Modélisation de la démonstration géométrique dans Geometry Tutor

D. Guin et le groupe I.A. de l'IREM de Strasbourg :
M.- A. Egret, B. Koch, G. Kuntz, G. Métivier, N. Vogel.

We study knowledge modelling of geometry proof in a well known system elaborated by J. R. Anderson and F.Boyle. This Intelligent Tutoring System is based on a cognitive theory ACT*. We present some features of Geometry Tutor which lead us to question the experimentation in a french classroom of such a system. This study is carried out with mathematical teachers (Artificial Intelligence Group, IREM Strasbourg).

I Introduction

Le Groupe Intelligence Artificielle de l'IREM de Strasbourg s'intéresse depuis plusieurs années à la modélisation des connaissances nécessaires à l'élaboration d'un environnement d'apprentissage informatisé[1] pour la démonstration géométrique (Guin et Rousselot, 1987). Après une étude des logiciels disponibles à l'époque (Guin et Groupe I.A., 1989), il était naturel que Geometry Tutor attire notre attention : c'est en effet un des tutoriels intelligents les plus fréquemment cités dans les ouvrages récents d'Intelligence Artificielle (Artificial Intelligence and Human Learning, 1988 ; Machine Learning I, 1984 ; Machine Learning II, 1986 ; Nicaud et Vivet, 1988 ; Wenger, 1987 ; Colloque Artificial Intelligence and Education, 1989 etc...), où il est souvent présenté comme un "modèle" du genre.

Après avoir succinctement présenté le fonctionnement du logiciel sur un exemple (II), nous reprendrons les grandes lignes de la théorie cognitive sous-jacente ACT* (III). En effet, Geometry Tutor a été conçu pour valider les hypothèses de ACT* ; il en résulte que celles-ci déterminent un grand nombre des caractéristiques de Geometry Tutor.

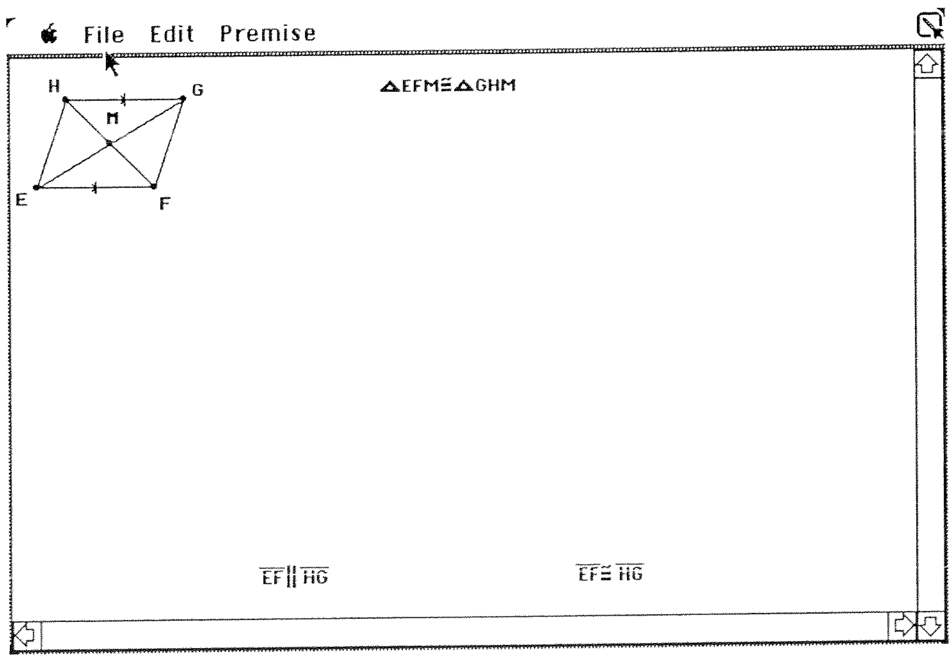
© Annales de Didactique et de Sciences Cognitives
4 (1991) (p.5- 40) - IREM de Strasbourg

[1] *Terminologie proposée dans (Balacheff, 1989)*

Après avoir présenté certains éléments de l'expertise mathématique et pédagogique (IV), nous discuterons de l'opportunité d'une éventuelle expérimentation dans une classe française d'un tel système (V).

II Présentation générale du logiciel - Version Macintosh

Prenons l'exercice classé 5.4.4. Le chapitre 5 concerne l'égalité des triangles, la notation 4.4 signifie qu'on a introduit quatre règles nouvelles depuis le début de ce chapitre et que l'exercice 5.4.4. est le quatrième utilisant la règle 4 (Règle AAS, cf. page 9). L'écran suivant [1] apparaît :

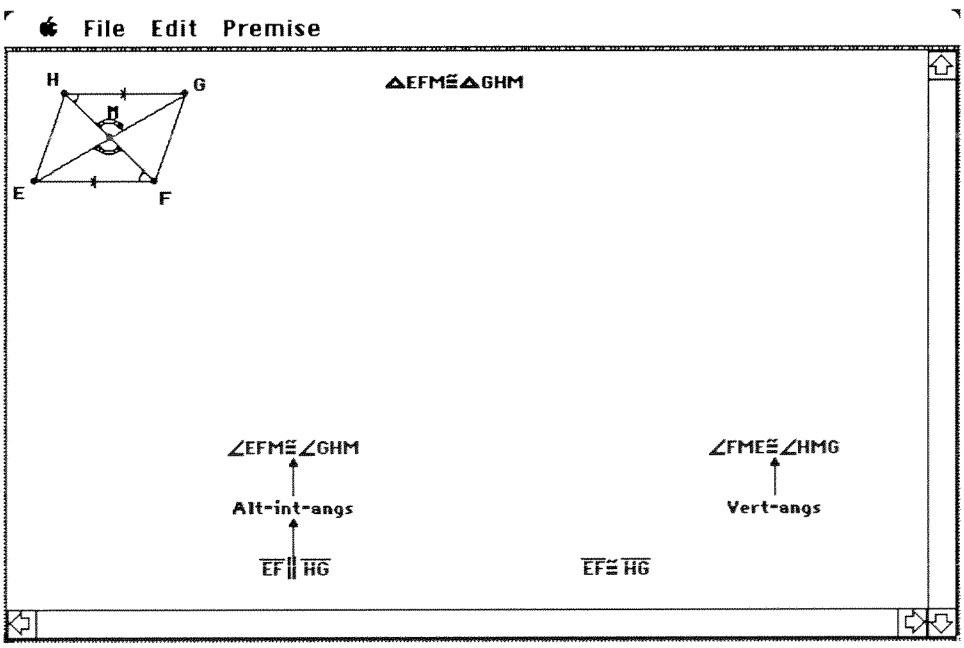


Au bas de l'écran, nous lisons les hypothèses : le segment EF est parallèle au segment HG et les segments EF et HG sont congruents. En haut de l'écran, sont présentés la figure et le but à démontrer : les triangles EFM et GHM sont égaux.

1 *Les notations utilisées par J.R. Anderson sont celles utilisées aux Etats-Unis :*
Δ UVW représente le triangle UVW
$\cong$ désigne la relation de congruence (correspondant à la superposition)
$\angle$ UVW représente l'angle géométrique UVW
le segment EF est noté $\overline{EF}$.

Nous devons maintenant construire un réseau représentant la démonstration ; pour cela, il faut procéder en plusieurs étapes *successives* suivant le même schéma : on sélectionne des prémisses, on cite la règle que l'on veut appliquer, on énonce le résultat. Chaque étape est ponctuée par un affichage à l'écran. Les prémisses peuvent bien sûr être des hypothèses ou des propriétés déjà démontrées, mais également la figure.

Appliquons par exemple le théorème des angles alternes-internes (alt-int-angs) puis le théorème des angles opposés (vert-angs). Ce dernier théorème a une seule prémisse, la figure. L'écran se présente alors ainsi :



Remarquons que les résultats démontrés sont codés au fur et à mesure sur la figure.

Sélectionnons alors comme prémisses les trois congruences et proposons la règle ASA (triangles ayant un côté compris entre deux angles respectivement égaux). Cette règle est refusée.

A la troisième tentative, nous obtenons le message :

The easiest way to figure out what rule to use is to name the **type** of premises. Side-Side-Side is **SSS,** Side-Angle-Side is **SAS,** Angle-Side-Angle is **ASA,** and Angle-Angle-Side is **AAS.** Notice that in this case the side is **NOT between** the angles! **OK**	La meilleure façon de trouver la règle applicable est de nommer les **types** de prémisses : Côté - Côté - Côté est **SSS** Côté - Angle - Côté est **SAS** Angle - Côté - Angle est **ASA** Angle - Angle - Côté est **AAS** Remarquez que dans ce cas le côté **n'est pas entre** les angles ! **OK**

Essayons d'abandonner les prémisses précédentes dans l'intention de prouver que les angles $\widehat{GHM}$ et $\widehat{EFM}$ sont égaux.

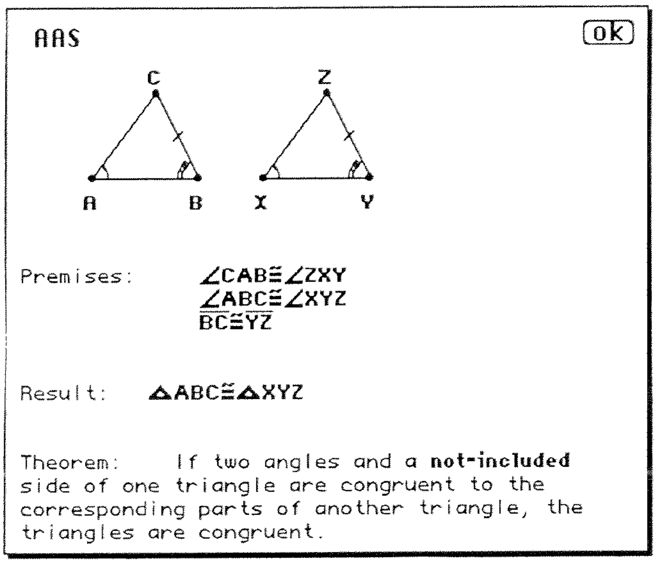
On nous indique :

The premises you have selected are fine. I strongly recommend that you go on to the result **OK**	Les prémisses que vous avez sélectionnées sont bonnes. Je vous recommande vivement de continuer votre raisonnement. **OK**

Demandons alors les règles que l'on peut utiliser,

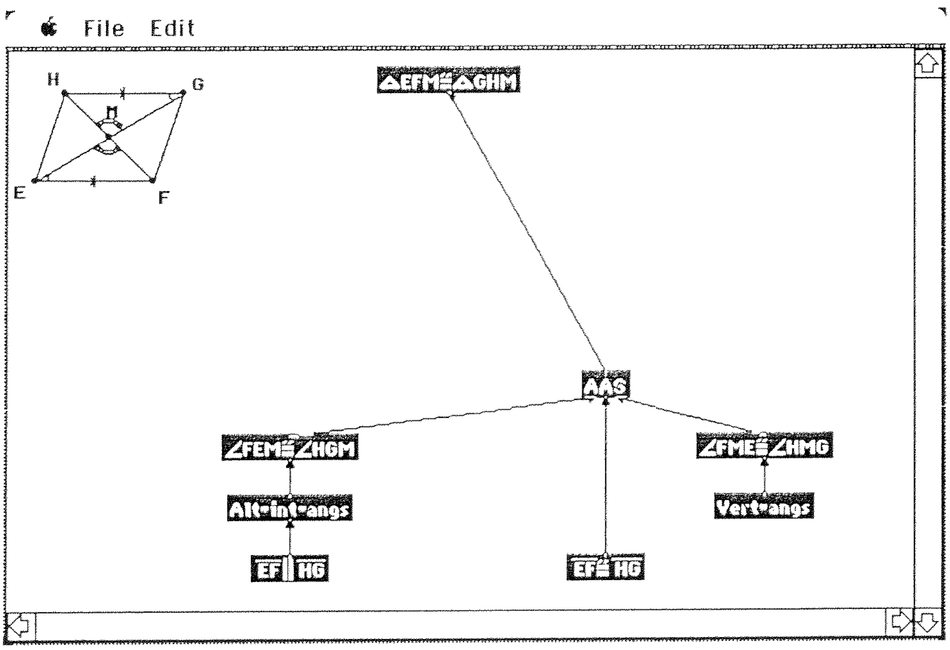
Applicable rules **OK** **AAS**	Règles applicables **OK** **AAS**

et l'énoncé de la règle AAS :



Théorème : Si deux angles d'un triangle et un côté **non compris** entre ces deux angles sont congrus aux éléments correspondants d'un autre triangle, les triangles sont congruents

Appliquons donc la règle AAS et énonçons le résultat correspondant au but à démontrer. L'exercice est terminé.



