



HAL
open science

Récurrance et récursivité: analyses de preuves de chercheurs dans une perspective didactique à l'interface mathématiques-informatique

Nicolas Leon, Simon Modeste, Viviane Durand-Guerrier

► To cite this version:

Nicolas Leon, Simon Modeste, Viviane Durand-Guerrier. Récurrance et récursivité: analyses de preuves de chercheurs dans une perspective didactique à l'interface mathématiques-informatique. IN-DRUM 2020, Université de Carthage, Université de Montpellier, Sep 2020, Cyberspace (virtually from Bizerte), Tunisie. hal-03113854

HAL Id: hal-03113854

<https://hal.science/hal-03113854>

Submitted on 18 Jan 2021

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Récurrance et récursivité : analyses de preuves de chercheurs dans une perspective didactique à l'interface mathématiques-informatique¹

Nicolás León¹, Simon Modeste¹, et Viviane Durand-Guerrier¹

¹IMAG, Université de Montpellier, CNRS, Montpellier, France,
nicolas.leon@umontpellier.fr

Nous présentons l'analyse de preuves de chercheurs sur l'équivalence de deux définitions du concept d'arbre en théorie des graphes, l'une des deux définitions étant de type récursif et l'autre non. L'analyse vise à mettre en lumière la relation entre les notions de récurrance et de récursivité, telle qu'elle est perçue par les experts, afin d'éclairer les questions didactiques que soulève leur apprentissage.

Keywords : Epistemological studies of mathematical topics, Teaching and learning of logic, reasoning and proof, Teaching and learning of number theory and discrete mathematics.

INTRODUCTION

Contexte et problématique

Cette communication s'inscrit dans le cadre du projet ANR DEMaIn, qui explore les interactions entre mathématiques et informatique d'un point de vue épistémologique à visée didactique, et s'intéresse aux relations entre le raisonnement par récurrance et la récursivité dans la perspective d'une ingénierie didactique (Artigue, 2014). Les travaux existants montrent que la récurrance et la récursivité s'avèrent problématiques, de nombreux étudiants ayant du mal à comprendre les preuves par récurrance et les structures et algorithmes récursifs (Michaelson, 2008 ; Rinderknecht, 2014). Certains travaux semblent indiquer que la relation entre ces deux notions pourrait être l'une des clés pour aborder les difficultés qu'elles suscitent (voir par exemple Leron et Zazkis, 1986 ; Polycarpou, 2006).

Nous présentons dans ce qui suit une étude exploratoire de cette relation en analysant des preuves produites à notre demande par des experts ayant recours à la récurrance et à la récursivité dans leur travail. À travers cette analyse de preuves, nous cherchons à identifier des invariants opératoires pouvant attester des conceptions (au sens de Vergnaud, 2009) relativement à ces notions. Nous avons retenu pour cette étude la preuve de l'équivalence de deux définitions en théorie des graphes, l'une donnée sous forme classique par une propriété et l'autre sous forme récursive. Après une explicitation de notre point de vue sur l'analyse de preuves d'experts, nous présentons et analysons l'énoncé que nous avons soumis aux chercheurs et présentons ensuite les résultats des analyses de preuves menées.

Point de vue sur les preuves d'experts dans une perspective didactique

L'analyse de preuves mathématiques fait partie de l'activité du mathématicien ou de l'informaticien, mais également des professeurs, des élèves et des étudiants. Suivant Durand-Guerrier & Arzac (2009) nous considérons que l'analyse logique de preuves remplit trois fonctions principales : contrôler la validité ; comprendre la stratégie de l'auteur ; comprendre et s'appropriier les preuves d'une part en tant que constitutif de l'étude des contenus en jeu et d'autre part comme moyen de comprendre ce que sont les preuves mathématiques et leurs spécificités éventuelles dans un domaine donné. En outre, l'analyse logique de preuve est un moyen de repérer les implicites inhérents à la rédaction de preuves dont nous faisons l'hypothèse qu'ils peuvent éclairer les pratiques expertes. En accord avec Weber & Alcock (2004) et Wilkerson-Werde & Vilensky (2011) nous faisons l'hypothèse que l'étude du travail d'experts avec les preuves est de nature à éclairer nos questions didactiques.

