

J. F. NICAUD

J. PH. DROUHARD

J. M. GELIS

P. SIMONNET

Atelier APLUSIX

Publications de l'Institut de recherche mathématiques de Rennes, 1991, fascicule S6
« Vième école d'été de didactique des mathématiques et de l'informatique », , p. 215-219

http://www.numdam.org/item?id=PSMIR_1991__S6_215_0

© Département de mathématiques et informatique, université de Rennes,
1991, tous droits réservés.

L'accès aux archives de la série « Publications mathématiques et informatiques de Rennes » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/conditions>). Toute utilisation commerciale ou impression systématique est constitutive d'une infraction pénale. Toute copie ou impression de ce fichier doit contenir la présente mention de copyright.

NUMDAM

Article numérisé dans le cadre du programme
Numérisation de documents anciens mathématiques
<http://www.numdam.org/>

THEME 7

Séminaire : "Atelier APLUSIX"

par J.-F. NICAUD ⁽¹⁾, J.Ph. DROUHARD ⁽²⁾,
J.M. GELIS ⁽¹⁾, P. SIMONNET ⁽¹⁾

(1) LRI, C.N.R.S. UA 410, Université Paris XI,
F91405 ORSAY Cédex

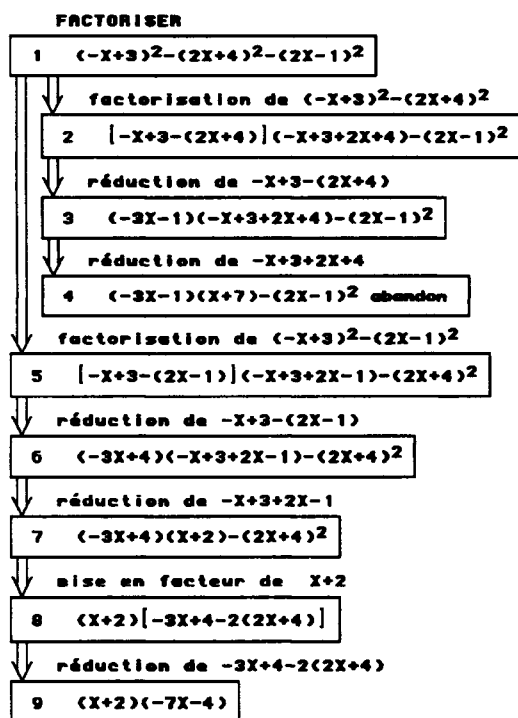
(2) I.N.R.P., 91, avenue Gabriel Péri, 92120 Montrouge

1. Introduction

APLUSIX est un projet d'EIAO portant sur les problèmes d'algèbre qui mettent en jeu des transformations d'expressions. Ce projet a été conduit au départ par des chercheurs en Intelligence Artificielle (IA) assistés d'enseignants de mathématiques de lycée. Plusieurs prototypes ont été développés dans le domaine des factorisations de polynômes, certains ont été expérimentés avec des élèves de seconde. Actuellement, le projet est conduit par une équipe pluridisciplinaire de 10 personnes et comporte plusieurs axes de recherche en algèbre élémentaire et non élémentaire.

La première partie du séminaire a comporté une brève présentation de la représentation interne des connaissances contenues dans les prototypes démontrés, une description des expérimentations qui ont été faites en 90 et 91, la présentation d'un protocole de 91. Le séminaire s'est poursuivi par la présentation et la discussion des nouvelles orientations d'APLUSIX en algèbre élémentaire.

2. Représentation interne des connaissances d'APLUSIX



*Exemple d'arbre développé
par le résolveur ou par l'élève.*

La maquette APLUSIX'90 montre un résolveur d'exercices de factorisation de polynômes cherchant à avoir une certaine pertinence didactique au niveau des connaissances et des mécanismes d'utilisation des connaissances. Ce résolveur développe un arbre, chaque application d'une transformation créant un nouveau nœud de cet arbre. Trois niveaux de connaissances ont été mis en œuvre dans ce système, la différence entre les niveaux porte sur les transformations applicables, les stratégies utilisables et la finesse des pas de calcul. Un module d'explication a été réalisé pour décrire la façon dont les transformations ont été choisies ainsi que la façon dont elles s'appliquent.

