MATHÉMATIQUES ET SCIENCES HUMAINES

MARYSE QUÉRÉ

Utilisation des ramifications dans l'analyse d'un contenu à enseigner

Mathématiques et sciences humaines, tome 53 (1976), p. 69-79
<http://www.numdam.org/item?id=MSH_1976__53__69_0>

Math. Sci. hum. (14e année, n°53, 1976, p.69-79)

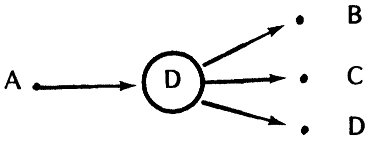
UTILISATION DES RAMIFICATIONS DANS L'ANALYSE D'UN CONTENU A ENSEIGNER

Maryse QUÉRÉ (*)

Les réflexions qui vont suivre sont assez anciennes, puisqu'elles datent de 1971. Elles étaient depuis restées dans un dossier, mais elles avaient débouché sur une recherche plus approfondie sur le problème de l'analyse et de la programmation d'un contenu à enseigner - travail qui n'est pas encore publié et qui ne le sera pas prochainement, puisqu'il s'insère dans une étude plus générale de l'Enseignement Assisté par Ordinateur, étude dont une communication faite à la seconde conférence IFIP «Informatique et Enseignement» (Marseille, 1975) a donné les contours - . Je croyais ces réflexions largement dépassées, mais à la lecture de publications récentes [1] [7] , je me suis aperçue qu'elles pouvaient encore présenter de l'intérêt puisque le modèle le plus largement utilisé dans le domaine reste celui des graphes (ou multigraphes), dont je pense personnellement qu'il n'est pas le plus adéquat.

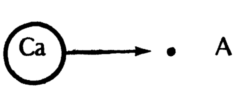
Rappelons ce dont il s'agit : un contenu à enseigner est représenté par un ensemble fini V d'éléments de connaissances et une ou plusieurs relations sur cet ensemble, par exemple [1] :

Relation D : Décomposition

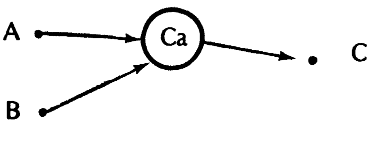


Le concept A est divisé dans les trois concepts B , C , D de nature différente.

Relation Ca : Caractérisation

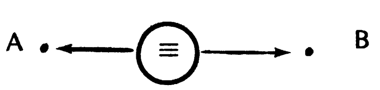


Le concept A est caractérisé par un certain nombre de descriptions, de résultats, d'exemples.....



Les caractéristiques du concept C sont données par les concepts A et B .

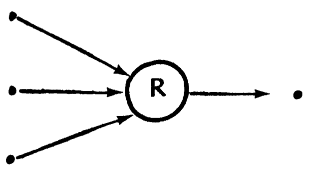
Relation ≡ : Equivalence



Dans le contexte où il est utilisé, le concept A (ou B) a la même signification que le concept B (ou A) .

(*) Maître-assistante à l'Université NANCY II .

Relation R : Production



L'ensemble des concepts A , B , C , sont conjointement nécessaires pour obtenir, suivant le raisonnement explicité, le concept D .

On dit alors en général qu'on l'a **analysé**. Nous verrons plus loin qu'à notre avis une analyse de contenu doit être un peu plus précise. Les éléments de l'ensemble devront être «présentés» aux personnes en apprentissage, soit de façon complètement séquentielle (cas de l'enseignement programmé skinnerien) , soit d'une façon qui permet plus ou moins les retours en arrière, le temps plus long passé à l'approfondissement d'une question, etc... (cas de l'enseignement assisté par ordinateur) . Dans tous les cas, il faut déterminer exactement quelle est l'information qui sera présentée après une autre dans le processus «normal» d'apprentissage, et qu'on appellera **successeur** d'une information. Si l'information α présentée au départ est connue, la suite α , α_1 = successeur (α) , α_2 = successeur (α_1) ..... détermine parfaitement l'ordre de présentation «théorique» des informations (c'est-à-dire un ordre qui ne tient pas compte des retours en arrière) . La détermination de la fonction successeur est appelée **programmation** du contenu.