Sur cet exemple simple, nous pouvons remarquer qu'en essayant d'utiliser au départ la règle "quad-sides-cong-par" (si les segments AB et CD sont parallèles et congruents alors ABCD est un parallélogramme), nous obtenons le message suivant :

Quad-sides-cong-par is not allowed to be used in this problem. **OK**	Il est interdit d'utiliser **Quad-sides-cong-par** dans cet exercice. **OK**

Ceci nous suggère déjà deux questions, que nous développerons plus loin :

— Pourquoi dessiner alors les segments HE et GF sur la figure ?
— Bien que le chapitre 5 précède le chapitre 7 des parallélogrammes, ne serait-il pas plus intéressant de permettre à un élève de revenir aux exercices du chapitre 5 avec les acquis du chapitre 7, d'autant plus que les auteurs du logiciel souhaitent que ce dernier puisse aussi servir pour des exercices de remédiation (manuel de l'utilisateur de Geometry Tutor) ?

Pour répondre à ces questions, il nous faut tout d'abord examiner la théorie cognitive sous-jacente à l'élaboration du logiciel.

III Le modèle cognitif ACT*

J.R. Anderson a travaillé durant de nombreuses années à une théorie générale de la cognition ACT (Adaptive Control of Thought). Après l'élaboration d'une version plus récente notée ACT*, son équipe s'est tournée vers la construction de tutoriels intelligents simultanément pour appliquer cette théorie et valider ses hypothèses. Le projet ACTP (the Advanced Computer Tutoring Project) interdisciplinaire regroupe des psychologues et des informaticiens, et repose donc sur le modèle cognitif de ACT*. Voici très brièvement les principes de cette théorie qui sont exposés en détails dans (Anderson et alii, 1987-2) :

Principe 1 : Représenter le modèle (ou comportement) de l'élève comme un ensemble de règles de productions [1] .
Principe 2 : Mettre en évidence la structure hiérarchique de buts sous-jacente à la résolution de problèmes.

[1] *Expressions de la forme : Si condition alors action.*

Principe 3 : Enseigner dans un contexte de résolution de problèmes (les compétences sont acquises par l'action).
Principe 4 : Favoriser un niveau abstrait de représentation de connaissance pour la résolution de problèmes (car il est mieux adapté à une bonne organisation de cette résolution).
Principe 5 : Minimiser le travail de la mémoire (afin d'éviter les erreurs dues à une perte de celle-ci).
Principe 6 : Réagir immédiatement aux erreurs.
Principe 7 : Adapter l'enseignement à l'expertise de l'élève.
Principe 8 : Faciliter la résolution par approximations successives.

Les deux hypothèses fondamentales de ce modèle sont les suivantes :

— les fonctions cognitives peuvent être représentées par des *règles de production* (principe 1),

— le mécanisme d'apprentissage, appelé *compilation* de la connaissance, se réduit à la transformation de la connaissance *déclarative* [1] enseignée en une connaissance *procédurale* et son organisation (Anderson,1986 et 1987-1). Cette compilation comporte deux formes :

— la *procéduralisation* qui transforme la connaissance en une règle de production applicable à un domaine spécifique,

— la *composition des règles* qui permet de définir une nouvelle règle en combinant les effets de plusieurs règles.

exemple (Anderson, 1984) : Considérons la connaissance déclarative relative au cas d'égalité des triangles appelé SAS (1 angle égal compris entre deux côtés égaux), J. Anderson la représente sous forme d'un *schéma* comportant le contexte (s1 est un côté du triangle XYZ, etc..), les hypothèses et la conclusion .

[1] *Exemple donné par J. Pitrat : "L'article s'accorde en genre et en nombre avec le nom" est une connaissance déclarative. "Quand on rencontre un article, on cherche le nom auquel il se rattache. On vérifie que le genre du nom est le même que celui de l'article, puis que le nombre du nom est identique à celui de l'article" est une connaissance procédurale.*

Considérons la règle générale (indépendante du domaine de la géométrie) :

Si le but est de démontrer une assertion
et qu'il existe un schéma ayant cette assertion comme conclusion
Alors prendre comme sous-buts :
identifier le contexte du schéma
démontrer les hypothèses du schéma

En appliquant cette règle générale, nous transformons la connaissance déclarative en une règle utilisable en inférence arrière :

Si le but est de démontrer Δ UVW $\cong \Delta$ XYZ [1]
Alors prendre comme sous-buts :
1. Montrer que les segments UV et XY sont congruents
2. Montrer que les segments VW et YZ sont congruents
3. Montrer que $\angle$ UVW $\cong \angle$ XYZ

Remarque : En appliquant une autre règle générale nous pouvons transformer la *même connaissance déclarative* en une règle utilisable en inférence avant :

Si le but est de déduire une assertion du fait que les segments UV et XY sont congruents
et que $\angle$ UVW $\cong \angle$ XYZ et que les segments VW et YZ sont congruents
Alors en déduire que Δ UVW $\cong \Delta$ XYZ

J.R. Anderson se place dans un contexte de *résolution de problèmes* (principe 3) : l'élève n'apprend efficacement que lorsqu'il agit. La compilation de la connaissance dépend étroitement de la structure hiérarchique des buts (principe 2) inhérente à la résolution de problème. Les tâches sont vues comme des *espaces-problèmes*. L'espace-problème (Newell et Simon, 1972) est une construction mentale composée d'un ensemble d'états et d'opérateurs permettant de passer d'un état à l'autre (ce sont les règles de production). Un problème est défini par la donnée d'un état initial et d'un état final. Une solution est alors un chemin dans cet espace entre l'état initial et l'état final déterminé par une suite d'opérateurs.

[1] *Pour les notations utilisées par J.R. Anderson voir page 6*

L'apprentissage des compétences est structuré : chaque cours fait intervenir un nouvel ensemble de règles primitives, qui devraient correspondre aux unités conceptuelles que les débutants peuvent acquérir. Le niveau d'expertise peut alors être défini en fonction de la compétence à combiner ces règles primitives. L'expertise est définie en termes d'acquisition de *compétences* (ou savoir-faire).

Un principe important de ACT* est la réaction immédiate aux "erreurs" (principe 6). Il n'est pas possible de s'éloigner trop d'un chemin de preuve, même si les règles appliquées sont correctes. C'est le paradigme de la modélisation par la trace *(model tracing)* : l'analyse du comportement de l'élève se fait par comparaison entre le modèle idéal et le modèle erroné . "IBR" (Ideal Buggy Rules) peut tracer le chemin solution de l'élève en identifiant la règle qu'il applique à une des règles du modèle idéal ou erroné. Celà simplifie la tâche de diagnostic des erreurs, et l'on peut donner ainsi une aide adaptée (Anderson et alii, 1985 ; Reiser et alii, 1985). Ce paradigme repose sur la compilation de la connaissance qui établit une étroite *correspondance* entre les unités du modèle interne du système (Wenger, 1987, p. 378) et les comportements observables de l'élève : ceci à la différence d'un concept qui doit être interprété *avant* qu'il ne se manifeste dans un comportement.

Enfin la théorie ACT* prétend modéliser l'acquisition des compétences *indépendamment* de la richesse du domaine et s'appliquer dans une *grande variété* de domaines. Il en résulte que l'apprentissage est un processus *relativement simple* (Anderson, 1989-1 et 2) si l'on analyse la tâche en terme de règles de production : l'essentiel est donc de mettre en évidence une décomposition appropriée de la tâche en unités de connaissance distinctes qui sont des règles de production spécifiques au domaine. Une fois la connaissance déclarative incorporée dans ces unités de connaissance, elle peut être éliminée (Anderson, 1987-1). L'acquisition de savoir-faire est alors obtenue en composant les fonctions simples d'apprentissage pour ces unités. La *modularité* de ces unités permet à l'élève de les acquérir indépendamment les unes des autres (Anderson, 1989-1). La règle de production est donc par conséquent la bonne unité pour analyser l'acquisition des compétences, d'où le paradigme de la modélisation par la trace. De plus, selon J.R. Anderson, le transfert des règles générales à travers les contenus et les langages confirme l'hypothèse que le processus d'apprentissage est simple.

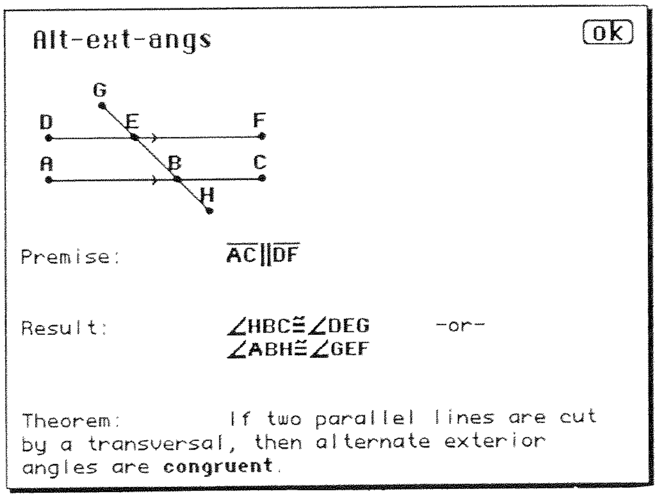
IV Sur l'expertise mathématique et pédagogique de Geometry Tutor

Avant d'aborder l'étude de l'expertise mathématique et pédagogique de Geometry Tutor, il faut tout d'abord préciser que l'activité exigée par ce tutoriel de la part de l'élève comporte simultanément une tâche heuristique (choix de théorèmes) et la construction d'une représentation de l'*organisation déductive* [1] : un graphe de preuve s'appuyant sur la structure ternaire (prémisses, règle, résultat).

Remarquons une possibilité intéressante du logiciel : Geometry Tutor autorise plusieurs solutions pour résoudre un exercice. Cependant, la règle proposée doit appartenir aux règles prévues pour résoudre cet exercice et la solution proposée ne doit pas s'*éloigner* de plus d'un *pas* de déduction du graphe prévu. Or, dans la plupart des cas, un pas d'éloignement ne suffit pas pour proposer une autre solution correcte.

a) Base de connaissances

Il n'existe pas, dans Geometry Tutor, certains types de relations ou certains objets, pourtant usuels en mathématiques : nous avons déjà évoqué les difficultés dues à l'absence de relations non binaires. Nous regrettons aussi l'absence de droites. Toute règle concernant des droites est rapportée à une règle sur des segments. Par exemple " alt-ext-angs" :



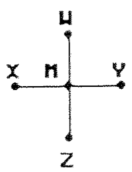
Théorème : Si deux droites parallèles sont coupées par une sécante, alors les angles alternes - externes sont congruents.

[1] *Notre conception de l'expertise est détaillée page 35.*

Cette restriction est d'autant plus gênante qu'il est impossible, dans un exercice, d'utiliser des points non donnés dans l'énoncé. Deux droites ne formeront donc des angles alternes-externes que si l'énoncé cite des points de la sécante situés à l'extérieur de la bande comprise entre les deux parallèles!