Le prototype APLUSIX'91 porte aussi sur la factorisation des polynômes et met en œuvre une activité de résolution d'exercices par l'élève selon un mode d'interaction favorisant l'apprentissage des savoir-faire et des stratégies : l'élève choisit des transformations à appliquer sur des expressions, l'ordinateur se chargeant des calculs. Les fonctionnalités pédagogiques de ce logiciel sont le contrôle des requêtes de l'élève et l'aide. Ce prototype ne comporte qu'un état de connaissance correspondant au niveau des classes de seconde.

APLUSIX est développé dans un environnement Lisp avec les modules de base suivants :

- un système de représentation interne des expressions algébriques,
- un ensemble de fonctions Lisp de calcul formel,
- SIM : un générateur de systèmes experts d'ordre un avec méta-règles,
- une interface de manipulation d'expressions algébriques à l'écran.

Les fonctionnalités raisonnées de résolution, explication, contrôle et aide sont implantées sous la forme de bases de connaissances écrites en SIM.

Résolution

Dans APLUSIX'90, le cycle de base du résolveur est :

- (1) déterminer les transformations applicables sur toutes les sous-expressions de l'expression courante (énoncé ou expression nouvellement engendrée),
- (2) appliquer les règles stratégiques pour attribuer une qualité (t-bien, bien, moyen...) à chaque transformation applicable obtenue,
- (3) choisir une expression pour continuer (expression courante ou autre expression de l'arbre plus prometteuse) et la meilleure transformation applicable à cette expression,
- (4) appliquer la transformation choisie.

Les connaissances en jeu sont structurées de la façon suivante (légèrement simplifiée) : une BC (Base de connaissances) pour les transformations, une BC pour des heuristiques attribuant une qualité aux transformations applicables sur des considérations assez générales (exemple : les transformations qui factorisent prennent la qualité bien), une BC de modification des qualités précédentes en tenant compte de propriétés de l'expression courante, une BC de modification des qualités précédentes en tenant compte de propriétés des autres expressions, une BC de choix de l'expression et de la transformation.

Exemple de règle stratégique :

si (?pb applicable ?ra)
(?ra effet devel)
(?ra substitution (?f ?f))
(?f polynome degre 2)
(soit ?g (expression-superieure ?f op +))
(?g polynome degre 2)
alors (favoriser ?ra 3)
commentaire (?g "est du second degré,"
"donc factorisable avec le discriminant")
propriete groupe local categorie degre2 *

*si le problème (l'étape) admet la règle applicable ?ra
et ?ra est une transformation factorisante
et on récupère dans ?f la valeur de la variable ?f de ?ra
et ?f est un polynôme de degré 2
et ?f se trouve dans une expression ?g qui est une somme
et ?g est un polynôme de degré 2
alors augmenter de 3 unités la qualité de ?ra*

Cette règle implante l'heuristique suivante :

*Dans un problème de factorisation, si une expression est du second degré
et est factorisable, on peut toujours la développer et la factoriser avec le discriminant.*

en favorisant les transformations de développement dans les contextes où cette heuristique est applicable. Cette heuristique n'est pas opérationnelle lorsque l'on se trouve à un niveau où le discriminant n'est pas connu. Cette règle est une méta-règle puisqu'elle porte sur une autre règle et sa substitution (par l'intermédiaire de la variable ?ra).

Pour plus d'informations sur le résolveur, se reporter à [Nicaud 89a, 89b].