Différents spécialistes de l'enseignement programmé se sont posé le problème de l'analyse d'un contenu en vue de sa programmation. Ayant choisi pour l'analyse des modèles différents, ils ont proposé pour la programmation des algorithmes différents, et souvent les ont opposés les uns aux autres. En outre, on relève deux tendances : plus la formalisation est poussée (dans le but d'automatiser le travail) , plus on s'éloigne de considérations pédagogiques [1] et réciproquement, si on ne pense qu'à la pédagogie, on fait preuve d'un manque de riqueur effrayant [3]. Mon propos est, en partant d'une analyse comparative des méthodes, une réunification à l'aide d'un objet mathématique couramment utilisé dans d'autres domaines de l'informatique que l'EAO, les **ramifications** (1) . Mais, comme je l'annonçais au début de cet avant-propos, nous en resterons ici à la description de l'existant et aux améliorations immédiates qu'on peut y apporter dans un sens didactique ou dans le sens d'une formalisation plus rigoureuse.

Les quatre méthodes décrites se proposent toutes d'inventorier un contenu et d'en donner l'organisation. Elles proposent toutes un ensemble V (plus ou moins rigoureusement défini, et recouvrant souvent des significations différentes) , des relations sur cet ensemble (auxquelles on peut appliquer la même remarque) . Le support graphique proposé est le graphe (ou multigraphe) , l'arborescence, ou inexistant (cas de [10] , mais on a vu que c'était convertible très simplement en ramification) .

A partir de cette organisation, trois des méthodes cherchent à présenter les éléments de V dans un ordre logico-pédagogique. La quatrième également, pas sous la forme que nous avons décrite, mais telle qu'on peut la voir mise en application dans des travaux ultérieurs [1], [7], [14] .

---

(1) Ma rencontre avec les ramifications a été facilitée par Monsieur Pair, que je remercie ici, et plus particulièrement par un de ses chercheurs [15] .

Comment améliorer ce bilan ?

- en permettant aux spécialistes de chaque discipline de définir l'ensemble V et les relations sur cet ensemble, à condition qu'ils le fassent de façon plus scientifique ; à cette occasion, on donnera des définitions, appuyées sur des exemples, de relations possibles. Ces relations seront des relations entre concepts, phénomènes, ou des relations liées à la didactique mais, à ce niveau, on ne devra s'embarrasser d'aucune considération technique.
- en donnant un support graphique (et informatisable) de V et des relations sur V, qui permette de faire apparaître des «cellules» [16], des niveaux de complexité (ou degré : [10]), de suggérer des révisions possibles de concepts antérieurement appris [3], qu'on puisse élaborer de façon automatique à partir de V, et dont la «lecture», possible grâce à un nouvel algorithme, fournira une programmation du contenu.
- en fournissant un algorithme de programmation.

Seuls les deux premiers points seront présentés ici.

1. Les informations et les relations entre les informations.

Soit V un ensemble dont les éléments seront appelés des **informations.** Deux sous-ensembles N et T disjoints tels que $V = N \cup T$ répartissent les informations en informations **connues** ou prerequisites (T) et en informations **à apprendre** (N).

Dans des exemples, nous avons essayé de montrer que le volume d'une information est déjà assez important. En particulier, nous refusons la bijection entre une information et un item de cours programmé, correspondance sur laquelle reposent l'analyse comportementale et l'analyse sémantique. Pour nous, toute la pédagogie relative à une information donnée (exemples, contre-exemples, démonstration, exercices, schémas,.....) est contenue dans cette information et ne constitue pas d'autres informations. En cela nous rejoignons Maurice Peuchot [12] avec son idée de module.

Donnons maintenant deux relations possibles. Elles sont transposables à d'autres domaines que les mathématiques [14].

1.1. L'antériorité directe $\mathcal{R}$.

$(\forall\, x \in V,\ y \in N)\ x\,\mathcal{R}\,y$ si l'apprentissage de y suppose **directement** la connaissance de x.

$\mathcal{R}$ est antiréflexive, antisymétrique et n'est pas transitive.

On peut définir $\mathcal{R}^p$ pour tout p entier, et on appellera $\mathcal{R}^*$ la fermeture réflexive et transitive de $\mathcal{R}$:

$$(\forall\, x, y \in V)\ x\,\mathcal{R}^* y \Leftrightarrow \exists\, n \in \mathbb{N} \wedge z_1, z_2 \ldots z_{n-1} \in V \text{ tels que}$$

$$x\,\mathcal{R}\,z_1\ ,\ z_{n-1}\,\mathcal{R}\,y \quad \text{et} \quad (\forall\, i \in \{1,\ldots,n\})\ z_{i-1}\,\mathcal{R}\,z_i$$

(l'apprentissage de y implique la connaissance de x par l'intermédiaire de celles de $z_{n-1},\ldots,z_1$).