Pour conduire les analyses de preuves, nous prenons comme logique de référence le calcul des prédicats du premier ordre avec un point de vue sémantique (Durand-Guerrier, Meyer & Modeste, 2019). Par ailleurs, Durand-Guerrier & Arzac (2009) identifient un certain nombre de questions à se poser lors de l'analyse de preuves. Nous listons ici celles que nous avons retenues pour nos analyses : Quels objets sont introduits tout au long de la preuve et quel est leur rôle ? À quelles évidences est-il fait appel et comment être sûr de pouvoir contrôler la validité de la preuve ?

Qu'entendons-nous par récurrence et récursivité ?

Dans cet article, le mot « récurrence » renvoie au sens habituel du terme, c'est-à-dire au type de raisonnement inductif portant sur les entiers naturels. Nous nous intéressons également au raisonnement dit « par induction structurelle », qui généralise en quelque sorte le raisonnement par récurrence.

La « récursivité » s'applique dans de nombreux contextes et avec des significations qui peuvent différer légèrement (par exemple, on associe la récursivité à des définitions, des algorithmes, des types de données, des suites, etc.). Pour penser aux articulations entre récurrence et récursivité (León, 2019), nous nous intéressons au cadre de la construction récursive (ou inductive) de structures, consistant à prendre un certain ensemble d'éléments de base et à en construire un autre par l'application réitérée de certaines fonctions, appelées « constructeurs » – c'est le cas, par exemple, des structures des listes et des arbres, en informatique, ou de celle des formules bien formées, en logique des propositions. Les raisonnements par induction structurelle permettent de prouver les propriétés des éléments de ces structures récursives, en montrant que ces propriétés tiennent pour les éléments de base, et que les fonctions constructeurs les préservent.

ÉNONCE SELECTIONNE ET ANALYSE MATHEMATIQUE A PRIORI

L'énoncé choisi est le suivant :

Démontrez l'équivalence des deux définitions ci-dessous :

Définition 1 : « Un graphe G est un arbre *si et seulement si* entre deux sommets quelconques de G , il existe un unique chemin ».

Définition 2 : « Un graphe G est un arbre *si et seulement si* G est soit un sommet isolé, soit un arbre auquel on ajoute un sommet pendant* ».

* Un sommet pendant d'un graphe est un sommet qui n'est adjacent qu'à un seul autre sommet.

Il s'agit de démontrer l'équivalence de deux définitions, l'une récursive (définition 2) l'autre par une propriété (définition 1). Nous faisons l'hypothèse que la rédaction de la preuve conduit à expliciter des connexions entre récurrence et récursivité.

Notre choix s'est porté sur la théorie des graphes car c'est un champ à l'interface des mathématiques et de l'informatique, qui peut être enseigné au niveau universitaire, et dans lequel la recherche est encore très active. Dans ce domaine, les différentes définitions ou caractérisations des objets sont souvent nécessaires pour pouvoir ensuite choisir le point de vue le plus adéquat, en fonction des besoins (preuves, définition, manipulation, traitement informatique, etc.). L'arbre est un objet récursif par excellence, suffisamment élémentaire pour produire des preuves accessibles, mais dont la construction récursive est suffisamment complexe pour être représentative d'une grande partie des situations récursives. Il existe de nombreuses définitions équivalentes de l'objet arbre, Ouvrier-Buffer (2015, p. 344) en identifie au moins 10, dont les définitions 1 et 2 ci-dessus, que nous avons retenues parce que la preuve de cette équivalence n'est pas classique dans l'approche courante sur la caractérisation des arbres. De plus, le fait que la récursivité ne soit apparente que dans l'une des deux définitions laisse ouverte la possibilité de voir apparaître des raisonnements par récurrence ou par induction structurelle.

Ouvrier-Buffer liste également la définition « G est un arbre *si et seulement si* G est un graphe connexe sans cycle », qui apparaît dans les démonstrations des chercheurs que nous avons analysées (voir la stratégie S2 ci-dessous).

Preuve de l'équivalence des définitions

Nous proposons d'abord une preuve complète de l'énoncé sélectionné.

Posons A_1 l'ensemble des graphes vérifiant la définition 1 et A_2 l'ensemble des graphes vérifiant la définition 2. On veut montrer $A_1 = A_2$.

Montrons $A_2 \subset A_1$:

Montrons cela par induction structurelle, c'est-à-dire, montrons que les éléments de base de A_2 sont dans A_1 et que l'opération de construction de A_2 préserve le fait de vérifier la définition 1.

