

VASSILIOS DAGDILELIS

La validation en programmation : à propos de conceptions des étudiants

Publications de l'Institut de recherche mathématiques de Rennes, 1989, fascicule S6
« Vème école d'été de didactique des mathématiques et de l'informatique », , p. 43-46

http://www.numdam.org/item?id=PSMIR_1989__S6_43_0

© Département de mathématiques et informatique, université de Rennes,
1989, 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/>

Vendredi 25 août 1989

Atelier : "La validation en programmation : à propos de conceptions des étudiants"

par Vassilios DAGDILELIS

Université J. Fourier, GRENOBLE

Introduction

La recherche qui a fourni le thème de l'atelier vise à l'identification et l'analyse des conceptions des étudiants relatives à la validation de programmes. Plus précisément, la partie de la recherche qui a été présentée à l'atelier concerne une préexpérimentation (compte tenu de la "faiblesse" de outils d'analyse alors disponibles), qui a été mise en place en 1988 avec la participation des 10 étudiants volontaires de l'Université de Grenoble I (niveau : deuxième année DEUG scientifique).

Les étudiants ont été répartis en cinq binômes avec une double tâche :

i) la construction, pendant une première phase, d'un algorithme pour la résolution d'un problème connu sous le nom "flot de télégramme" (énoncé : annexe II); et

ii) la "correction", pendant la deuxième phase, de l'algorithme pour le même problème, construit par un autre binôme (consigne : annexe I).

Quelques éléments épistémologiques

Une étude préalable, à caractère épistémologique, nous a permis de dégager certaines hypothèses quant au développement des moyens de validation en programmation. Nous avons plus précisément formulé l'hypothèse de l'existence d'au moins deux "univers" de programmation (celui des professionnels et celui des scientifiques) qui correspondent en fait à deux systèmes de contraintes et d'exigences et qui, par conséquent, déterminent deux ensembles de critères de validité. A partir de cette hypothèse nous avons fait un "retour" sur l'enseignement et la formulation d'un certain nombre de questions quant aux contraintes qui lui sont propres.

Sur le choix du problème

Le problème proposé aux étudiants, problème qui est devenu classique dans l'histoire de programmation, se place nettement dans "l'ambiance de programmation" habituelle (institutionnalisée dans l'enseignement en cause); mais la prise en compte de tous les "détails" du traitement rend son "découpage" non évident. Par ailleurs, la résolution du problème nécessite un effort quant à la détermination de la "nature" et la structuration des données sur lesquelles l'algorithme doit agir (qu'est-ce que c'est un télégramme ? un mot ?).

Ainsi la résolution du problème nécessite un certain nombre de décisions de la part du programmeur et par conséquent la mise en œuvre de divers procédés de contrôle de l'algorithme; par ailleurs, le fait que l'algorithme doit être corrigé dans un deuxième temps, rend nécessaire l'explicitation des choix qui ont été faits.

Validation, preuve et vérification de programmes

Nous avons repris l'hypothèse que la validation peut être considérée comme l'ensemble des processus qui visent à assurer la validité d'un énoncé ou d'un "objet" (ex. programme informatique), c'est à dire sa conformité à un certain nombre de règles. Nous avons essayé de montrer que, pour les étudiants, très souvent une des ces règles est que le produit final doit "marcher", c'est à dire c'est un mécanisme qui doit fonctionner correctement pour, au moins, une classe de données; ces données ne sont pas déterminées rigoureusement mais ont un caractère "général", elles ne sont pas des "cas limites".

Dans ce cadre, les étudiants s'assurent de la validité de leur produit par un double procédé de la façon suivante :

i) la construction de l'algorithme repose principalement sur une expérience mentale qui permet de décrire le traitement des données, de façon générale et générique au début mais de plus en plus détaillée par la suite. Certaines décisions, au cours de cette expérience, sont prises en fonction de la reconnaissance de "schémas" de traitement familiers (fortement institutionnalisés par l'enseignement); par ailleurs, la référence au familier, au "sens commun" a permis, dans certains cas, de trancher le débat.

La validité du produit réside, en quelque sorte, dans sa construction même.

ii) Si la tâche est complexe il se peut que le mécanisme produit ne fonctionne pas correctement : alors est mis en œuvre un certain nombre de tests (exécutions mentales ou simulations), de mécanismes de contrôle qui en cas d'erreurs fournissent des indications quant à la source des ces erreurs.