$\mathcal{R}^*$ étant une relation d'ordre, son graphe est sans circuit non réduit à une boucle. On notera $\mathcal{R}^+$ la relation d'ordre strict obtenue en prenant $n \neq 0$ dans la définition précédente (fermeture transitive) . $\mathcal{R}^+$ est sans circuit.

Cette relation $\mathcal{R}$ formalise les relations de [6] , [16] et [3] , et la «décomposition» introduite en [14] .

Elle traduit l'organisation de la matière.

1.2 La proximité pédagogique $\mathcal{P}$.

La relation que nous allons introduire ici traduit, elle, certaines idées pédagogiques (qui en particulier peuvent varier) :

$(\forall\ x, y \in N)$ $x\, \mathcal{P}\, y$ si l'apprentissage complet (ou une révision complète) de x facilite l'apprentissage de y s'il est fait immédiatement, ou après.

La définition laisse croire à une certaine «symétrie» de la relation $\mathcal{P}$, on pourra s'imposer la contrainte :

(1) $(\forall\ x, y \in N)$ non $(x\, \mathcal{P}\, y$ et $y\, \mathcal{P}\, x)$

pour ne pas introduire de circuit dans le graphe.

L'adverbe «immédiatement» introduit dans la définition a pour conséquence que pour tout x de N , il existe au plus un y tel que $y\, \mathcal{P}\, x$.

Cependant, si on veut mettre n éléments de N en proximité, on pourra toujours fabriquer une chaîne $x_1, x_2, \dots, x_n$ avec pour tout $i \leqslant n$: $x_i \mathcal{P} x_{i+1}$.

1.3. Rapports entre $\mathcal{R}$ et $\mathcal{P}$.

$\mathcal{P}$ et $(\mathcal{R}^*)^{-1}$ sont incompatibles. En effet, $\mathcal{R}$ (et donc $\mathcal{R}^*$) traduisent une organisation «logique» de la matière. $\mathcal{P}$ ne doit pas détruire cette organisation en créant un circuit dans le graphe (V, $\mathcal{R}$ ou $\mathcal{P}$) .

On peut avoir $x\, \mathcal{R}\, y$ et $x\, \mathcal{P}\, y$: cela peut signifier, par exemple, que x est le plus important des prédécesseurs de y pour la compréhension de ce dernier.

Remarquons que les éléments de T n'ont de prédécesseur ni pour $\mathcal{R}$, ni pour $\mathcal{P}$.

2. L'analyse de contenu.

Ayant recensé tous les éléments de V et explicité les relations $\mathcal{R}$ et $\mathcal{P}$ sur cet ensemble, nous pouvons les représenter graphiquement par un multigraphe.

Nous appellerons **analyse de contenu** un «découpage» de V en sous-ensembles «complets» d'informations, que nous noterons V_i . Ce découpage ne sera pas comparable à celui d'un livre en chapitres, un ensemble complet d'informations n'étant pas la réunion disjonctive de «tranches de savoir» , mais plutôt un ensemble qui grossit chaque fois qu'est présenté un lot d'informations nouvelles. Les V_i seront donc choisis emboîtés $(V_i \subset V_{i+1})$, le dernier étant égal à V . En conséquence, toute différence $V_{i+1} \setminus V_i$ peut, elle, être comparée à un chapitre de livre, ou encore aux «cellules» définies en [16].

Chacune de ces cellules devra correspondre à l'apprentissage d'un **sous-objectif** de l'objectif terminal du cours correspondant à l'ensemble V, c'est-à-dire d'une information qui est le prédécesseur pour la relation $\mathcal{R}$ d'au moins deux éléments de V (sauf les prérequis qui ne sont pas considérés comme des sous-objectifs) . On donnera une représentation graphique claire de la restriction des relations $\mathcal{R}$ et $\mathcal{P}$ aux sous-ensembles V_i et aux cellules $V_{i+1} \setminus V_i$.

L'originalité de cette étude, par rapport à celles qui précèdent, est d'avoir accordé de l'importance aux informations qui servent plusieurs fois, et qui étaient génératrices d'ambiguités dans les méthodes analysées ([6],[16]) . Sans doute est-ce parce que cette situation est très fréquente dans l'enseignement des mathématiques.