Le statut des règles est mal défini et leur présentation n'est pas homogène. Elles contiennent à la fois la définition d'un objet, un énoncé direct permettant de démontrer qu'un objet répond à la définition et un énoncé réciproque permettant de prouver les propriétés spécifiques de l'objet défini. Ces trois énoncés distincts portent le même nom. Examinons, par exemple la règle "def-perp-lines" :

Def-perp-lines ok



Premise: $\overline{XY} \perp \overline{ZW}$
Result: rt $\angle XMW$ -or-
rt $\angle XMZ$ -or-
rt $\angle YMZ$ -or-
rt $\angle YMW$
--or--
Premise: rt $\angle XMW$ -or-
rt $\angle XMZ$ -or-
rt $\angle YMZ$ -or-
rt $\angle YMW$
Result: $\overline{XY} \perp \overline{ZW}$

Definition: Two lines are perpendicular if and only if the angles they form are **right angles**.

Définition : Deux droites sont perpendiculaires si et seulement si les angles qu'elles forment sont des angles droits

Dans cette définition, on trouve trois énoncés :

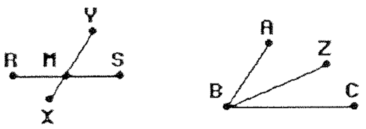
— le premier pour pouvoir démontrer que deux segments sont perpendiculaires,
— une "réciproque",
— une définition en langue naturelle.

Comme dans un grand nombre d'énoncés, nous trouvons plusieurs assertions séparées par des "ou". Le statut des "ou" est différent dans les prémisses et le résultat, et ne correspond ni au point de vue logique (les "ou" du résultat devraient alors être remplacés par des "et"), ni à un point de vue pédagogique (qui préciserait que si l'un des quatre angles est droit, les trois autres le sont aussi). La règle "def-perp-lines" contient cinq formulations différentes ayant comme conclusion "deux droites sont perpendiculaires" : l'expression en langue naturelle a comme prémisses plusieurs angles droits alors que les autres formulations n'en ont qu'une seule. Nous retrouvons cette situation dans la plupart des définitions ; il est donc très difficile pour l'utilisateur peu averti qui a effectivement besoin de consulter la base d'en tirer profit.

De nombreux énoncés de règles sont illisibles par des élèves, c'est-à-dire qu'il est difficile d'y repérer les prémisses ou le résultat obtenu lors de l'application de la règle. La procéduralisation (cf. page 11) est alors peu explicitée. Ceci,

— soit à cause de la présentation (dans la règle "def-bisector[1] " que sont les points X, Y, M, R et S ?)

Def-bisector ok



Premise: $\overline{XY}$ **bisects** $\overline{RS}$
Result: $m\overline{RM} = m\overline{MS}$ -or- $\overline{RM} \cong \overline{MS}$

--or--

Premise: $m\overline{RM} = m\overline{MS}$ -or- $\overline{RM} \cong \overline{MS}$
Result: $\overline{XY}$ **bisects** $\overline{RS}$

Definition: Segment $\overline{XY}$ bisects segment $\overline{RS}$ if and only if the lengths of segments $\overline{RM}$ and $\overline{MS}$ are **equal** or segments $\overline{RM}$ and $\overline{MS}$ are **congruent**.

----------- OR -----------

Premise: $\overline{BZ}$ **bisects** $\angle ABC$
Result: $m\angle ABZ = m\angle ZBC$ -or- $\angle ABZ \cong \angle ZBC$

--or--

Premise: $m\angle ABZ = m\angle ZBC$ -or- $\angle ABZ \cong \angle ZBC$
Result: $\overline{BZ}$ **bisects** $\angle ABC$

Definition: Segment $\overline{BZ}$ bisects angle **ABC** if and only if the measures of angles **ABZ** and **ZBC** are **equal** or angles **ABZ** and **ZBC** are **congruent**.

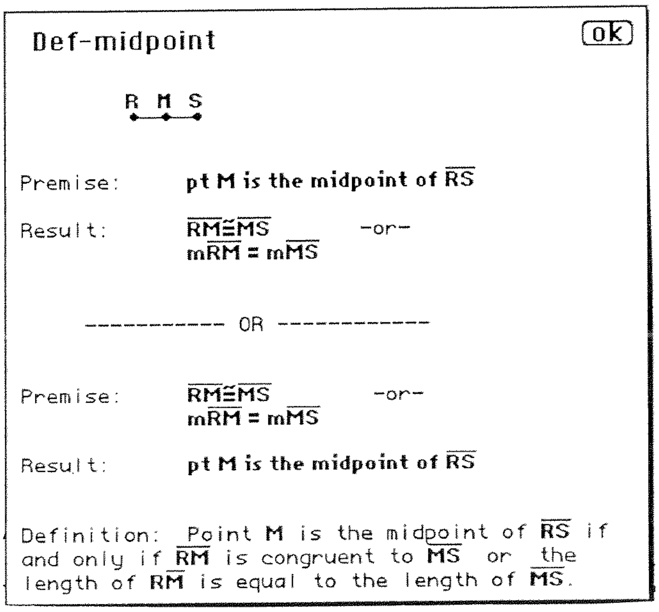
Définition : Le segment XY coupe le segment RS en son milieu si et seulement si les longueurs des segments RM et MS sont égales ou si les segments RM et MS sont congruents.

Définition : Le segment BZ est la bissectrice de l'angle ABC si et seulement si les mesures des angles ABZ et ZBC sont égales ou si les angles ABZ et ZBC sont congruents.

— soit parce qu'il y a confusion entre théorème direct et théorème réciproque dans l'application des règles. Or, dans l'enseignement, il faut se battre constamment contre la

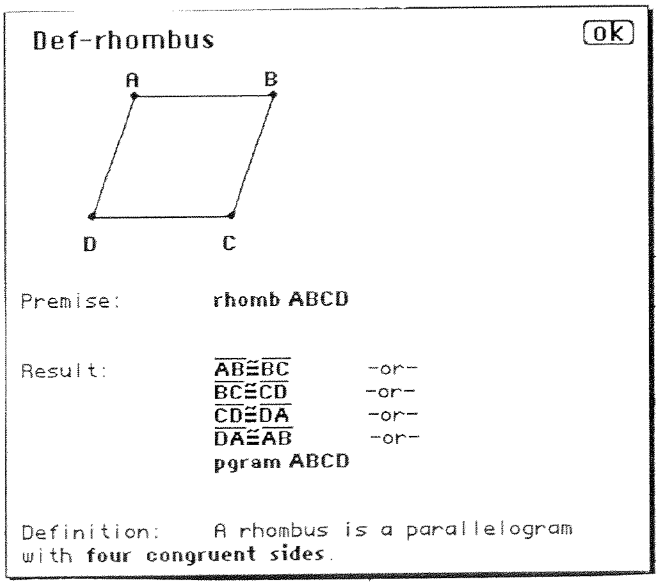
[1] *Remarque: "Def - bisector" s'applique à deux objets différents: un segment et un angle. Bisector représente pour un angle la bissectrice, pour un segment, une notion non utilisée en France: une sécante à un segment en son milieu.*

tendance naturelle des élèves à faire cette confusion. Il nous paraît *indispensable* de distinguer nettement théorème direct et théorème réciproque. Les auteurs du logiciel en sont d'ailleurs quelquefois conscients : par exemple, dans la règle "def-midpoint", il y a un énoncé direct et un énoncé réciproque. Ils sont séparés par un “OR” (en majuscules)!



Définition : Le point M est le milieu de RS si et seulement si RM est congru à MS ou la longueur de RM est égale à la longueur de MS.

Par contre, dans la règle "def-rhombus" seul l'énoncé direct est présenté :



Définition : Un losange est un parallélogramme avec quatre côtés congruents.

Avec une telle définition, on peut se demander si un quadrilatère ayant quatre côtés égaux est un losange! Cette définition est une exception parmi les règles proposées car les cinq propriétés séparées par des "ou" ne sont pas de même nature. De plus, elle ne devrait fournir aucun moyen de démontrer qu'une figure est un losange, *pourtant* le logiciel accepte l'utilisation de "def-rhombus" pour démontrer qu'un parallélogramme ayant deux côtés consécutifs égaux est un losange : il en est ainsi, par exemple, dans l'exercice 8.4.4 !

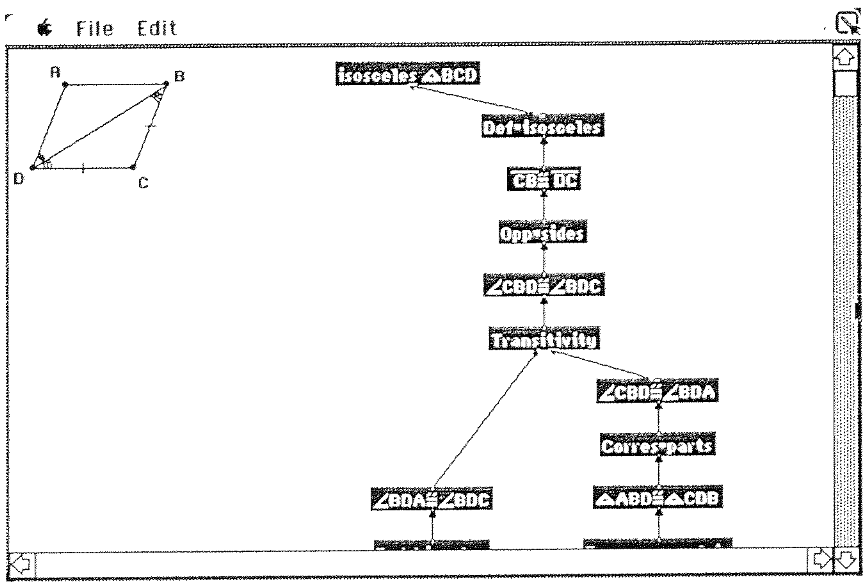
La base est souvent *incomplète*, alors que certaines règles ne semblent y figurer qu'à cause du fonctionnement du logiciel.

— soit l'élève n'a connaissance que d'*une seule* propriété caractéristique d'un objet : dans l'exercice 8.6.2, il faut prouver qu'un quadrilatère est un carré. Lorsqu'on propose les prémisses :

AL congru à LK

ALKJ est un parallélogramme

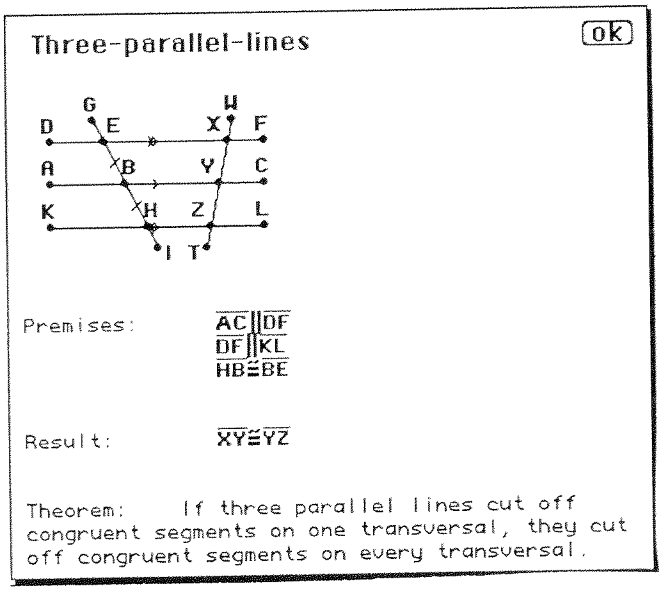
pour appliquer la règle "def-rhombus", on obtient le message d'aide (?) " Def-rhombus ne peut être utilisée ici ". Cet essai est rejeté parce que dans la base de connaissances, un carré est un rectangle ayant deux côtés consécutifs égaux mais pas un losange ayant un angle droit ! De même, un triangle isocèle est un triangle ayant deux côtés égaux mais pas un triangle ayant deux angles égaux. (exercice 7.4.1)



— soit certaines règles *font défaut*, alors que leur réciproque figure dans la base de connaissances. Ainsi "diags-perp-rhomb" permet d'établir que les diagonales d'un losange sont perpendiculaires mais il n'y a pas de règle pour démontrer qu'un parallélogramme dont les diagonales sont perpendiculaires est un losange.