Explication

L'explication apporte une réponse aux questions (1) comment a-t-on appliqué cette transformation ? (2) pourquoi a-t-on choisi cette transformation ? (3) pourquoi N'a-t-on PAS choisi la transformation T ? (4) pourquoi a-t-on effectué un retour en arrière ?

Les explications sont engendrées par méta-règles explicatives qui vont chercher les informations dans les transformations et dans les heuristiques appliquées et composent un texte à partir d'éléments de textes pris dans ces règles et d'éléments de textes propres.

Pour plus d'informations sur le module d'explication, se reporter à [Nicaud 90a].

Contrôle de la résolution de l'élève

Dans APLUSIX'91, l'élève résout des exercices en présentant des requêtes (transformation à appliquer sur une expression, demande d'aide, demande d'abandon). Le système a en charge d'évaluer la requête et de l'appliquer ou de la refuser.

Dans le cas d'une demande de transformation d'expression, l'évaluation de la requête est effectuée par des règles spécifiques et non pas par comparaison avec les transformations applicables du module de résolution. Cela vient du fait que le résolveur n'envisage pas toutes les transformations licites (qui seraient trop nombreuses). Exemples de messages émis lors de refus d'une requête :

Il n'est pas possible de factoriser X^2+X+1 avec $A^2+2AB+B^2$

Cette expression est réduite

Mettre 1 en facteur n'a pas d'intérêt

Préciser la partie de l'expression que vous voulez réduire

Aide

L'élève peut demander à voir les transformations applicables à une étape de son choix (premier type d'aide). Il peut aussi demander une aide générale (deuxième type d'aide), celle-ci est en fait une aide stratégique fondée sur les grands principes de la stratégie en vigueur. Dans le cas d'APLUSIX'91, qui a été conçu pour les élèves de seconde, les grands principes choisis consistent en une hiérarchie entre les transformations de réduction, de factorisation et de développement (après avoir éliminé les transformations qui ont une qualité très faible) : lors d'une demande d'aide générale, APLUSIX indique, dans l'ordre, les réductions et factorisations qui n'ont pas été appliquées en privilégiant celles de l'étape courante ; s'il n'y en a pas il indique qu'il faut se résoudre à (ou continuer de) développer. Cette aide utilise, bien sûr, des informations fournies par le résolveur.

3. Expérimentations

En 1990, 24 élèves d'une classe de seconde indifférenciée ont travaillé avec le logiciel durant quatre séances de 45 minutes. Un pré-test et un post-test, sous une forme papier-crayon, ont complété cette expérimentation. Les exercices de ces tests, comme ceux de l'expérimentation, étaient d'une difficulté dépassant le niveau habituel des classes de seconde. L'étude des pré et post-tests montre un incontestable progrès global des élèves après les 4 séances de travail avec APLUSIX. Globalement, la réussite passe de 10% à 34%. Entre ces deux tests, on constate aussi des progrès particuliers au niveau de certains savoir-faire. Ainsi, les appariements faux du type *mise ne facteur de $2X-3$ dans $(3-2X)(3X+6)-(8+4X)(2X+3)$* passent de 12 à 1. Pour plus de détails sur l'analyse de ces tests, voir [Nicaud 89b] ; pour une première analyse des protocoles, voir [Nicaud 90b].

En 1991, une autre expérimentation a eu lieu sur des élèves de seconde avec des exercices globalement plus faciles. Les pré- et post-test ont montré des progrès moins importants qu'en 1990, en particulier sur les exercices faciles.

4. Présentation d'un protocole

Le protocole présenté par Jean-Michel Gélis a été choisi parmi ceux recueillis en 91 et concerne un élève qui a factorisé avec succès 19 expressions sans commettre d'erreur grave dans la manipulation des différentes transformations. Ces 19 factorisations n'ont pas nécessité de retour en arrière, les expressions proposées étant suffisamment simples à factoriser pour que l'enchaînement des transformations qui semblaient les plus prometteuses au cours de l'exercice, suffise à arriver à la solution.