La méthode d'analyse de contenu que nous avons choisi de présenter n'est pas indépendante de la programmation que nous voulons faire ultérieurement. Dans cet esprit, certains choix qui peuvent paraître arbitraires ici seront justifiés dans d'autres travaux. Nous utilisons les relations $\mathcal{R}$ et $\mathcal{P}$, mais n'importe quel pédagogue désireux d'utiliser d'autres relations peut s'inspirer de la méthode pour se fabriquer la sienne propre.

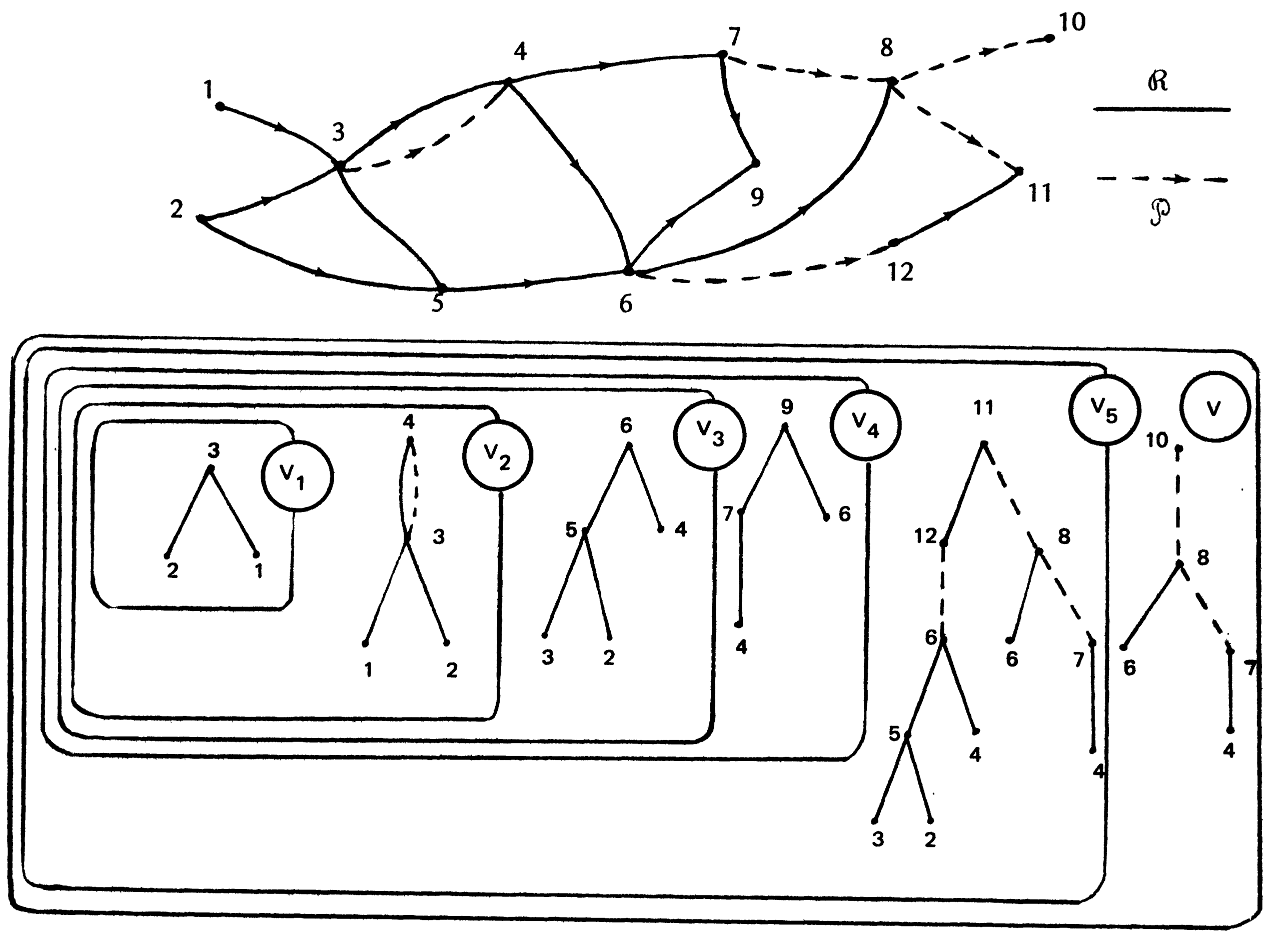
Nous avions le choix entre conserver le multigraphe ou généraliser les arborescences en ramifications. C'est la seconde voie que nous avons suivie, de façon peut-être assez intuitive au début (un gros graphe est illisible) , puis parce que la programmation sur une ramification était plus simple que sur un multigraphe. Aussi sans doute parce que le modèle de ramification rendait compte de l'analyse conceptuelle, donc dans un souci de généralisation. Surtout parce que rien n'est plus naturel comme démarche, pour définir l'ensemble V , que la construction «en remontant» de l'analyse comportementale.