— soit certaines règles n'ont, à notre avis, aucun intérêt pédagogique. C'est le cas des règles "réflexivité" ou "def-congruence" qui ne sont pas utilisées en France dans l'apprentissage de la démonstration en géométrie. Il est nécessaire d'expliciter toutes les règles au cours de la résolution des exercices, ce qui agace rapidement l'utilisateur : le niveau d'exigence semble artificiel. Il paraît peu vraisemblable que l'exigence d'une règle comme la règle "réflexivité" soit un choix didactique.

D'ailleurs, l'on constate rapidement une *incohérence* entre le niveau de rigueur exigé de la part de l'utilisateur et le manque de rigueur du logiciel. Par exemple, la règle "transitivité" est exigée lorsque l'on veut prouver que deux droites parallèles à une même troisième sont parallèles entre elles alors que les auteurs du logiciel rendent la règle précédente implicite dans la définition de "three-perp-lines".



Théorème : Si trois droites parallèles découpent des segments congruents sur une sécante, elles découpent des segments congruents sur toute sécante.

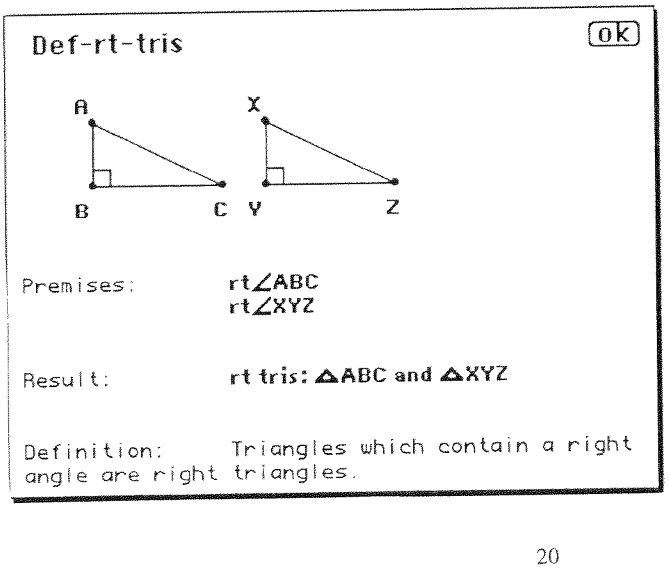
Dans cet énoncé, il est question de trois droites parallèles alors que l'on ne peut utiliser que des relations binaires dans le logiciel. C'est pourquoi, le parallélisme des trois droites est décomposé en deux relations binaires dans les prémisses. Cependant, l'énoncé précédent n'est pas équivalent à :

Prémisses : AC // DF
AC // KL
HB = BE

Résultat : XY = YZ

où c'est le segment AC, cité deux fois dans les prémisses, qui contient les milieux B et Y. Pourtant, il est possible d'appliquer la règle "three-perp-lines" sous cette forme sans utiliser au préalable la transitivité du parallélisme.

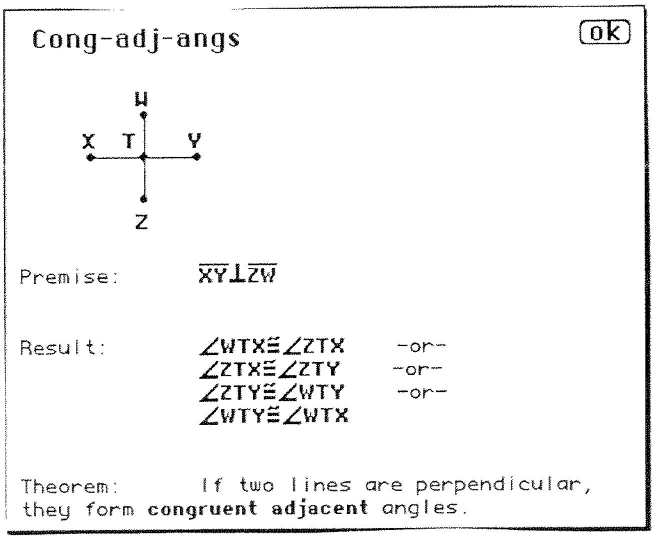
Les objets définis *n'héritent pas des propriétés* des objets parents. Il est impossible de déduire directement qu'un carré est un parallélogramme : il faut auparavant utiliser la règle "def-rectangle". Quelle peut être alors l'*évolution des conceptions* des élèves quand leur expertise augmente ? En particulier, on ne peut pas utiliser des propriétés équivalentes, ce qui est pourtant essentiel en mathématiques : par exemple, il n'est pas possible de déduire immédiatement que si un triangle ABC est un triangle rectangle en B alors les droites AB et BC sont perpendiculaires. Il sera nécessaire de faire appel à deux règles dont l'une, la définition des triangles rectangles, est très amusante :



Définition : Des triangles qui ont un angle droit sont rectangles

La figure et l'énoncé sous forme prémisses-résultat montrent que le pluriel n'est pas un lapsus, mais que son usage est délibéré. Pour qu'un triangle soit rectangle, il semble donc *indispensable* qu'il ait un jumeau (isométrique) nanti, lui aussi un angle droit. La raison évoquée dans le manuel de référence (page 46) est "le nombre de prémisses des cas d'égalité des triangles rectangles est ainsi le même que celui des cas d'égalité des triangles quelconques".

Les figures dessinées dans les règles sont très souvent des *figures particulières* : il est pourtant essentiel de varier les représentations et de ne surtout pas fixer des images de cas particuliers (cong-adj-angs).



Théorème : Si deux droites sont perpendiculaires, elles forment des angles adjacents congruents.

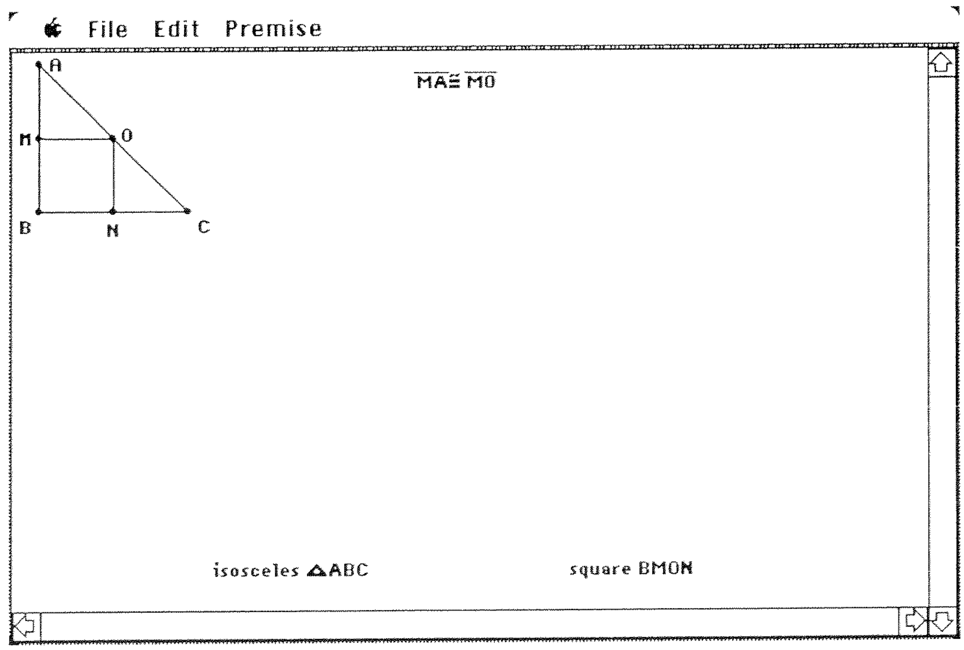
b) Présentation des exercices

Lors de l'exécution d'un exercice, la figure, les hypothèses en bas de l'écran et la conclusion en haut apparaissent tout d'abord. Trois opérations successives (choisir les prémisses, donner la règle d'application, proposer le résultat) sont demandées. Il n'y a donc pas d'erreur possible sur les *statuts* des énoncés. Or, pour apprendre ce qu'est le raisonnement déductif, il est nécessaire que l'élève comprenne que les énoncés sont organisés non en fonction de leur contenu, mais en fonction de leur *statut opératoire* (Duval, 1989 [1]). Il est donc important qu'il reconnaisse seul les différents statuts opératoires des énoncés.

[1] *Voir page 26 de l'article cité.*

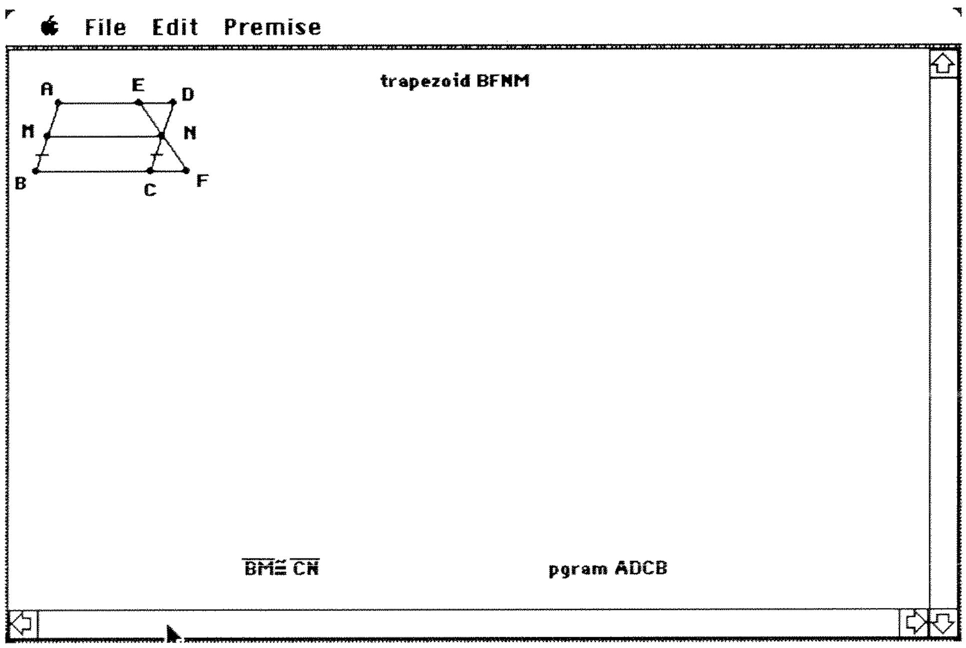
Nous apprécions la possibilité de choisir la figure comme prémisse, mais il n'en reste malheureusement, dans la suite de l'exercice, aucune trace sur l'écran : cette idée pourrait être exploitée au-delà de l'application d'une règle, puisque l'on constate qu'au cours d'une démonstration, on utilise fréquemment des *figures extraites*, en particulier pour élaborer un plan (Guin, 1989) : les élèves leur donnent alors un statut qu'ils appellent " ce que je vois" (Egret, 1989).

Il faut aussi noter que certains énoncés occultent le problème d'*existence* de la figure parce qu'elle est déjà construite. Par exemple, dans l'énoncé de l'exercice 8.6.3., les hypothèses sont les suivantes : ABC est un triangle isocèle en B et BMON un carré. De plus, comme semble le suggérer la figure, les points (B, M, A), (B, C, N) et (A, O, C) sont alignés ; est-il évident que la construction est possible ? M est-il quelconque (pour une fois, ce n'est pas par hasard que M se situe au milieu du segment AB) ?

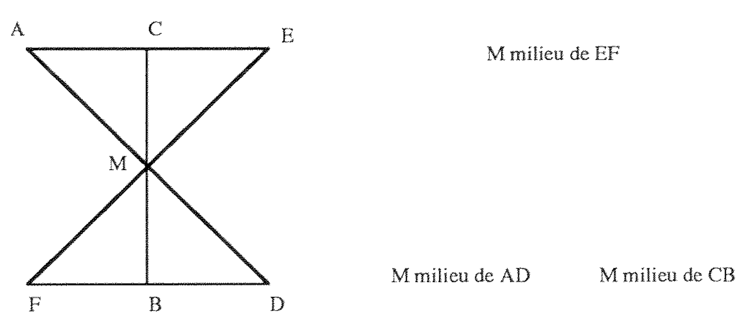


Les énoncés sont souvent *incomplets* et doivent être *lus sur la figure* : dans l'apprentissage de la démonstration, ne s'agit-il pas justement d'arriver à ce que les élèves prennent conscience que l'organisation déductive porte sur des *énoncés* et non sur des propriétés lues sur la figure ?