Ces exercices sont donc jusqu'à ce moment conformes à ceux pratiqués en classe, qui ne comportent pas de possibilité de chemins menant à une impasse, et donc pas de réflexion d'ordre stratégique.

Pour factoriser la 20^{ème} expression $(-X+3)^2-(2X+4)^2-(2X-1)^2$, l'élève utilise l'identité remarquable $(-X+3)^2-(2X+4)^2$ et, après les développements et réductions qui s'imposent, obtient $(-3X-1)(X+7)-(2X-1)^2$. Ses choix l'ont conduit à ce qui semble être une impasse. Pour mener à bien sa résolution, l'élève va devoir remettre en cause sa direction de travail.

Cette remise en cause s'opère difficilement, tout d'abord on trouve des affirmations (fausses) que l'expression est factorisée, des demandes d'aide sur les transformations possibles pour $(-3X-1)(X+7)-(2X-1)^2$, des enchaînements de développements et réductions inintéressants. Ensuite l'élève effectue un premier retour en arrière sur l'expression $(-3X-1)(X+7)-(2X-1)^2$. Revenir à cette expression, déjà exploitée et étudiée par l'élève, n'est pas prometteur, mais c'est ici que se pose le problème. L'élève enchaîne à nouveau développements, réductions, factorisations sans intérêt, avant d'opérer, après un très long temps de réflexion, un retour en arrière à l'expression initiale pour factoriser $(-X+3)^2-(2X-1)^2$, ce qui le conduit à la solution.

Nous proposons l'hypothèse suivante : le contrat didactique mis en place en classe pour les factorisations ne nécessite pas de retour en arrière, et donc pas de réflexion véritablement stratégique. Il semble bien ici que c'est la rupture de ce contrat qui pose problème à l'élève et provoque des demandes d'aide répétées, des erreurs sur le caractère factorisé d'une expression, des suites stériles de développements et réductions jamais observées auparavant dans son travail. L'expression $(-3X-1)(X+7)-(2X-1)^2$ a focalisé ces attitudes en étant le lieu visible de la rupture du contrat didactique, rupture qui va permettre à l'élève d'accéder à une véritable réflexion stratégique. L'utilisation d'environnement interactifs d'apprentissage, comme APLUSIX, permet de déplacer certains objets d'enseignement, en l'occurrence *d'ouvrir le monde clos de la factorisation* [Tonnelles 79] installé en classe.

5. Nouvelles orientations d'APLUSIX en algèbre élémentaire

Parmi les axes de recherche actuels du projet APLUSIX se trouve une reconception d'APLUSIX en algèbre élémentaire en intégrant des connaissances en didactique des mathématiques. Il s'agit de profiter des connaissances acquises dans les travaux précédents et de mettre à contribution les compétences didactiques de nouveaux membres de l'équipe pour réaliser un nouvel environnement d'apprentissage avec les objectifs suivants :

- être utilisable sur une assez longue phase d'apprentissage, i.e. une phase pendant laquelle la connaissance de référence évolue au niveau des concepts, des procédures, des savoir-faire ;
- être fortement paramétrable au niveau de l'interface, de l'aide, du contrôle, du guidage. Le paramétrage permet d'ouvrir le champ des possibilités pour les expérimentations ;
- être utilisable par un élève seul ou en groupe, avec ou sans professeur.

Un première ébauche de méthodologie de travail en 7 points a été fournie par J-F Nicaud :

1) modélisation des connaissances de référence en algèbre élémentaire et étude de l'algèbre enseignée dans les classes concernées : ce qui est connu de tous, ce qui devrait être connu, ce qui est objet de l'enseignement. Ceci doit permettre de définir différents états de connaissances de référence ainsi que les liens entre ces états ;

2) analyse des domaines de problèmes des classes concernées et des activités traditionnelles. Définition d'activités incorporant l'ordinateur dans les contextes classe+professeur+ordinateur, un-élève+ordinateur, 2-élèves+ordinateur...