Construction d'une ramification $\mathcal{O}$.

Il s'agit donc de transformer le multigraphe $(V,\mathcal{R},\mathcal{P})$ en une ramification faisant apparaître les V_i et les cellules. L'idée de l'algorithme est la suivante : quand une information est un sous-objectif de l'objectif du cours (au sens défini plus haut) , elle doit se comporter dans un V_i comme l'objectif du cours se comporte dans V (c'est-à-dire apparaître aux racines de la ramification) . Cependant on ne doit pas oublier ses relations à ses successeurs (qui seront dans d'autres V_j) , donc il existera d'autres occurrences de cette information dans la ramification, qui seront des feuilles, ce qui correspondra à l'idée de révision, alors qu'une occurrence aux racines correspond à l'idée d'apprentissage.

Comment se comporte la relation $\mathcal{P}$? On a dit en 3.2.2. que si $x \mathcal{P} y$, ce n'est pas seulement x qui facilite l'apprentissage de y , mais sa présentation «complète» , c'est-à-dire celle de x et de ses prédécesseurs. Cependant, il faudra bien fixer une limite à cette présentation des prédécesseurs (on ne peut pas «remonter» jusqu'aux prérequis).

Donnons un petit exemple (figure 1) où différents cas de rapports entre $\mathcal{R}$ et $\mathcal{P}$ se présentent. Les objectifs sont les points 9, 10 et 11 , les prérequis sont les points 1 et 2 . On choisit un objectif terminal (ici 10) et on construit l'arborescence dont il est la racine, en «s'arrêtant» dès qu'on rencontre un sommet qui «sert au moins deux fois» au sens de la relation $\mathcal{R}$ (c'est le cas de 6 et 4) . On choisit ensuite les autres objectifs, et on opère de même. Regardons ce qui se passe pour le sommet 6 , qui sert



Ramification $\mathcal{O}$

Figure 1. **Exemple de graphe et ramification associée (1).**

deux fois, mais qui est relié à 12 par $\mathcal{P}$: comme on pense qu'il est bon, avant d'apprendre 12 , de bien se rappeler comment 6 a été construit, on ne se contente pas de faire une révision de 6 , mais aussi de ses prédécesseurs, en s'arrêtant toutefois à 4 et 3 qui servent eux-aussi deux fois.

Quand tous les objectifs ont été pris en compte, on recommence le travail avec les sous-objectifs.

Algorithme de transformation du multigraphe en ramification.

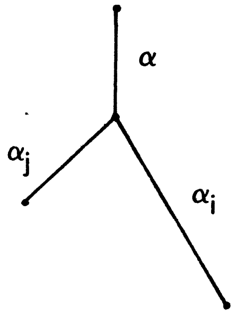
L'algorithme que nous décrivons procède par itérations. Nous allons décrire la $k^{\text{ième}}$.

1. Soit G_k le multigraphe $(V_k, \mathcal{R}, \mathcal{P})$; on dispose pour chaque sommet a du demi-degré extérieur pour $\mathcal{R}$ noté $s_k(a)$.
2. Soit π_k une pile (pour la définition, voir [4]) dont les états u_i sont des éléments de V^* et dont la construction est expliquée à la figure 2 .

---

(1) **Notation :** Quand nous entourons d'un trait marqué V_i une ramification, cela signifie que V_i est l'ensemble des étiquettes de cette ramification.