Alors comment peuvent-ils acquérir cette distinction s'ils doivent lire sur la figure que les points (B, M, A) puis (B, C, F) et enfin (C, N, D) sont alignés, mais démontrer sans " voir " que les droites BF et MN sont parallèles ? (exercice 8.7.3)



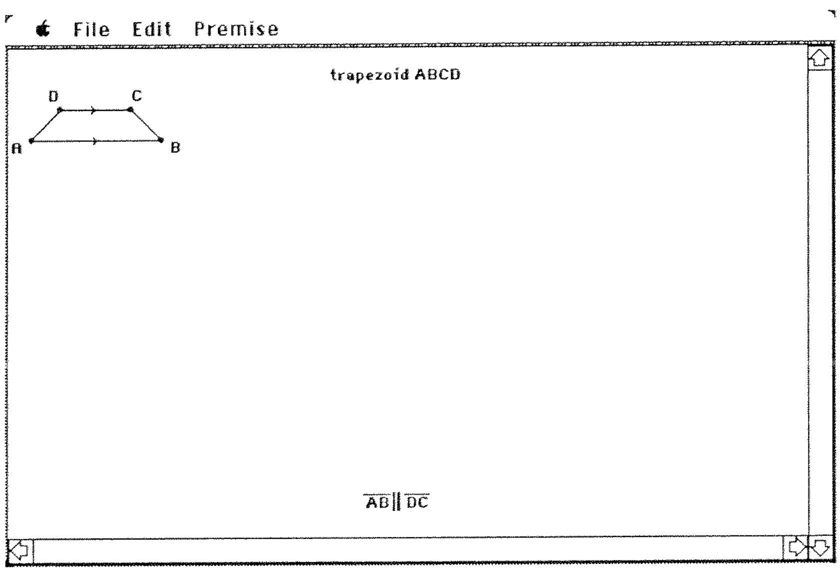
Remarquons que ce n'est pas une particularité de la version Macintosh du logiciel, et citons un exemple donné dans (Anderson et alii 1985) :



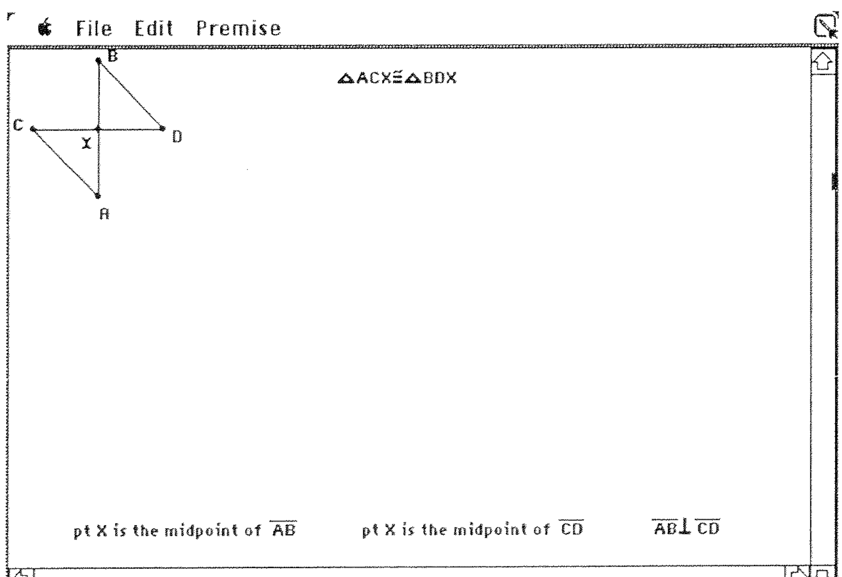
Comme c'est le cas dans la base de connaissances, les figures construites sont trop souvent des *figures particulières*.

Il faut insister sur la nécessité de varier les représentations que les élèves ont des objets géométriques :

— un trapèze est-il toujours isocèle (exercice 8.7.1) ?



— les extrémités de deux segments perpendiculaires se coupant en leur milieu sont-elles toujours les sommets d'un carré (exercice 6.6.3) ?

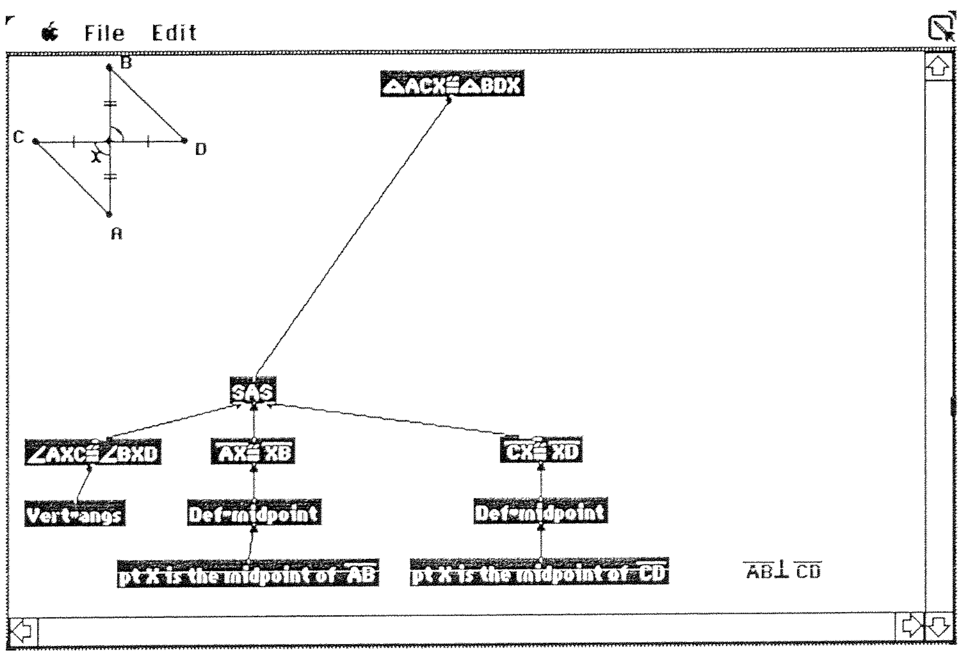


— un point sur un segment en est-il toujours le milieu (cf. les points M et N de la figure page 23) ?

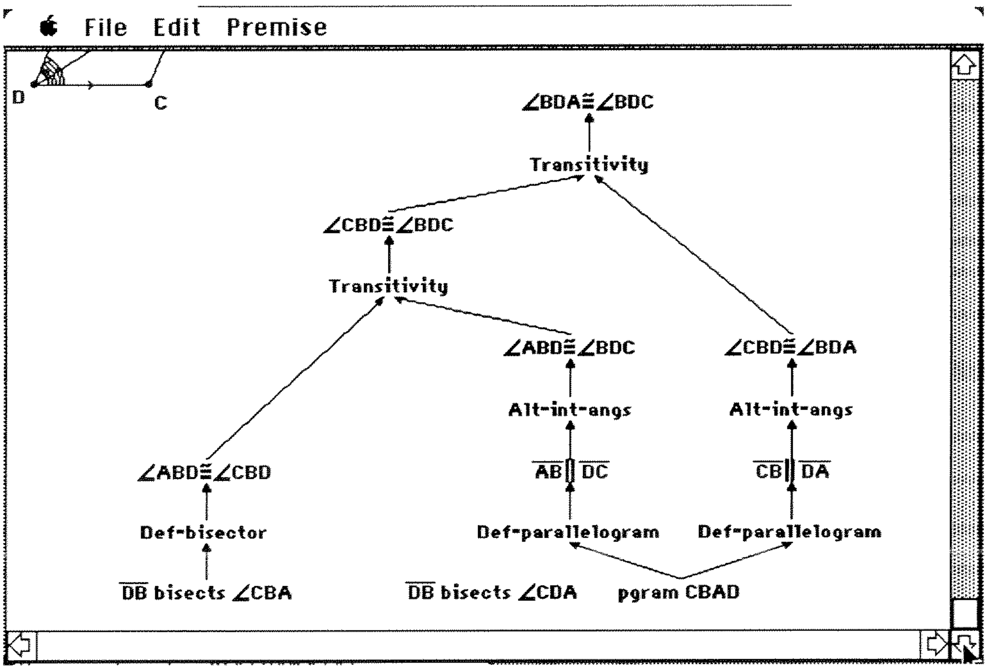
Lorsque l'on se rappelle que tous les *alignements* doivent obligatoirement être *lus sur la figure*, peut-on vraiment demander à un élève de collège de comprendre l'activité qui lui est demandée? Et en particulier comment apprendra-t-il à prouver un alignement de points? Il semble que cette préoccupation a échappé aux auteurs du logiciel puisque dans "def-midpoint" (cf. page 17) la prémisse d'alignement des points R, M, S n'est même pas citée.

Une dernière remarque au sujet des exercices : ils comportent parfois des *hypothèses superflues* :

— soit l'énoncé est un cas particulier d'un cas plus général comme dans l'exercice 6.6.3 où une des hypothèses n'est pas utilisée pour résoudre l'exercice,



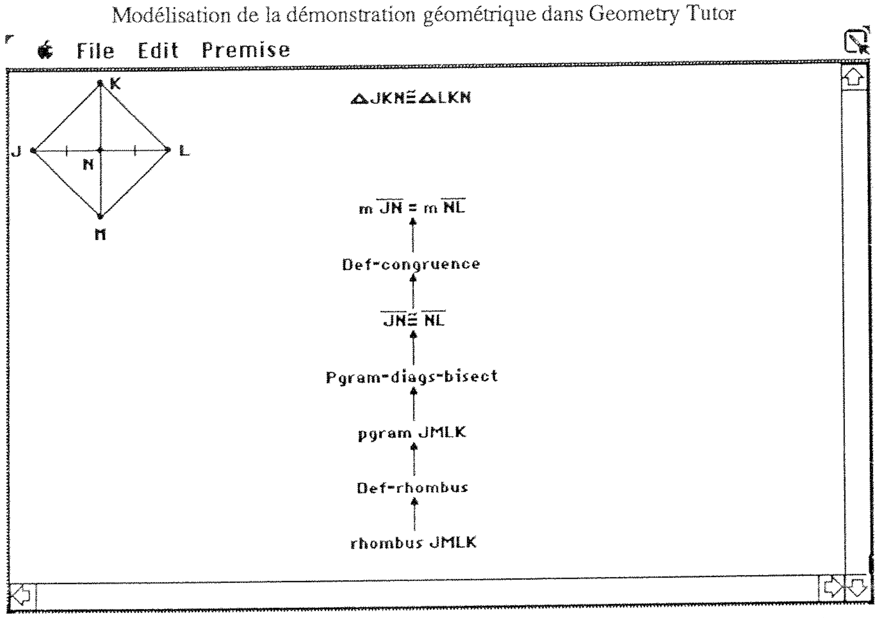
— soit l'une des hypothèses peut être démontrée en utilisant les autres (dans l'exercice 7.4.1, il est possible de démontrer que DB est la bissectrice de l'angle CBA en utilisant les deux autres prémisses.)