- Soit $a \in A_2$, un sommet isolé. Alors $a \in A_1$.

- Montrons que l'ajout d'un sommet pendant à un graphe de A_1 le maintient dans A_1 .
Soit $a' \in A_1$ et a obtenu par l'ajout d'un sommet pendant à a' .
Dans a' , pour tous sommets s et s' , il existe un unique chemin de s à s' .
Soient s_1 et s_2 deux sommets de a , montrons l'existence d'un unique chemin entre eux dans a .

Existence :

- Si s_1 et s_2 sont des sommets de a' : c'est garanti par $a' \in A_1$.
- Si s_1 est le sommet pendant ajouté à a' , notons s' son voisin dans a .
Comme s' et s_2 sont dans A_1 , il existe un chemin (s', \dots, s_2) dans a' .
On construit alors un chemin (s_1, s', \dots, s_2) dans a entre s_1 et s_2 .
- Si s_2 est le sommet pendant ajouté à a' : idem.

Unicité : Soit $C=(s_1, \dots, s_2)$ chemin de s_1 vers s_2 dans a . Montrons son unicité.

Soit s' le sommet pendant ajouté et s son voisin.

Comme $\deg_a(s') = 1$, si $s' \in C$, alors $s' = s_1$ (ou de façon similaire $s' = s_2$).

- Si $s' = s_1$, alors $C = (s_1, s, \dots, s_2)$ et le chemin entre s et s_2 est unique car on n'a pas ajouté d'autres arêtes dans a' que (s_1, s) . Et alors, le chemin C est unique dans a .
- Sinon, il existe un unique chemin entre s_1 et s_2 dans a' , car on n'a pas ajouté d'autres arêtes dans a' que (s_1, s) .

On en déduit que $A_2 \subset A_1$.

Montrons $A_1 \subset A_2$:

Soit $a \in A_1$, montrons que l'on peut construire a à partir d'un sommet isolé en ajoutant des sommets pendants.

- Si a a un seul sommet : alors $a \in A_2$.
- Sinon, il existe s un sommet pendant (donc de degré 1). (Sinon, tous les sommets seraient de degré supérieur ou égal à 2 et il existerait un cycle dans a , donc il n'y aurait pas unicité des chemins entre deux sommets).

On retire le sommet s à a (et l'arête adjacente) pour obtenir un graphe a' .

a' est un élément de A_1 , puisque tout chemin entre deux sommets de a (différents de s) est préservé et reste unique.

En réitérant le processus, et puisque le nombre de sommets diminue strictement, on obtient une déconstruction de a , jusqu'à un sommet isolé, par élimination de sommets pendants. En inversant cet algorithme, on peut construire a à partir d'un sommet isolé par l'ajout de sommets pendants.

Donc $a \in A_2$. \square

Variante de $A_1 \subset A_2$ par « élément minimal ».

Supposons qu'il existe des éléments de A_1 qui ne soient pas dans A_2 , et soit a un plus petit élément de A_1 (au sens du nombre de sommets) qui n'est pas dans A_2 .

a a au moins 2 sommets. De plus, dans a il existe s un sommet pendant (donc de degré 1) : sinon, tous les sommets seraient de degré supérieur à 1 et il existerait un cycle dans a , donc il n'y aurait pas unicité des chemins entre deux sommets.

On retire le sommet s à a (et son arête adjacente) pour obtenir un graphe a' .
 a' est un élément de A_1 , puisque tout chemin entre deux sommets de a (différents de s) est préservé et reste unique.
 Et $a' \in A_2$ par minimalité de a . Et donc, par construction, $a \in A_2$. Contradiction.
 Donc $A_1 \setminus A_2 = \emptyset$, autrement dit $A_1 \subset A_2$.

Cette preuve nous servira de référence dans l'analyse des preuves de chercheurs, en particulier pour identifier des implicites.

Preuve générique de l'équivalence de deux définitions

Comme d'autres preuves sont envisageables, et afin de permettre une analyse plus complète, nous présentons une étude de la preuve générique de l'équivalence entre une définition par propriété et une définition récursive. De façon générale, on a deux définitions l'une sous forme d'une propriété, l'autre récursive :

Soit E un ensemble.