3. Construction d'une ramification r_k dont les chemins des racines aux feuilles sont les états «maximaux» de la pile π_k, c'est-à-dire les états u_i tels que $\forall\ \alpha \in V^*$, $\alpha \neq \Lambda \Rightarrow u_i\alpha$ n'est pas un état de la pile. Deux états maximaux u_i et u_j (u_i ayant précédé u_j) ayant un facteur gauche α commun donnent la ramification



$$u_i = \alpha\alpha_i$$
$$u_j = \alpha\alpha_j$$

4. $\mathcal{O}_k = r_k + r_{k-1}$.

5. G_{k+1} est le graphe obtenu en supprimant dans G_k les points condamnés qui n'ont pas de successeur ou dont tous les successeurs par $(\mathcal{R}$ ou $\mathcal{P})^*$ sont eux-mêmes condamnés, et les arcs correspondants.
   Calcul de $s_{k+1}(a)$ pour tout a . On applique ensuite la formule :

   $$s_{k+1}(a) = \text{ si } s_{k+1}(a) = 0 \text{ alors } 0 \text{ sinon } s_k(a) \quad ,$$

   ceci pour se souvenir des «sous-objectifs» , même si à l'étape $k + 1$ il ne leur reste plus qu'un successeur, sauf s'ils sont devenus à leur tour des objectifs.

6. L'initialisation se fait à $G_1 = G$, $r_o = \Lambda$. L'arrêt se fait pour $V_n \subset T$.

Remarques sur l'algorithme.

a) Rappelons que le graphe est sans circuit, donc l'algorithme converge (à chaque étape disparaît au moins un point du graphe et ce dernier est fini).

b) Les points x tels que $s_k(x) \geqslant 2$ ne disparaissent du graphe que lorsqu'ils ont été appelés dans la pile par chacun de leurs successeurs et que eux-mêmes sont devenus racines de la ramification (grâce à l'artifice qui consiste à conserver leur demi-degré extérieur constant tant que tous leurs successeurs n'ont pas disparu du graphe).

c) Tous leurs prédécesseurs (au sens de la fermeture transitive) sont protégés de la disparition, bien qu'ils aient pu être condamnés, par la clause 5 .

d)
$$\begin{cases} \mathcal{O} = r_1 + r_2 + \ldots + r_p \quad \text{où } (\forall i)\ r_i \text{ est une pseudo-arborescence} \\ V_1 = \{ x \in V \mid x \text{ a une occurrence dans } r_1 \} \\ V_2 = \{ x \in V \mid x \text{ a une occurrence dans } r_1 + r_2 \} \\ \vdots \\ V_p = \{ x \in V \mid x \text{ a une occurrence dans } \mathcal{O} \} \end{cases}$$

est une analyse du contenu V .

$u_o = \Lambda$.

si $u_i = \Lambda$ et s'il existe un point z du graphe appartenant à N , tel que $s_k(z) = 0$ et n'ayant pas d'entrée inférieure à i , i est entrée de z , sinon u_i est le dernier état de la pile.

(entrée d'un objectif - ou sous-objectif si $k \geq 1$ - non encore pris en compte)

si $u_i \neq \Lambda$ et a pour sommet x ,

si $s_k(x) \leqslant 1$, *(x ne sert qu'une fois au sens de $\mathcal{R}$, on va entrer ses prédécesseurs)*

s'il existe y tel que $y \mathcal{P} x$ et n'ayant pas d'entrée inférieure à i dans la pile au-dessus du même mot (2) , i est une entrée de y *(on commence par la relation $\mathcal{P}$)*

sinon

s'il existe y tel que $y \mathcal{R} x$ et n'ayant pas d'entrée inférieure à i dans la pile au-dessus du même mot, i est une entrée de y *(on continue par $\mathcal{R}$)*

sinon i est une sortie de x qui est condamné

(ce qui signifie que x ne servira peut-être pas à l'itération suivante)

sinon, *(x sert plusieurs fois au sens de $\mathcal{R}$)*

si non $x \mathcal{P} t$, où t est l'avant-dernière lettre de u_i , i est une sortie de x

(x sera une feuille de l'arborescence, mais est conservé pour une étape ultérieure)

sinon,

s'il existe y tel que $y \mathcal{P} x$ et n'ayant pas d'entrée inférieure à i dans la pile au-dessus du même mot, i est une entrée de y .

sinon,

s'il existe y tel que $y \mathcal{R} x$ et n'ayant pas d'entrée inférieure à i dans la pile en tant que successeur de x , i est une entrée de y .

sinon i est une sortie de x .