Notre hypothèse est que ces outils mis en œuvre dans les phases que nous venons de décrire ne sont pas, d'un point de vue "technique", fondamentalement différents : ce qui les distingue par contre c'est leur statut par rapport au projet des étudiants, c'est à dire par rapport à leur finalité; ainsi nous sommes amenés à penser que dans le premier cas ils peuvent être considérés comme de preuves tandis que dans le deuxième cas ils peuvent être considérés plutôt comme de vérifications (puisque leur rôle principal n'est pas de fournir des explications mais de corroborer la validité de l'algorithme ou de fournir des indications pour une rectification en cas d'échec).

Organisation pédagogique et temporelle de l'atelier

Une partie relativement importante de l'atelier (environ 20') a été consacrée à une brève présentation des hypothèses épistémologiques et une description de "l'ambiance" de programmation dans l'enseignement donné. Ce

choix était en fait une des conséquences de nos hypothèses sur le statut épistémologique de la programmation et le processus de sa didactification : pour un niveau scolaire donné, l'enseignement de la programmation n'obéit pas - pas encore - aux mêmes contraintes que les matières déjà "scolarisées" depuis longtemps (ex. les Mathématiques), phénomène qui exige une explicitation des spécificités de l'enseignement auquel on se réfère.

Le matériel distribué aux participants de l'atelier était constitué par :

i) l'énoncé du problème et les consignes aux étudiants (annexes I et II); et

ii) un certain nombre de protocoles (extraits des dialogues entre les étudiants). Ces protocoles ont été choisis en fonction des nos hypothèses sur la construction et validation de programmes.

A titre d'exemple, un premier extrait a été analysé après un travail d'environ 10' minutes, pour lancer ensuite l'étude des autres protocoles. La phase finale (environ 10') était un essai de présentation synthétique de nos hypothèses sur la validation présentées au paragraphe précédent (Validation, preuve et vérification de programmes).

ANNEXE I

La situation

On vous demande de résoudre un problème connu sous le nom "flot de télégrammes". Vous devez construire un programme pour résoudre ce problème et puis corriger un programme écrit par un de vos collègues. Vous allez travailler par binômes (deux personnes ensemble).

Il y aura donc deux phases:

- Pendant la première phase vous allez écrire un programme (pas nécessairement en PASCAL) pour le problème "flot de télégrammes". Dès que vous aurez fini, votre copie sera reprise par vos collègues, qui vont la corriger. Vous avez donc tout intérêt à fournir toutes les informations qui leur permettront de comprendre votre solution et d'être convaincus qu'elle est correcte.

Pour la résolution du problème vous n'allez utiliser qu'un seul stylo noir. Vous êtes priés de ne rien effacer. Vous pouvez, au début, travailler avec les feuilles "brouillon". La solution finale, avec les commentaires éventuels, sera écrite sur les papiers "programme" lesquelles seront distribuées à vos collègues.

Vous devez aussi signer vos papiers "programme" et vos collègues vont aussi signer après leur correction. Ainsi ils seront aussi responsables, avec vous, en ce qui concerne l'exactitude du programme.

- Pendant la deuxième phase votre copie sera corrigée par un de vos collègues. En même temps vous recevrez une copie à corriger. Après l'avoir corrigée vous devez aussi signer cette copie: vous serez ainsi co-responsables en ce qui concerne l'exactitude du programme corrigé par vous. Pour la résolution du problème vous n'allez utiliser qu'un seul stylo rouge.

ANNEXE II

Traitement d'un flot de télégrammes

On vous demande d'écrire un programme pour le traitement d'un flot de télégrammes, c'est à dire d'une suite de télégrammes.

Ce flot est disponible sur un ruban sous forme d'une suite de lettres et d'espaces.

Les mots de chaque télégramme sont séparés par un certain nombre d'espaces.

Le dernier mot de chaque télégramme est 'ZZZZ' et le flot entier se termine par un "télégramme vide", c'est à dire un télégramme qui ne contient d'autres mots qu'un 'ZZZZ' suivi d'un espace.

Chaque télégramme doit être traité pour déterminer le nombre de ses mots et pour signaler la présence d'au moins un mot qui est "très long", c'est à dire d'un mot qui a plus de 12 caractères. Cependant, aucun mot n'a plus de 20 caractères.

Les mots 'STOP' et 'ZZZZ' ne doivent pas être comptés, ni imprimés.

Le traitement qu'on vous demande consiste à imprimer chaque télégramme séparément, accompagné du nombre des ces mots et d'un message qui signale l'existence éventuelle d'au moins un mot "très long". Pour l'impression, les espaces redondants doivent aussi être éliminés.