Def 1 : Un $x \in E$ est de type 1 si et seulement si $P(x)$.

Def 2 : Un $x \in E$ est de type 2 si et seulement si $x \in \{x_0, \dots, x_k\}$ ou x est obtenu par application d'un opérateur f_1, \dots, f_n à un élément de type 2.

L'analyse mathématique et logique *a priori* des preuves possibles de l'équivalence de deux telles définitions nous permettra d'identifier certains types de stratégies et de les repérer dans les preuves des chercheurs.

Pour prouver que Def 1 équivaut à Def 2, il faut montrer que x est de type 1 si et seulement si x est de type 2.

type 2 \Rightarrow type 1 :

Il faut prouver que les x_i sont de type 1 (i.e. vérifient P) et que les opérateurs f_j préservent le fait d'être de type 1 (si a est de type 1 alors $f_j(a)$ est de type 1).

C'est une induction structurelle, que l'on pourrait aussi transformer en récurrence classique sur \mathbb{N} .

type 1 \Rightarrow type 2 :

Il y a plusieurs alternatives pour parvenir à cette preuve. Par exemple :

1. On peut montrer que si un élément a satisfait $P(x)$, alors
 - i. Soit il est égal à un x_i , et donc il est de type 2.
 - ii. Soit il existe un constructeur f_j tel que $a = f_j(b)$, avec $b \in \{x_0, \dots, x_k\}$ ou $P(b)$. Par un argument de finitude de cette « déconstruction » on s'assure qu'il existe un x_i à partir duquel on peut construire a par application répétée d'opérateurs de type f_j . Ceci implique que a est de type 2.
2. On peut raisonner par l'absurde et par élément minimal : on suppose qu'il existe un élément a de type 1 qui n'est pas de type 2. Sans perte de généralité, on suppose qu'il est minimal. Ensuite, on montre qu'on arrive à le construire à partir d'un élément « plus petit » b par application de l'un des constructeurs f_j .

Par l'hypothèse de minimalité de a , b est de type 2. On en déduit que a est aussi de type 2, ce qui est contradictoire avec l'hypothèse de départ.

Quelle que soit la méthode choisie, il faut noter l'importance d'avoir un ordre bien fondé, plus ou moins explicite, assurant que la « déconstruction » se « termine ».

Sur la base de ces analyses, nous avons étudié les preuves des chercheurs. Nous rendons compte d'une partie de nos analyses dans la section ci-dessous.

ANALYSE DES DEMONSTRATIONS DES CHERCHEURS

Méthodologie et données recueillies

Dans le cadre plus général de notre étude, nous avons réalisé des entretiens avec des chercheurs en mathématiques et en informatique. Les chercheurs ont été invités, à la fin de l'entretien, à démontrer l'énoncé que nous avons présenté et à nous envoyer leurs preuves plus tard. Nous leur avons indiqué seulement qu'elles seraient analysées pour mieux cerner les pratiques des experts : en particulier, nous n'avons pas spécifié qui pourrait être le lecteur hypothétique de la preuve. Les chercheurs n'ont eu aucune contrainte de temps pour fournir leur réponse. Parmi les 10 interviewés, nous avons reçu des preuves de trois chercheurs (C1, C2, C3). En particulier, C1 nous a envoyé non seulement sa démonstration finale, mais aussi trois brouillons que nous avons également étudiés. C1 est un informaticien ayant fait des recherches en algorithmique et complexité, C2 est un mathématicien spécialisé en mathématiques discrètes et C3 est un mathématicien qui développe des recherches en informatique théorique.

L'analyse détaillée des preuves des experts, qui ne sera pas présentée ici, a été menée à la lumière de l'analyse mathématique *a priori*. Cela nous a permis d'identifier et d'explicitier leurs stratégies de preuve. Nous voyons ces stratégies comme de potentiels invariants opératoires, pouvant rendre compte des conceptions des chercheurs autour de la récurrence et de la récursivité. En outre, nous avons étudié les implicites dans les preuves conformément à notre positionnement didactique.

Stratégies des chercheurs

Pour la suite, nous introduisons les notations suivantes :

- $UC(G) :=$ entre deux sommets quelconques de G , il existe un unique chemin.
- $R(G) :=$ soit G est un sommet isolé, soit G contient un sommet pendant x tel que $R(G \setminus \{x\})$.
- $AC(G) := G$ est acyclique et connexe.