Figure 2. **Pile associée à l'algorithme.**

Il suffit de se reporter en arrière et de vérifier les contraintes imposées. En fait, au-delà des conditions qui étaient peu contraignantes, c'est l'utilisation du modèle, dont on verra quelques aspects ci-dessous, qui est intéressante.

3. CONCLUSIONS ET PERSPECTIVES.

Nous allons essayer de voir en quoi le modèle proposé nous semble pouvoir déboucher sur davantage d'applications que ses prédécesseurs. D'abord il permet, au même titre qu'eux, une programmation agréable du contenu. Ensuite il prend en compte la notion de révision ou rappel d'une information acquise antérieurement. On retrouve ici l'idée des modules «imbriqués» [1], [12] : même si l'élève garde la liberté de passer ou non par le rappel, ce dernier est prévu par le modèle.

(2) «au-dessus du même mot» signifie qu'on n'entrera pas deux fois dans la pile un sommet y appelé par x s'il n'y a pas eu modification de ce qui est entré avant x dans la pile.

Les expériences de Régnier ont montré que dans le cas d'un enseignement bien organisé, des contrôles effectués non pas après chaque élément de N mais après des cellules [16] rendaient l'enseignement programmé plus efficace. Non seulement la transformation du graphe en ramification n'interdit pas de garder leur définition des cellules (il suffit alors de donner une définition de fonction de ramification appropriée) , mais encore, on peut avoir l'idée d'envisager une planification automatique des fonctions de contrôle et de synthèse, gérée par l'ordinateur de façon conversationnelle avec le pédagogue. Pour cela, il faut aller un peu plus loin dans la description du processus de construction de la ramification associée à un cours.

On peut imaginer d'attacher à chaque terminal (ou non terminal) a des fonctions à valeurs entières ou réelles, par exemple :

* $\nu_a(\theta)$, **nombre d'occurrences de a** dans la ramification, qui signifie intuitivement que a **«sert beaucoup»** pendant l'apprentissage.

* $|\beta(a,\theta)|$ est le **poids** (nombre de sommets) des branches $\beta(a,\theta)$ de racine a dans la ramification.

  - Envisagée seule, elle peut mesurer la **complexité** de a si on suppose déjà connues les étiquettes des pseudo-arborescences situées à gauche de la première qui contient a dans la ramification.
    Notons que $(\forall\ a \in T) \qquad |\beta(a,\theta)| = 1$.

    Elle est dans ce cas à rapprocher de la notion de cellule.

  - Envisagée de façon cumulative, c'est-à-dire en considérant pour l'ensemble V_a des $x \in V$ tels que x a une occurrence dans $\beta(a,\theta)$ la somme

    $$\sum_{b \in V_a} |\beta(b,\theta)|$$

    puis, pour chaque $b \in V_a$, l'ensemble V_b et la somme

    $$\sum_{c \in V_b} |\beta(c,\theta)| .$$

    et ainsi de suite, elle permet d'avoir une bonne idée de la complexité de a dans l'ensemble du contenu.

Ceci est à rapprocher, quoique différent, du **degré** d'une variable selon Perriault, qui est attachée à la **hauteur** cumulée et non au poids cumulé de la variable [10] .

  - i(a) est la **quantité d'information** délivrée par a ; toutes les hypothèses sont permises au pédagogue pour la définition de ce nombre ; nous pensons qu'elle est directement liée à l'objectif du programme. Si le programme veut enseigner une terminologie, c'est à la définition des mots qu'on donnera le plus grand poids. S'il veut apprendre à résoudre des problèmes, c'est sur les propositions au contraire qu'on mettra l'accent. On peut bien sûr cumuler cette quantité d'information, comme dans le paragraphe précédent, en utilisant les familles de prédécesseur a .

Décider de faire un contrôle ou une synthèse après la présentation de a , et quelle sorte de contrôle ou de synthèse, va dépendre de la valeur prise par ces fonctions :

« i(a) $\geqslant$ seuil fixé » peut vouloir dire que a est un sous-objectif dont il faut vérifier l'acquisition,

« $\nu_a(\mathcal{G})$ $\geqslant$ seuil fixé » impose de bien s'assurer de l'acquisition de a , qui est une information clé de la discipline,

« $|\beta(a,\theta)|$ simple ou cumulée $\geqslant$ seuil fixé » impose de poser des questions «intelligentes» sur les constituants de a , nombreux à être mis en relation,

« i(a) cumulée $\geqslant$ seuil fixé » peut signifier qu'il est temps de faire le point.