3) génération d'exercices ;

4) spécification fonctionnelle globale pour permettre les activités choisies : (i) communication entre un utilisateur anonyme et l'ordinateur (pas de modèle d'élève, communication uniquement par rapport à la connaissance de référence), (ii) communication entre un élève répertorié et l'ordinateur (modèle de l'élève, communication personnalisée), (ii) communication entre le professeur et l'ordinateur (description de ce qui a été fait lors d'une session, de ce qui a été appris, du modèle de l'élève ; paramétrage du logiciel).

5) représentation interne des connaissances et des mécanismes d'inférence ;

6) spécification et mise en œuvre des différents modules ;

7) expérimentation et évaluation.

Jean-Philippe Drouhard est intervenu sur le premier point indiquant les problèmes qui se posent au niveau de la recherche des informations et donnant des pistes pour ce travail.

Ces pistes sont évoquées dans le tableau suivant qui indique, pour chacune des questions soulevées par le premier point de la méthodologie, les domaines de recherche didactique particulièrement pertinents a priori et les difficultés spécifiques à la question. Ce tableau est bien entendu très schématique ; en particulier, les recherches et les difficultés spécifiques ne sont pas cantonnées à telle ou telle question, et la liste n'est pas complète.

Question soulevée :	Domaines de recherche pertinents a priori :	Spécificités :
Connaissances de référence	* Point de vue validation Langage algébrique	Stratification (méta) Découpage du champ
Algèbre enseignée	Evaluation (cf. EVAPM) Transposition didactique	
Domaine des problèmes, activités traditionnelles	Théorie des Situations	Effets de contrat et de coutume
Activités impliquant l'ordinateur	Communication, Débats Imagiciels	Ergonomie Institutionnalisation

Pascale Simonnet a présenté les premières idées guidant son travail de conception d'un générateur interactif d'exercices. Celui-ci est appelé à fournir, en fonction de certains objectifs et de certaines contraintes prédéfinies, un exercice à tout moment de la session. Dans cette perspective il est nécessaire de définir des contextes d'apprentissage ainsi que des langages de description appropriés. Le générateur s'appuiera sur un réseau de modèles génériques devant permettre un contrôle de la progression des exercices sur les différentes connaissances qu'ils mettent en jeu.

6. Conclusion

APLUSIX a été conçu par des chercheurs en IA avec des problématiques d'IA mais aussi avec la recherche de fonctionnalités permettant de favoriser certains apprentissages. Les expérimentations effectuées semblent confirmer l'intérêt des principales fonctionnalités mises en œuvre. La poursuite des travaux en prenant en compte des connaissances de didactique des mathématiques devrait permettre la réalisation d'un logiciel s'intégrant mieux dans l'enseignement.

7. Références

- [Nicaud 89a] J.F. Nicaud : APLUSIX : un système expert pédagogique et un environnement d'apprentissage dans le domaine du raisonnement algébrique. TSI, vol 8, No 2, 1989.
- [Nicaud 89b] J.F. Nicaud, C. Aubertin, A. Nguyen-Xuan, M. Saïdi, P. Wach : APLUSIX : un environnement d'apprentissage à plusieurs niveaux en algèbre. Actes des premières journées EIAO de Cachan. 1989.
- [Nicaud 90a] J.F. Nicaud, M. Saïdi : *Explications en résolution d'exercices d'algèbre*. Revue d'Intelligence Artificielle, Vol 4, N°2, 1990.
- [Nicaud 90b] J.F. Nicaud, C. Aubertin, A. Nguyen-Xuan, M. Saïdi, P. Wach : APLUSIX : a learning environment for acquiring problem-solving abilities. AFCET Cognitiva-90, 1990.
- [Tonnelles 79] J. Tonnelles : le mode clos de la factorisation au premier cycle. Mémoire de DEA de l'université d'Aix-Marseille II, 1979.