S1 : Chercher une injection vers \mathbb{N} ou vers \mathbb{N}^m .

Il s'agit de la recherche d'une injection de la structure de départ (ici, l'ensemble des graphes non orientés) vers \mathbb{N} , ou plus généralement vers \mathbb{N}^m , permettant d'effectuer un raisonnement par récurrence. Pour cela on peut considérer la fonction qui associe à un graphe G son ordre $|G|$, comme le font C1 et C2. Notre analyse mathématique *a*

a priori montre l'utilité de cette injection pour prouver que $UC(G)$ implique $R(G)$ dans la variante « par élément minimal ».

Nous avons observé un autre choix dans le premier brouillon de C1, où il cherche une injection vers \mathbb{N}^2 , associant à un graphe G le couple (v, l) constitué du nombre de chemins de taille maximale dans G et de la longueur de chacun de ces chemins. L'injection peut également permettre d'effectuer un travail par disjonction de cas, comme le fait C2 quand il considère d'abord les graphes d'ordre 1 et ensuite ceux d'ordre supérieur. C1 utilise lui aussi une injection vers \mathbb{N} , mais qui envoie chaque sommet vers son degré, pour réaliser une disjonction de cas.

S2 : *Passage vers une définition alternative équivalente*

Les chercheurs se servent de définitions équivalentes du même objet pour prouver ses propriétés plus aisément. Dans notre cas, le couple de propriétés « acyclique et connexe » joue ce rôle de médiateur qui facilite le passage d'une définition à l'autre. C1 rend explicite l'équivalence $UC(G) \Leftrightarrow AC(G)$, dont il se sert à plusieurs reprises, dans un sens et dans l'autre. Pour C3 il est suffisant d'établir que $UC(G) \vee R(G)$ implique que G est connexe, et puis de faire appel implicitement à l'implication $UC(G) \Rightarrow AC(G)$ et explicitement à l'implication $R(G) \Rightarrow AC(G)$.

S3 : *Déconstruire/reconstruire le cas générique*

La déconstruction du graphe générique G peut se faire à plusieurs niveaux, selon le but recherché. Puisque la définition récursive fait référence à la présence d'un sommet pendant, il semble naturel de déconstruire le graphe en enlevant ce sommet pendant, pour explorer les relations entre le sous-graphe ainsi obtenu et le graphe de départ. Bien entendu, dans le cas où l'on présuppose $UC(G)$, l'existence d'un sommet pendant doit être prouvée au préalable. Autant C1 que C2 recourent à cette déconstruction. Nous trouvons une approche différente chez C3, qui fournit un algorithme de construction d'un graphe G tel que $UC(G)$, à partir d'un sommet quelconque lui appartenant s_0 . Cet algorithme ajoute les sommets de G par « couches » selon leur distance par rapport à s_0 et est conçu pour montrer qu'effectivement G peut être obtenu à partir du sommet s_0 par des ajouts successifs de sommets pendants. Cette façon de procéder est proche de celle que nous avons conçue lors de l'analyse mathématique *a priori* ($A_1 \subset A_2$) ; la seule différence étant que dans notre algorithme les sommets sont ajoutés un par un.

S4 : *Décrire un algorithme construisant un objet pour prouver son existence*

Pour prouver que si G est tel que $|G| \geq 2$ et $UC(G)$, alors G contient au moins deux sommets pendants distincts, C2 élabore un algorithme permettant de trouver un chemin de taille maximale à partir d'un chemin quelconque dans G . Les deux sommets pendants correspondent alors aux extrémités du chemin maximal. Le contenu d'une itération de l'algorithme est présenté par l'ajout d'un sommet au chemin de départ, sous l'hypothèse que ce chemin n'est pas déjà de taille maximale. Le choix de montrer cet algorithme de construction nous semble intéressant. Il aurait

été envisageable d'argumenter qu'un chemin maximal doit exister, les sommets du graphe étant finis, sans avoir à détailler comment on l'obtient – d'ailleurs, C2 fait appel à la finitude du nombre de sommets pour justifier que son algorithme s'arrête.