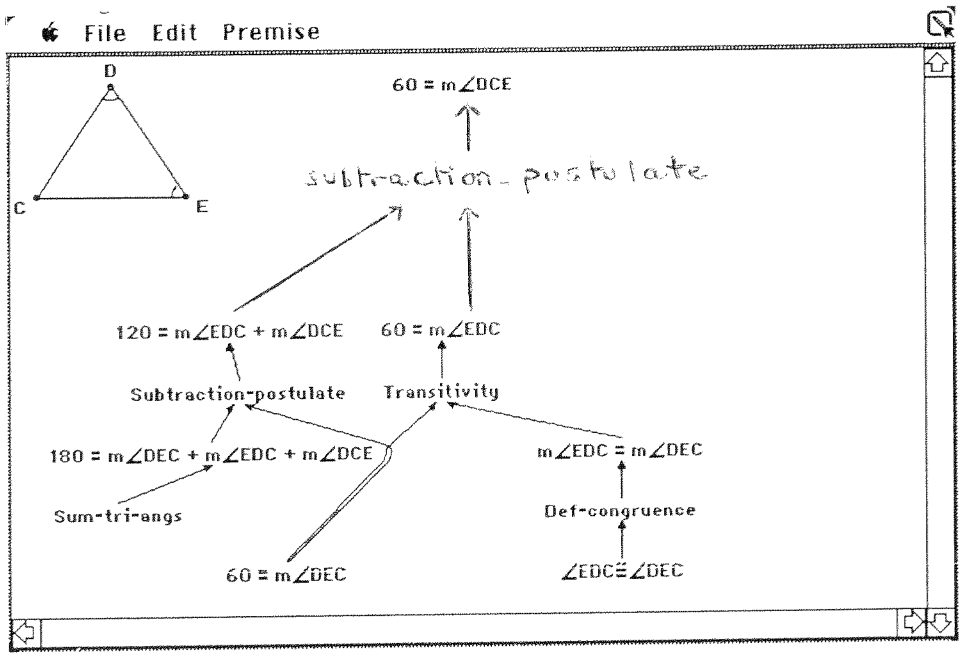
— soit l'un des objets est superflu (quel est le rôle du point E dans l'exercice 8.7.3 de la page 23 ?).

c) Application des règles :

Il arrive que le logiciel donne plusieurs formes équivalentes d'une même définition, mais celles-ci ne le sont plus dans les exercices : par exemple la congruence de deux segments RM et MS est équivalente à l'égalité des longueurs des segments RM et MS (def-midpoint, cf. page 17), mais il est impossible d'affirmer que puisque les diagonales du parallélogramme JKLM se coupent en leur milieu N, les longueurs JN et NL sont égales. Seule la congruence des segments JN et NL est acceptée. (exercice 8.5.2 et def-midpoint citée ci-dessus) :



Geometry Tutor refuse certaines *applications correctes* de règles. Dans l'exercice 4.2.3. la règle "substraction-postulate" permet de démontrer rapidement le résultat demandé, comme le montre le diagramme complété ci-dessous.

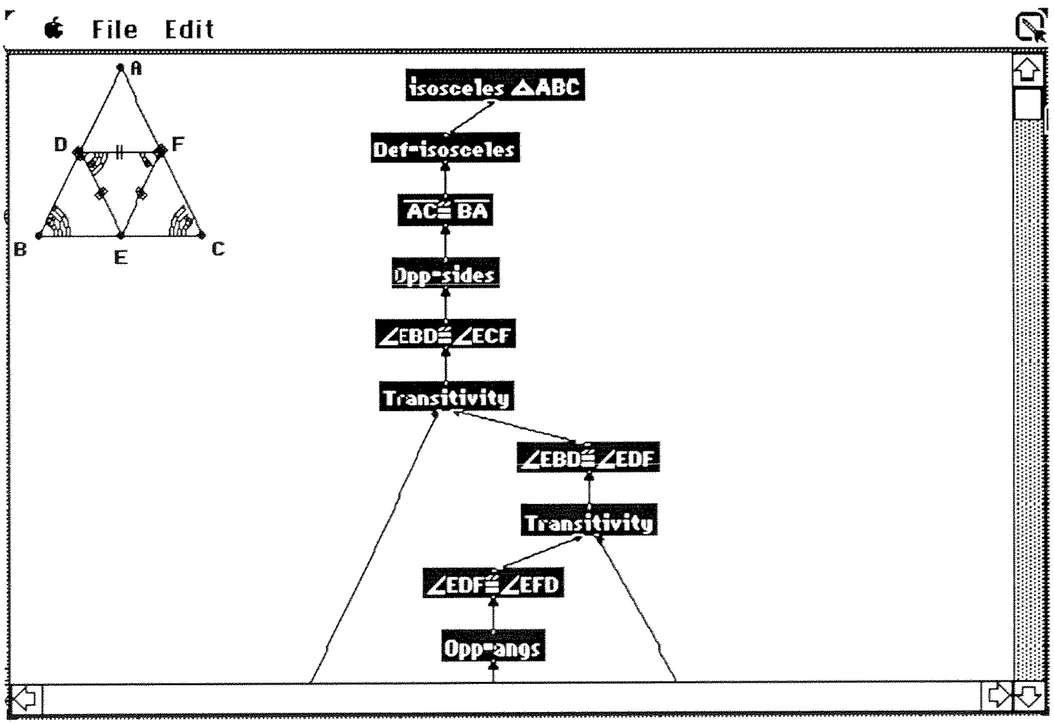


Cette solution parfaitement correcte est refusée : ce chemin n'a pas été retenu parmi les démonstrations correctes, il est donc rejeté parce qu'il s'éloigne de plus d'un pas du modèle IBR.[1] On peut remarquer que dans le même exemple, la règle substraction-postulate a pu être appliquée avec une seule des prémisses.

d) Aides :

Les aides proposées sont diverses, en voici quelques-unes :

— le *codage* : les propriétés sont codées au fur et à mesure de la démonstration, ce qui est une aide efficace pour les élèves. Mais pourquoi le codage de plusieurs segments égaux n'est-il pas le même? La règle de transitivité, si souvent demandée par ailleurs, n'est plus traduite sur la figure. Dans l'exercice 8.5.4, on peut remarquer que les angles et les segments qui ont même mesure ne sont pas codés en conséquence. Le défaut de codage rend alors la résolution de l'exercice nettement plus difficile.



[1] *Cf. page 13*

— des aides sous forme de questions ou d'affirmations qui sont des aides uniquement *locales* (il n'y a pas d'heuristique plus générale que celle concernant l'application d'une règle) et qui ne sont pas adaptées au travail fait par l'élève. En particulier, la démarche de l'élève n'est pas interprétée.

exemples : Quelle sorte de quadrilatère a **deux côtés parallèles ?**

Quelle **définition** contient le mot "bisector" ?

Si deux triangles sont congruents, vous pouvez prouver que leurs côtés correspondants sont congruents.

— s'il fait deux "erreurs" consécutives, le pas de déduction est fourni à l'élève, ce qui permet de résoudre entièrement l'exercice sans aucune réflexion.

Il faut noter qu'il arrive que ces aides soient erronées. Dans l'exercice 8.7.2., on obtient l'aide :

Notez que les côtés AE et BC du quadrilatère AECB sont parallèles

alors que l'une des prémisses nécessaires est l'alignement des points A, E et D qui est complètement occulté.

Pour compléter ces aides, il faudrait avoir, outre les remarques déjà faites :

— la possibilité de *changer de contexte* (par exemple, passage du contexte du parallélogramme à celui du triangle), de nommer un point sur une figure ou de compléter cette dernière. Ces aides seraient utiles pour la phase heuristique.

— la possibilité de s'*éloigner du graphe* de plus d'un pas d'un chemin du modèle idéal, ce qui n'est pas permis même si la démarche est juste. Il est alors très gênant que le logiciel ne fournisse pas, au moins, un message d'aide qui signale à l'utilisateur qu'il ne s'est peut-être pas trompé, mais que le logiciel n'a pas les moyens d'analyser sa stratégie (cf.2ème écran, page 27).

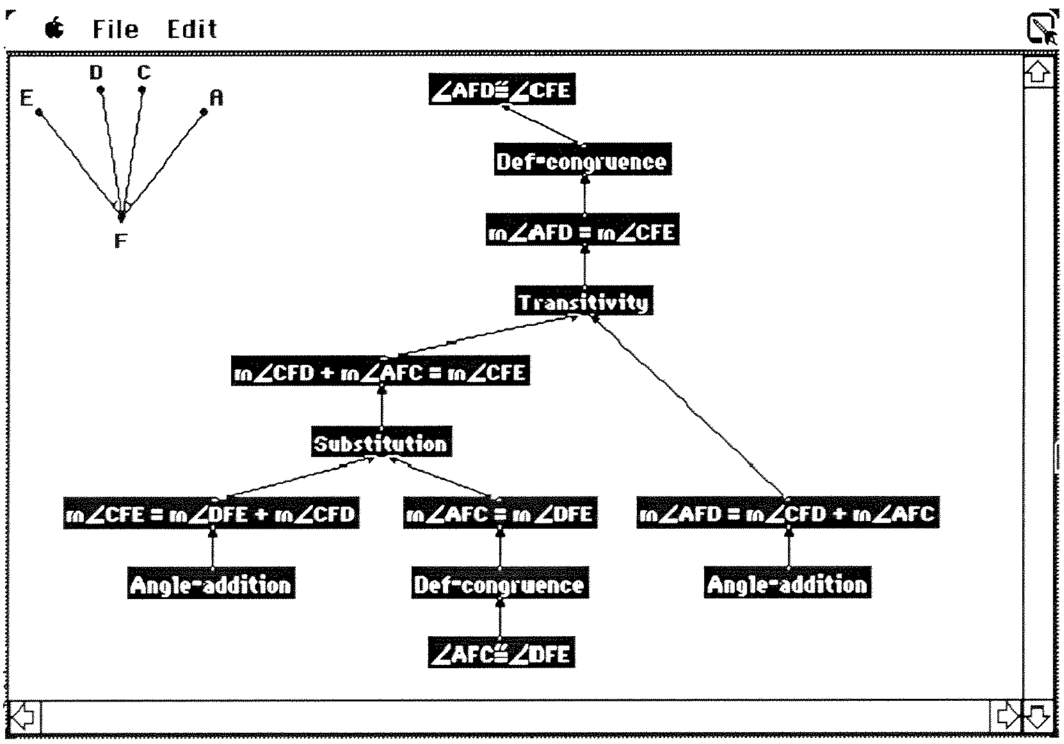
— un diagnostic des erreurs moins élémentaire (par exemple que le nom de la règle à utiliser soit faux ou qu'il n'y ait pas de règle proposée, l'élève se verra guidé de la même manière après deux essais faux). Dans l'exercice 8.7.1. (cf. page 24) où l'on cherche à démontrer qu'un quadrilatère est un trapèze, on obtient le message suivant :

" Que savez-vous des côtés opposés d'un trapèze ?"
Il y a là *confusion* entre une règle et sa réciproque. L'aide devrait porter sur les différentes façons de démontrer qu'un quadrilatère est un trapèze.

— une trace du comportement observable de l'élève. En effet, l'enseignant pourrait alors tenter d'interpréter ces comportements afin de mieux adapter son aide individuelle à l'élève.

e) types d'exercices

Nous avons remarqué précédemment que la base de connaissances était incomplète. Les exercices proposés font très souvent appel à des règles de calcul et ne permettent pas une réelle réflexion sur des objets géométriques. Après avoir résolu l'exercice 1.10.5, l'élève aura-t-il une conception qui lui permettra d'aborder d'autres problèmes sur ce sujet ?



De plus, les exercices proposés sont soit d'une complexité artificielle parce qu'utilisant des règles de calcul peu explicitées dans l'enseignement comme la substitution ou la transitivité de l'égalité (exercices 1.10.5 et 8.5.4.), soit très simples : il faut fréquemment appliquer une seule règle pour arriver à la conclusion (exercice 8.4.1 ou 8.7.2.).

Ce logiciel comporte beaucoup d'insuffisances du point de vue mathématique et pédagogique. Il ne permet sans doute pas à un élève de collège de prendre réellement conscience de ce qu'est une démonstration, mais lui permet de voir la structure d'un pas de déduction. Par contre, ce logiciel ne peut guère s'adresser à un élève plus initié car l'*atomisation* et la *lourdeur* apparaissent comme des *obstacles* à la *découverte* de la solution : une démonstration en géométrie suppose la mise en évidence préalable de *plan*. On ne peut émettre que des hypothèses sur l'origine de ces insuffisances : sont-elles dues à la théorie de l'apprentissage ou aux limites de la technique informatique? Peut-être sommes-nous aussi gênés par l'approche différente de l'enseignement de la démonstration géométrique en France et aux Etats-Unis ?