Il est bien entendu qu'on peut mélanger toutes ces variables. Imaginons donc les informations contenues dans une base de données. Pour chaque information, les prédécesseurs par $\mathcal{R}$ sont connus. Le pédagogue livre à l'ordinateur l'objectif du cours, qui est un ensemble d'informations. De proche en proche, le graphe est construit : pour chaque information a , le pédagogue répond à des questions du type :

- quelle est la quantité d'informations délivrée par a ?
- est-ce que a $\in$ T ?
- quels sont les prédécesseurs de a pour les autres relations possibles dans le système ? (par exemple $\mathcal{P}$)

Quand l'ensemble d'informations qui constitue le cours a été énuméré, l'algorithme de construction de la ramification est mis en œuvre. Les fonctions ν_a et le poids de a sont calculées et présentées au pédagogue qui pourra donc planifier des synthèses et des contrôles destinés à une utilisation de l'enseignement assisté en mode tutoriel.

Puisque nous abordons maintenant les modes d'enseignement, donnons un aperçu de l'utilisation qui peut être faite du modèle en cas d'échec de l'étudiant à un contrôle prouvant que l'information a n'est pas acquise :

Les analyses de Perriault (3.4.3) sur la recherche des causes d'échec nous semblent très intéressantes. Elles devraient être poursuivies et complétées par des expériences sur la méthodologie qui suit l'échec :

- le retour se fait-il seulement sur le sommet a ?
- se fait-il sur la branche de racine a ?
- faut-il remonter encore plus loin dans les prédécesseurs de a ?
- si on lui présente un raisonnement type, l'étudiant est-il capable de choisir lui-même les sommets qu'il veut réapprendre ?
- etc.....

Pour résumer, c'est maintenant l'utilisation de la ramification par l'automate d'enseignement qu'il s'agit d'étudier.

BIBLIOGRAPHIE

[1] BENSMAINE M., **Contribution à l'étude du découpage et à la structuration d'une matière à enseigner,** TOULOUSE, Université Paul Sabatier, 1974.

[2] BERGE C., **Théorie des graphes et ses applications,** PARIS, DUNOD, 1958.

[3] CLOUZOT O., **«La méthode d'analyse sémantique d'un contenu», Enseignement Programmé,** n° 6, (1969), 21-30.

[4] DERNIAME J.C., PAIR C., **Quelques problèmes de cheminement dans les graphes,** PARIS, DUNOD, 1971.

[5] GAVINI G.P., **Manuel de formation aux techniques de l'enseignement programmé,** PARIS, Hommes et Techniques, 1965.

[6] INFA, **L'analyse comportementale,** document interne.

[7] JAYEZ J.H., **Etude des relations d'organisation de la matière en enseignement assisté par ordinateur,** TOULOUSE, Université Paul Sabatier, 1974.

[8] DE MONTMOLLIN M., **L'enseignement programmé,** PARIS, Presses Universitaires de France, 1971.

[9] MORGANOV I.B., **«L'utilisation des graphes dans l'élaboration des programmes», Enseignement Programmé,** n° 1, (1966), 19-33.

[10] PERRIAULT J., **«Nécessité d'une organisation conceptuelle pour l'enseignement de l'hématologie»,** in **La recherche en enseignement programmé, tendances actuelles,** PARIS, DUNOD, 1969.

[11] PERRIAULT J., DONIO J., **Nomenclature et structure des relations liant les concepts en hématologie,** IRIA, note interne, 1967.

[12] PEUCHOT M., **«Le dialogue ordinateur-élève. Introduction à l'analyse modulaire», Pédagogie,** n° 9, (1968), 101-111.

[13] PEUCHOT M., **«Conception modulaire de l'enseignement assisté par un ordinateur», Techniques de l'ingénieur,** n° 3 (1974), H8702 1-6.

[14] POINTEAU C., **L'analyse canonique,** IRIA, note interne, 1970.

[15] QUÉRÉ A., **Etude des ramifications et des bilangages,** NANCY, Faculté des Sciences, 1969.

[16] REGNIER J., DE MONTMOLLIN M., **Reconnaissance de l'organisation, recherche de l'ordonnancement des éléments et choix du mode d'enseignement de la matière,** in **La recherche en enseignement programmé, tendances actuelles,** PARIS, DUNOD, 1969