chaque bloc se décompose de plusieurs manières en éléments de base :

Bloc-H —> Hypothèse "et" Hypothèse
Bloc-H —> Hypothèse
Bloc-T —> "on sait que" Théorème
...

A partir de l'énoncé suivant :

> Soit ABC un triangle isocèle en A. La parallèle à (AB) passant par C coupe la parallèle à (AC) passant par B en un point K.
> Montrer que ABKC est un losange.

voici un exemple de texte de démonstration produit avec Mentoniezh :

> On sait que (AB) // (CK) et (AC) // (BK), or si un quadrilatère a ses côtés opposés parallèles deux à deux, alors c'est un parallélogramme, donc ABKC est un parallélogramme.
>
> ABKC est un losange parce que AB = AC et ABKC est un parallélogramme, or si un quadrilatère est un parallélogramme et s'il a deux côtés consécutifs de même longueur, alors c'est un losange.

Cette approche permet à l'élève de composer un texte et de s'assurer de sa correction. Elle comporte néanmoins des limites : les phrases sont rédigées séquentiellement et sont toutes basées sur le même modèle, il n'est donc pas possible de regrouper deux pas en une phrase, comme cela se produit fréquemment dans les textes.

La rédaction dans ARRIA

Le logiciel ARRIA (Aide au Raisonnement et à la Rédaction en Intelligence Artificielle) est entièrement consacré à la rédaction. Il se base sur le principe du puzzle : l'élève dispose d'un ensemble de fragments, conçus par le professeur, qu'il doit ordonner logiquement de manière à reconstituer un texte cohérent. Le travail avec ARRIA se déroule suivant trois étapes : une lecture active de l'énoncé, utilisant des aides hypertexte, le typage des fragments de preuve (en hypothèse, conclusion, outil ou résultat intermédiaire) et enfin la rédaction proprement dite. Pour rédiger son texte, l'élève compose une séquence de termes pris parmi les fragments ou parmi une liste de connecteurs :

ou
et
or
donc
car

, (virgule)
; (point-virgule)
. (point)

Contrairement à Mentoniezh, ARRIA effectue les contrôles sur la rédaction au fur et à mesure de la saisie. Ces contrôles portent sur la validité logique et la validité syntaxique de la rédaction.

Les liens logiques entre fragments sont fournis par le professeur au moment de la conception de l'exercice. Ils sont exprimés au moyen d'une liste de déductions possibles sous la forme (prémisses) —> conclusion. Par exemple, les relations suivantes :

(1) —> 3
(2,3) —> 4
(5) —> 4
...

indiquent que le fragment 3 se déduit du fragment 1 et que le fragment 4 se déduit soit des fragments 2 et 3, soit du fragment 5. Cette représentation permet d'accepter des rédactions différentes de la part de l'élève, du moment que les relations logiques entre fragments sont respectées.

La validité syntaxique de la rédaction est vérifiée au moyen d'une grammaire formelle. Cette grammaire définit les tournures classiques :

Phrase —> Hypothèses "or" Outil "donc" Résultat
Phrase —> Outil "donc" Résultat "car" Hypothèses "."
...

Elle autorise également l'utilisation implicite de l'outil :

Phrase —> Hypothèses "donc" Résultat
Phrase —> Résultat "car" Hypothèses
...

et enfin elle permet de regrouper plusieurs pas de déduction en une seule phrase, évitant ainsi de citer à nouveau un résultat qui vient d'être obtenu. Par exemple, avec les déductions logiques :

(H1,O) —> R1
(R1) —> R2
(R1,H2) —> R3
...

les rédactions suivantes seront licites :

H1 et O donc R1. R1 donc R2.	
H1 et O donc R1 donc R2.	Pas de rappel de R1
H1 donc R1 car O donc R2.	Pas de rappel de R1 éloigné
H1 et O donc R1 or R2 donc R3.	Enchâssement avec "or"
H1 et 0 donc R1 donc R3 car H2.	

Cette grammaire offre donc des possibilités plus variées que celle employée dans Mentoniezh.

Le logiciel ARRIA a été diffusé et expérimenté en classe. Le lecteur intéressé pourra se reporter à [Bruillard 1991] pour plus de détails.

Conclusion

Nous avons vu qu'il n'existait pas d'interaction, au sein de la recherche en intelligence artificielle, entre démonstration automatique et traitement du langage naturel. Ces domaines peuvent néanmoins converger en EIAO, du fait de la situation scolaire qui met en avant la forme textuelle de la démonstration.