S5 : *Dérécurser une définition récursive*

C3 nous met en garde contre les « dangers » des définitions récursives : « Les définitions récursives sont toujours un peu dangereuses dans le sens qu'elles font parfois penser aux fausses preuves qui supposent le résultat pour le démontrer. ». Pour s'en débarrasser, il propose une définition non-récursive qui est censée être équivalente à la définition 2 : « Un graphe G est un arbre *si et seulement si* G peut être obtenu à partir d'un sommet isolé en insérant successivement ses autres sommets comme des sommets pendants. »

Cette nouvelle définition « algorithmique » pose cependant deux questions : celle de valider son équivalence à la définition 2, et celle de décider si la nouvelle définition n'est pas elle-même récursive de façon détournée, cachée par l'expression « successivement ». Quoi qu'il en soit, dès que l'équivalence entre une définition récursive et une définition non récursive a été établie, on peut ensuite faire appel au point de vue le plus adéquat en fonction des besoins.

S6 : *Décrire/évaluer des cas simples d'une construction inductive ou de l'hérédité d'une récurrence*

C3 étudie des cas particuliers de l'hérédité d'une propriété, par exemple $P(1) \Rightarrow P(2)$, avant d'aller vers la preuve de $P(n) \Rightarrow P(n+1)$. On peut interpréter que cette redondance a des fins purement explicatives, ou supposer que c'est par l'exploration de ces cas élémentaires que C3 parvient à rédiger ensuite la preuve du cas général, voire s'assurer que les cas élémentaires y correspondent bien. Par ailleurs, il applique une procédure analogue pour définir une suite finie de sommets $(s_i)_{i=0, \dots, |G|}$ et une relation de filiation qu'il utilise pour prouver que $\forall G R(G) \Rightarrow UC(G)$.

Identification d'implicites

Le chercheur peut choisir de sauter une étape de la preuve pour plusieurs raisons. Par exemple, il peut considérer que cette partie a peu de valeur explicative ou trouver qu'elle est fastidieuse pour le lecteur, ou pour lui-même. Mais on peut supposer que, dans certaines situations, le chercheur peut sauter une étape d'une manière moins volontaire, parce qu'un certain geste répétitif lui est devenu si familier qu'il n'a plus besoin de franchir les pas intermédiaires pour arriver à sa conclusion. Même s'il est difficile d'identifier les causes des ellipses rencontrées dans les démonstrations, nous pensons qu'elles nous renseignent sur les possibles raccourcis dans le raisonnement des chercheurs. Parmi les implicites, nous pointons ceux qui sont en lien direct avec les raisonnements de type inductif :

Faire l'ellipse du (des) cas de base

On trouve cette sorte d'ellipse dans le travail de C1, avec des omissions plus ou moins drastiques : au premier brouillon, il qualifie le cas de base de « trivial », mais il

consacre quand même une ligne à l'expliquer ; au deuxième brouillon, il continue à qualifier le cas de base de « trivial », mais ne donne plus aucune explication ; et la récurrence de la version finale ne mentionne même pas le cas de base. Nous voyons donc, que même chez un seul chercheur la pratique consistant à omettre la preuve du cas de base n'est pas systématique et pourrait dépendre de la difficulté relative qu'il y attribue, ou de sa familiarité avec ce cas, parmi d'autres facteurs.

Faire l'ellipse du raisonnement inductif en entier

Nous voulons dire par là que le chercheur affirme sans démonstration un fait qui *pourrait* être démontré par un raisonnement inductif, et que ce raisonnement est soit mentionné mais non développé, soit complètement éclipsé. Nous trouvons un exemple du premier cas dans la preuve de C1 où ce qui serait formellement énoncé comme un raisonnement par induction structurelle est résumé par la phrase « $R(G) \Rightarrow AC(G)$ trivial, car le procédé de construction maintient invariant les propriétés de connexité et d'acyclicité ». Nous soulignons ici que ce raisonnement inductif, associé à la preuve de $UC(G) \Leftrightarrow AC(G)$, que C1 qualifie également de triviale, constitue l'une des deux implications ($R(G) \Rightarrow UC(G)$) de l'énoncé à prouver. Nous pensons que C1 est prêt à résumer cette procédure inductive aussi brièvement car il est très habitué à ce qu'il suffise de trouver un invariant du (des) constructeur(s) pour en déduire l'implication universellement quantifiée associée.