V Discussion

a) Un tutoriel pour les toutes premières étapes de l'apprentissage ?

Geometry Tutor est un tutoriel qui est disponible sur Macintosh et qui tourne effectivement. Il est "intelligent" dans le sens où il peut "interpréter" ce que fait l'élève en comparant les règles qu'il applique avec celles de "IBR" (cf. paragraphe III p.13). Signalons que, au contraire du modèle idéal, il n'y a aucun exemple de règle erronée dans les articles sur Geometry Tutor : nous n'avons donc aucune information sur le modèle erroné, ce qui ne permet pas de comprendre le fonctionnement du diagnostic d'erreurs. Une conséquence inévitable d'un tel fonctionnement est la contrainte pour l'élève de ne pas trop s'éloigner d'un chemin de preuve (cf. exemples page 27). Nous avons vu qu'il y a élimination des réseaux corrects non retenus par le logiciel et que, si l'élève s'écarte trop du modèle idéal, on lui suggère immédiatement la meilleure étape pour s'en rapprocher : si la réaction était retardée, les raisons de l'erreur seraient plus difficiles à trouver, il faudrait analyser le chemin. Remarquons que la version sur Macintosh permet uniquement un fonctionnement en marche avant pour des raisons techniques, alors que la version originelle comportait également un fonctionnement en marche arrière.

Le travail porte essentiellement sur les connaissance procédurales (savoir appliquer un théorème) qui constituent la première étape dans l'acquisition de la démonstration. L'apprentissage est très structuré : il repose sur l'hypothèse de *modularité* des connaissances (cf. paragraphe III page 10). Chaque chapitre comporte une série d'exercices dont le but est d'appliquer un certain type de théorèmes (cf. paragraphe II page 6) : ces exercices ne peuvent donc être résolus par une autre méthode, même si cette dernière fournit une solution plus simple et plus rapide. Nous sommes conscients que l'hypothèse de modularité des connaissances exprimées sous forme de règles de production (194 règles pour la version initiale, cf Anderson 1987-2 p. 113 ; 56 règles pour la version Macintosh) facilite le fonctionnement du tutoriel et n'est pas trop gênante dans les toutes premières étapes de l'apprentissage. L'atomisation des tâches se retrouve également dans Lisp Tutor : dans le contrôle des compétences, la durée moyenne d'une tâche est de l'ordre de 20 secondes (Anderson, 1989-1). C'est pourquoi J. Reiser parle de tâches concernant le *code de surface* (Reiser et alii, 1989).

Dans ce contexte, les aides sont de deux types : le premier, *procédural*, concerne l'application d'un théorème (exemple : vous n'avez choisi qu'une prémisse , le théorème choisi en nécessite deux). L'autre, *heuristique*, comporte la liste des théorèmes applicables qu'un élève peut essayer systématiquement d'ailleurs sans comprendre du tout ce qu'il fait : il pourra ainsi résoudre tous les exercices s'il sait appliquer correctement les théorèmes proposés. C'est pourquoi nous considérons que le travail demandé développe essentiellement les connaissances procédurales. Dès que l'on veut commencer à développer les connaissances heuristiques, une aide où l'on suscite le choix d'un théorème en rappelant qu'une de ses hypothèses ou sa conclusion a un rapport certain avec l'exercice nous paraît plus adaptée. Geometry Tutor en comporte quelques-unes[1] . De même, en ce qui concerne le diagnostic des erreurs, il nous paraît nécessaire d'un point de vue pédagogique de préciser le *type* d'erreur : pour que l'élève progresse, il doit comprendre la nature de l'erreur. C'est d'autant plus vrai lorsque "l'erreur" consiste à ne pas travailler dans le contexte souhaité par l'enseignant qui a conçu les exercices : il s'agit ici du *contrat didactique*[2] qui n'est pas clairement explicité. Il semble nécessaire d'adapter ensuite l'aide au type d'erreur : les messages d'erreurs deviennent particulièrement agaçants lorsque les erreurs sont dues à des fautes de frappe!

Le meilleur atout de Geometry Tutor est, sans aucun doute possible, la construction *progressive* d'un réseau déductif. C'est un instrument de *réification* puissant (Wenger, 87 p. 317) qui permet de mettre en évidence la structure ternaire (cf. page 14). Cette notion de réseau a

[1] *Voir page 29*
[2] *Etude des liens entre enseignant et élève qui déterminent, de façon très souvent implicite, le rôle de chacun et sont susceptibles d'affecter le produit de l'apprentissage*

déjà été exploitée auparavant dans les mathématiques en dehors de tout contexte informatique (Balacheff, 1978), et des recherches plus récentes confirment l'efficacité de son utilisation dans l'enseignement de la démonstration (Egret , 1989). De ce fait, l'article de (Schofield et alii, 1989) ne nous permet pas de dégager les véritables raisons des résultats encourageants donnés par les expérimentations en classe avec Geometry Tutor : il est possible que ces résultats soient dûs essentiellement à l'utilisation du réseau déductif qui donne une représentation concrète de la preuve. Celà confirmerait les résultats précités de Didactique des Mathématiques. De toutes manières, l'intérêt d'un tel outil dans un tutoriel de démonstration en géométrie se trouve confirmé.

Comme J.R. Anderson le reconnaît lui-même (Anderson, 1987-2), Geometry Tutor ne respecte pas le principe 7 (cf. page 11) car l'élève expert doit travailler comme le novice. C'est donc un tutoriel qui ne peut être vraiment utile que dans les premières étapes de l'apprentissage du "savoir comment" pour des tâches vraiment simples (comme pour Lisp Tutor, qui concerne l'apprentissage de la syntaxe et de la sémantique du langage). Dans l'utilisation de ce tutoriel, J.R. Anderson confirme que les élèves novices sont moins gênés par la directivité du système (principe 6), mais que ceux qui ont déjà une expérience de la programmation se montrent plus impatients (Anderson, 1987-2 p. 118). Il pense que le logiciel devrait pouvoir s'adapter au niveau de l'élève (principe 7) en combinant des règles primitives du modèle idéal, mais nous verrons dans le paragraphe suivant qu'à notre avis la compétence de l'expert ne se résume pas à savoir combiner les règles. Elle consiste également à comprendre les *raisons* des liens résumés par les règles, à imaginer les étapes d'un plan de démonstration en fonction du contexte et à donner les *grandes lignes* d'une démonstration avant d'entrer dans les détails. Ainsi, il nous semble que, lorsque l'expertise grandit, le principe 6 devrait être abandonné, ne serait-ce que pour pouvoir prendre en compte le principe 8 : une résolution par approximations successives est difficilement compatible avec une réaction immédiate aux erreurs. Le fonctionnement du tutoriel serait alors remis en question puisqu'il est basé sur l'évaluation du comportement de l'élève en fonction du modèle "IBR"(cf. page 13). Nous concevons aisément que la réalisation effective d'un tutoriel qui tourne nécessitait d'en réduire les ambitions dans un premier temps.

b) Distinction entre savoir-faire et expertise :

(S.J. Payne, 1988) souligne l'hypothèse implicite dans ACT* que la connaissance déclarative cesse de jouer un rôle dans les savoir-faire après le stade novice et n'est pas modifiée tout au long du développement de l'expertise : il nous semble qu'elle évolue simultanément avec les méthodes puisque ces dernières dépendent des représentations. Un élève ayant de bonnes compétences (qui peuvent être contrôlées) dues à l'acquisition de méthodes valables dans un certain domaine, peut se trouver complètement démuni si on étend ou si l'on modifie légèrement l'approche de ce domaine. C'est qu'il est nécessaire de distinguer le fait de savoir accomplir une tâche et celui de comprendre la structure conceptuelle de la tâche. L'aptitude au changement de représentations fait partie de l'expertise mathématique, c'est un des objectifs de l'enseignement des mathématiques que d'*entraîner* l'élève à ces changements de représentations en fonction du contexte adapté à la résolution (cf. pages 10 et 20).

La résolution de problèmes nécessite donc, aussitôt que l'on quitte le domaine des tâches élémentaires, de changer de représentation et, par conséquent, de modifier les connaissances procédurales correspondantes. D'où la nécessité d'une *représentation conceptuelle* de la connaissance dans le développement de l'expertise qui inclut :

— la capacité à résoudre un nouveau problème,
— le transfert des compétences à des domaines apparemment différents,
— la capacité de comprendre les raisons des liens résumés par les règles afin de préciser leur domaine de validité et de pouvoir justifier l'expertise.

Ces trois capacités devraient pouvoir être explicitées dans le modèle idéal. L'acquisition d'un savoir-faire lié au contexte, qui peut être testée, doit être distinguée de l'acquisition de l'expertise plus difficile à évaluer. Il est évidemment plus aisé de traiter les manifestations mesurables de l'apprentissage plutôt que les changements complexes des représentations. De toutes façons, l'acquisition du savoir-faire est un aspect nécessaire dans l'expertise, donc la réalisation *effective* d'un tel logiciel est une étape encourageante. Il faut savoir adopter un point de vue pragmatique et réduire ses ambitions pour réaliser quelque chose qui "tourne". Il serait souhaitable cependant de poursuivre la recherche cognitive de manière à pouvoir élargir l'objectif du logiciel à l'acquisition de l'expertise : c'est ce qu'a entrepris Reiser dans le domaine du Lisp (Reiser et alii, 1989), mais cela nécessite de faire intervenir de manière plus spécifique les *contenus* de connaissance afin de donner des explications mieux adaptées.

c) *Notre conception de l'expertise de la démonstration :*

Nous présenterons succinctement notre analyse cognitive et didactique de la démonstration, pour plus de détails, il est possible de consulter (Guin et Groupe IA, 1989 ; Guin, 1989). Nous distinguons deux types d'activités dans l'apprentissage de la démonstration en géométrie qui, dans un premier temps, doivent être dissociées : l'organisation déductive qui consiste à comprendre les règles du jeu de la démonstration et la recherche d'une solution .

La première activité est l'*organisation déductive* : pour que l'élève soit capable de découvrir une solution, il faut qu'il ait compris parfaitement les règles du jeu. Cette activité comprend tout d'abord le *décodage* de l'énoncé. Ensuite, toutes les règles et les assertions nécessaires à la résolution du problème sont fournies à l'élève, il doit alors les *organiser* dans un ordre qu'il établit, choisir une règle et contrôler à chaque étape la substitution effectuée quand il applique la règle. Cette activité lui permet de comprendre les différents *statuts opératoires* des assertions, et de développer ses connaissances procédurales. La création d'un réseau déductif par l'élève avec des conventions qu'il se donne (les statuts opératoires peuvent être mis en évidence par différentes couleurs...) est alors un outil de contrôle pour cette activité (Duval et Egret, 1989).

La deuxième activité concerne la découverte de la solution. En essayant d'expliciter les connaissances heuristiques qu'utilisent les experts, nous avons dégagé la notion de *plan* : il s'agit d'imaginer les grandes lignes d'un chemin entre les données et le but. Notre conception du plan n'est pas celle du "cognitivisme classique" comme par exemple celle de (Py, 1989), c'est-à-dire une description précise et intégrale de l'action. C'est plutôt celle de (Suchman, 1987), à savoir une façon schématique d'anticiper l'action, et de la guider, de pouvoir poser des jalons. L'*incomplétude* fondamentale des plans est une condition nécessaire de leur efficacité : un premier plan est élaboré, puis testé. S'il échoue, les raisons de l'échec sont utilisées pour modifier le plan initial, la solution sera atteinte par *approximations successives* (Chaudron, 1990). L'utilisation d'un *réseau de plan* (distinct du réseau déductif) où les noeuds peuvent être des sous-buts et les flèches des heuristiques (par exemple, identification de figures *prototypes*, Guin et Groupe I.A., 1989) pourrait favoriser cette deuxième étape à condition qu'un formalisme précis ne soit pas exigé à ce stade.

d) Comparaison avec Geometry Tutor :

Dans Geometry Tutor, l'élève n'a pas la charge du décodage de l'énoncé puisque le réseau lui est présenté avec les hypothèses et la conclusion mises en évidence : l'énoncé est donné sous une forme prédigérée. L'élève n'a pas non plus la charge entière de l'organisation déductive, puisqu'on lui demande *successivement* de choisir des prémisses (cf. page 7), une règle et de donner le résultat de l'application de la règle. La tâche est trop localisée pour qu'il y ait erreur sur les statuts opératoires des assertions. Dans ces conditions, il est très difficile pour l'élève de prendre conscience de ces statuts, alors que leur reconnaissance est une condition nécessaire pour la compréhension du processus de la démonstration.