Toutefois, on peut s'interroger sur l'intérêt que représenterait une interaction plus forte entre ces deux domaines, et sur les obstacles qui restent à lever.

En ce qui concerne la production de textes de démonstration, la difficulté majeure est liée à la notion d'implicite. Les démonstrateurs automatiques traitent généralement toutes les inférences par le même mécanisme. Pour produire ensuite un texte de démonstration, il faut pouvoir distinguer ce qui est essentiel de ce qui est "évident", afin d'omettre les détails. Par ailleurs, au fur et à mesure que les ordinateurs seront davantage utilisés pour produire des démonstrations, la nécessité d'un texte se fera peut-être moins forte : une représentation logique ou graphique pourra être considérée comme tout aussi convaincante.

En ce qui concerne la compréhension de textes de démonstration, il resterait à effectuer un important travail de représentation de connaissances mathématiques avant d'envisager qu'un ordinateur puisse "lire" une démonstration. Cependant, en se limitant à des domaines simples, il est possible de concevoir des systèmes capables de comprendre un énoncé de problème donné en français et de produire une représentation logique de cet énoncé.

C'est dans le domaine de l'aide à la production et à la vérification de textes que se situent probablement les voies les plus prometteuses. Les logiciels Mentoniezh et ARRIA en constituent deux exemples. Ces systèmes représentent un compromis entre l'ouverture du logiciel (l'éventail des formes textuelles acceptées) et sa performance (la reconnaissance des phrases correctes et l'analyse des erreurs). Il serait intéressant d'étendre ce type de logiciel de manière à ce qu'il s'adapte à différents niveaux d'exigence, selon que le professeur souhaite une rédaction plus ou moins détaillée.

Remerciements

Je remercie Pascale Sébillot pour son aide précieuse lors de la rédaction du paragraphe sur le traitement du langage naturel.

Références bibliographiques

ANDERSON J.R. , BOYLE C.F. , YOST G. (1985)
The geometry tutor.
Proceedings of 9th International Joint Conference on Artificial Intelligence.
Los Angeles.

BAZIN J.M. (1993)
GEOMUS : un résolveur de problèmes de géométrie qui mobilise ses connaissances en fonction du problème posé.
Thèse de l'Université de Paris VI.

BERNAT Ph. (1994)
Conception et réalisation d'un environnement interactif d'aide à la résolution de problèmes. CHYPRE : un exemple pour la démonstration en géométrie.
Thèse de l'Université de Nancy.

BRUILLARD E. (1991)
Mathématiques et enseignement intelligemment assisté par ordinateur.

Thèse de l'Université du Maine.

COELHO H. , PEREIRA L.M. (1979)
GEOM : a Prolog geometry theorem prover.
Rapport interne n° 525, Laboratoire National d'Ingénierie Civile, Lisbonne.

DESMOULINS C. (1994)
Étude et réalisation d'un système tuteur pour la construction de figures géométriques.
Thèse de l'Université Joseph Fourier, Grenoble.

GELERNTER H. (1963)
Realization of a geometry theorem-proving machine.
Computers and thought, pp. 134-152, Feigenbaum et Feldman (Eds).

GIORGIUTTI I. , BAULAC Y. (1991)
Interaction micromonde / tuteur en géométrie.
Actes des deuxièmes journées EIAO de Cachan. École Normale Supérieure de Cachan.

LABORDE J.M. (1995)
Des connaissances abstraites aux réalités artificielles, le concept de micromonde Cabri. Environnements Interactifs d'Apprentissage avec Ordinateur.
Guin, Nicaud et Py (Eds.), Eyrolles, Paris, p. 29-42.

NEVINS A.J. (1975)
Plane geometry theorem-proving using forward chaining.
Artificial Intelligence, pp. 1-23.

NICOLAS P. (1989)
Construction et vérification de figures géométriques dans le système Mentoniezh.
Thèse de l'Université de Rennes I.

PINTADO M. (1994)
Apprentissage et démonstration automatique de théorèmes.
Thèse de l'Université de Paris VI.

Py D. (1996)
Aide à la démonstration en géométrie : le projet Mentoniezh.
Sciences et Techniques Educatives, vol.3, n° 2 , Editions Hermès.