Nous trouvons aussi des ellipses complètes, où un fait mathématique pouvant être prouvé par récurrence est affirmé, sans aucune allusion à la preuve. Par exemple, dans un brouillon, C1 affirme sans démonstration qu'il ne peut y avoir de cycle dans les arbres selon la définition 2. On peut prouver ce fait par induction structurelle, en montrant que le graphe à un seul sommet est acyclique, et que si l'on ajoute un sommet pendant à un graphe acyclique, il reste acyclique. L'acyclicité des arbres est une propriété classique présente dans plusieurs définitions et il peut être difficile de travailler sur les arbres sans utiliser ces propriétés bien connues.

Nous voyons un autre exemple dans l'argument que C3 donne pour affirmer la connexité de chaque graphe vérifiant sa version de la définition 2 : « si un graphe s'obtient à partir d'un sommet isolé en y insérant des sommets pendants, c'est que le sommet isolé initial est relié à tous les autres sommets pendants insérés par la suite ». Ici, l'existence d'un chemin allant du dernier sommet ajouté vers le sommet initial pourrait faire l'objet d'une preuve par récurrence.

CONCLUSIONS ET PERSPECTIVES

Nous avons analysé des pratiques d'experts en lien avec la récurrence et la récursivité, afin d'envisager leurs conceptions sur ces notions. Notre analyse mathématique *a priori* s'est révélée utile pour montrer le rôle que peuvent jouer les notions ciblées dans l'activité de preuve en étude, ainsi que pour anticiper certaines stratégies que les chercheurs ont adoptées par la suite. Nous faisons l'hypothèse que les stratégies ayant émergé lors des analyses sont des invariants opératoires. De plus, nous avons relevé certains implicites qui pourraient également contribuer à éclairer la

conceptualisation. Nous nous appuyerons sur els éléments dégagés dans cette étude lors de la conception de séquences didactiques visant à faire travailler la récurrence et la récursivité et leurs interactions.

RÉFÉRENCES

- Artigue M. (2014) Didactic Engineering in Mathematics Education. In: S. Lerman (Eds.) *Encyclopedia of Mathematics Education*. Dordrecht, Pays Bas: Springer
- Durand-Guerrier, V. & Arsac, G. (2009) Analysis of mathematical proofs : Some questions and first answers. In: F.-L. Lin, F.-J. Hsieh, G. Hanna, M. de Villiers (Eds.) *Proceedings of the 19th ICMI Study Conference, Proof and Proving in mathematics education, vol 1* (pp. 148-153). Taipei, Taiwan : The Department of Mathematics, National Taiwan Normal University.
- Durand-Guerrier V., Meyer A., Modeste S. (2019) Didactical Issues at the Interface of Mathematics and Computer Science. In: G. Hanna, D. Reid, M. de Villiers (Eds.) *Proof Technology in Mathematics Research and Teaching. Mathematics Education in the Digital Era* (pp. 115-138). Cham, Suisse : Springer.
- León N. (2019) *An epistemological study of recursion and mathematical induction in mathematics and computer science*. Poster, CERME 11, Utrecht, Pays Bas.
- Leron, U. & Zazkis, R. (1986). Computational recursion and mathematical induction. *For the Learning of Mathematics*, 6(2), 25-28.
- Michaelson, M. T. (2008). A literature review of pedagogical research on mathematical induction. *Australian Senior Mathematics Journal*, 22(2), 57-62.
- Ouvrier-Bufferet, C. (2015). Modéliser l'activité de définition : vers de nouvelles perspectives en didactique. *Recherches en Didactique des Mathématiques*, 35(3), 313-356.
- Polycarpou, I. (2006). Computer science students' difficulties with proofs by induction: an exploratory study. In *Proceedings of the 44th annual Southeast regional conference* (pp. 601-606). Florida, États Unis : ACM
- Rinderknecht, C. (2014). A survey on teaching and learning recursive programming. *Informatics in Education-An International Journal*, 13(1), 87-120.
- Vergnaud, G. (2009). The theory of conceptual fields. *Human development*, 52(2), 83-94.
- Weber, K. & Alcock, L. (2004). Semantic and syntactic proof production. *Educational Studies in Mathematics*, 56(2-3), 209-234.
- Wilkerson-Jerde, M.H & Wilensky, U.J (2011). How do mathematicians learn math? Resources and acts for constructing and understanding mathematics. *Educational Studies in Mathematics*, 78(1), 21-43.