L'activité demandée est un mélange de tâche heuristique (choix d'un théorème) et d'organisation déductive (application d'une règle). La démarche imposée de *pas à pas* est souvent un obstacle à la découverte de la démonstration qui n'est pas forcément une combinaison de marche avant et arrière. La reconnaissance d'un plan qui nécessite une véritable structure hiérarchique des buts (principe 3, cf. page 11) ou d'une *figure prototype* ne peut pas être prise en compte : l'élève idéal doit travailler comme le novice.

L'élève n'a pas la possibilité de s'éloigner trop d'un des chemins de preuves retenus par Geometry Tutor, même s'il n'y a pas d'erreur dans l'application des règles : dans ces conditions, il ne peut comprendre soit *pourquoi* une bonne solution est refusée par Geometry Tutor, soit *pourquoi* la solution proposée n'est pas bonne. Il n'a donc pas de possibilité de tester un plan pour comprendre pourquoi il échoue afin de l'améliorer. Pour une *mise au point* de la démonstration (de même que pour la mise au point d'un programme Lisp), il est nécessaire de pouvoir choisir le niveau auquel on travaille (Kaltenbach,1989) et de pouvoir procéder par approximations successives (principe 8).

Evidemment, cette activité n'est envisageable qu'avec des *aides heuristiques* pour l'élaboration d'un plan, ce qui exige une *interprétation* des *comportements observables* de l'élève pour modéliser ses conceptions. N. Balacheff a mis en évidence la nécessité de distinguer dans l'architecture des systèmes deux types de modélisation : le *modèle épistémique* (modélisation des conceptions de l'élève) s'obtenant, seulement après un *diagnostic* exigeant des choix, à partir du *modèle comportemental* de l'élève (Balacheff, 1989). Remarquons que ce dernier nécessite également des choix où intervient explicitement l'interface du système : le découpage en observables du comportement exige une discrétisation du réel. C'est à partir du modèle épistémique qu'une *explication* adaptée (Nicaud et Vivet, 1988) pourra être développée, que ce soit dans les aides heuristiques ou le diagnostic d'erreurs (Carrière et alii, 1989; Nicaud et Saïdi, 1989).

Pour comprendre le sens de la démonstration, l'élève doit construire entièrement *lui-même* son réseau déductif. L'acquisition des connaissances procédurales se réalisera simultanément. Une fois qu'il a compris les règles du jeu, il peut s'attaquer à la découverte de la solution dont l'apprentissage est plus délicat : celui-ci nécessite un travail par approximations successives avec des aides adaptées à chaque situation, le niveau de formalisme exigé durant cette activité doit donc être réduit.

e) Vers un tutoriel intégrant effectivement l'expertise

Afin de pouvoir envisager l'élaboration de tutoriels ayant pour objectif un processus d'acquisition des connaissances plus poussé, S.J. Payne propose une nouvelle représentation cognitive qu'il désigne par *modèle mental* (Payne, 1988) qui pourrait évoluer en fonction du niveau d'expertise. Le modèle mental est une représentation cognitive permettant de relier la connaissance conceptuelle et les compétences : elle est spécifique à un individu donné, pour un domaine particulier, et peut produire des simulations comportant des inférences et des explications. Le modèle mental fournirait les espaces-problèmes, représentant la connaissance minimale nécessaire à la réalisation d'une recherche orientée vers un but précis, mais contiendrait de plus la connaissance conceptuelle nécessaire à l'élaboration de l'espace-problème. Bien que l'idée soit prometteuse, cette théorie n'est pas encore aussi élaborée que celle de J.R. Anderson, mais elle ouvre la possibilité de ne pas seulement décrire les mécanismes des changements de représentation, mais encore les *conditions* de ces changements, que S.J. Payne désigne par les *contextes cognitifs*.

VI Conclusion

La démarche de J.R. Anderson qui consiste à élaborer un tutoriel à partir d'une analyse *cognitive* nous paraît la seule valable. Geometry Tutor a le mérite d'avoir été entièrement réalisé jusqu'à la version commerciale, ce qui est rare dans le domaine des tutoriels intelligents qui restent souvent à l'état de prototypes. Le modèle ACT* (cf. paragraphe III) sur lequel il est basé semble plus adapté aux premières étapes de l'apprentissage. Pour des apprentissages plus approfondis, même dans d'autres domaines que la géométrie (comme Lisp), il nous semble indispensable de faire également une analyse *didactique* faisant intervenir explicitement les *contenus* de connaissance.

Actuellement les enseignants de mathématiques français peuvent considérer que le tutoriel, sous sa forme actuelle, n'est pas utilisable même dans la classe de $4^{ème}$ où commence l'enseignement de la démonstration en géométrie. Toutefois, les critiques qui ont été faites en ce qui concerne l'expertise mathématique et pédagogique (cf. paragraphe IV) pourraient être prises en compte pour une amélioration du logiciel qui constituerait un module d'initiation expérimentable dans une classe. De toutes façons, cette réalisation est une étape nécessaire si l'on veut un jour pouvoir élaborer des tutoriels orientés vers les étapes ultérieures de l'apprentissage de la démonstration : il devrait pouvoir s'adapter au niveau d'expertise de l'élève, en lui permettant d'élaborer progressivement sa démonstration et de choisir sa méthode de résolution. Enfin, il faut reconnaître que les difficultés ne résident pas seulement dans les problèmes techniques que rencontrent les informaticiens, mais qu'il reste auparavant un travail important à fournir de la part des cogniticiens et des didacticiens pour définir ce que devrait être l'expertise d'un tel tutoriel.

Geometry Tutor a le mérite d'exister et d'indiquer les limites actuelles des Sciences Cognitives, de la Didactique et de l'Intelligence Artificielle. Notre discussion révèle simplement que Geometry Tutor doit être considéré comme une première réalisation effective dans le domaine des tutoriels intelligents en géométrie qui ne pourra être améliorée que grâce à une collaboration étroite entre les recherches cognitives, didactiques et informatiques.

BIBLIOGRAPHIE

Anderson J.R., 1984, Acquisition of proof skills in Geometry, in *Machine Learning, Ed R.S. Miichalski, J.G. Carbonnell, T.M.Mitchel , Springer Verlag.*

Anderson J.R., Boyle C.F., Yost G., 1985, The Geometry Tutor, in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Ed A. Joshi, Morgan Kaufmann Publishers, Los Altos.*

Anderson J.R., 1986, Knowledge compilation : The General Learning Mechanism, in *Machine Learning II, Ed R.S. Michalski, J.G. Carbonnell, T.M. Mitchell , Morgan Kaufmann Publishers, Los Altos .*

Anderson J.R., 1987-1, Skill Acquisition : Compilation of Weak -Method Problem Solutions, in *Psychological Review* , vol 94 n° 2, pp 192-210 .

Anderson J.R., Boyle C.F., Farell R., Reiser B.J., 1987-2, Cognitive principles in the design of computer tutors, in *Modelling Cognition, Editeur P Morris, John Wiley § Sons Ltd.*

Anderson J.R., 1989-1, Skill Acquisition and the Lisp Tutor, in *Cognitive Science*, vol 13, pp 467-505.

Anderson J.R., 1989-2, Psychology and Intelligent Tutoring, in *Artificial Intelligence and Education , Ed D. Bierman , J. Breuker , J. Sandberg , IOS Amsterdam .*

Balacheff N., 1978, Les graphes de démonstration : outil pour l'étude des démonstrations naturelles, Thèse de 3ème cycle, *Institut National Polytechnique de Grenoble.*

Balacheff N., 1989, Exigences épistémologiques des recherches en E.I.A.O, Actes des journées E.I.A.O du PRC I.A (J.-F. Nicaud et M. Baron), Cachan.

Carrière E., Delozanne E., Vivet M., 1989, Des connaissances pour produire des explications dans un tutoriel intelligent, Compte-rendu des Journées explications PRC IA, L.R.I., Gif-sur-Yvette.

Chaudron L., 1990, Raisonnement approché en mathématiques : l'"inférence continue", Actes du colloque de l'A.R.C, *Editeur INRIA.*

Duval R., Egret M.-A., 1989, L'organisation déductive du discours, Annales de Didactique et de Sciences Cognitives , vol 2, *Editeur IREM de Strasbourg.*

Egret M.-A., Duval R., 1989, Comment une classe de 4ème a pris conscience de ce qu'est une démarche de démonstration, Annales de Didactique et de Sciences Cognitives, vol 2, *Editeur IREM de Strasbourg.*

Guin D., Rousselot F., 1987, Aide à la recherche d'une démonstration (géométrie de 4ème), Actes du colloque E.A.O., Cap d'Agde, *Editeur A.D.I.*

Guin D., Groupe IA, 1989, Réflexions sur les logiciels d'aide à la démonstration, Annales de Didactique et de Sciences Cognitives, vol 2, *Editeur IREM de Strasbourg.*

Guin D., 1989, A cognitive analysis of geometry focused on Intelligent Tutoring Systems, Actes du workshop Modelling Student Knowledge in Geometry, Grenoble, à paraître chez Springer Verlag.

Kaltenbach M., Frasson C., 1989, Dynaboard : user animated display of deductive proofs in mathematics, in *Int J.Man-Machine Studies* ,vol 30 .

Newell A., Simon H.A., 1972, Human Problem Solving, *Editeur Prentice Hall, Englewood Cliffs, New Jersey.*

Nicaud J.-F., Vivet M., 1988, Les tuteurs intelligents : réalisations et tendances de recherches, T.S.I. vol 7 n°1, *Editeur AFCET-Bordas.*

Nicaud J.-F., Saïdi M., 1989, Explication en résolution d'exercices d'algèbre, Compte-rendu des Journées explications PRC IA , L.R.I , Gif-sur-Yvette.

Payne S.J., 1988, Methods and mental models in theories of cognitive skill, in *Artificial Intelligence and Human Learning, Ed John Self, Chapman and Hall Computing*, London.

Py D., 1989, Mentoniezh : An Intelligent Tutoring System in Geometry, in *Artificial Intelligence and Education, Ed D. Bierman, J. Breuker, J. Sandberg, IOS Amsterdam.*

Reiser J., Anderson J. R., Farrell R., 1985, Dynamic student modelling in an intelligent tutor for LISP programming, in *Proceedings of the Ninth International Joint Conference on Artificial Intelligence, Ed A. Joshi, Morgan Kaufmann Publishers, Los Altos.*

Reiser J., Ranney M., Lovett M. C., Kimberg D.Y., 1989, Facilitating Students' reasoning with causal Explanations and Visual representations, in *Artificial Intelligence and Education, Ed D. Bierman, J. Breuker, J. Sandberg, IOS Amsterdam.*

Schofield J. W., Evans-Rhodes D., 1989, Artificial Intelligence in the classroom : the Impact of a Computer-based Tutor on Teacher and Students, in *Artificial Intelligence and Education, Ed D. Bierman, J. Breuker, J. Sandberg, IOS Amsterdam .*

Suchman L. A., 1987, Plans and Situated Actions - The Problem of Human / Machine Communication, Intelligent Systems Laboratory, Xerox Palo Alto Research Research Center, *Cambridge University Press .*

Wenger E., 1987, Artificial Intelligence and Tutoring Systems, *Ed J.S. Brown and J.Greeno, Morgan Kaufmann Publishers, Los Altos.*