ANALYSE DE STRUCTURES DE DONNÉES DYNAMIQUES ET HISTOIRES DE FICHIERS

P. FLAJOLET, C. PUECH

La théorie des histoires de fichiers permet d'analyser le coût de suites d'opérations portant sur des fichiers dont la taille varie avec le temps; elle permet donc de comparer, vis à vis de différentes mesures de cout, plusieurs structures de données dynamiques. Nous donnons ici un panorama de cette théorie: motivations, principaux resultats et applications.

1. INTRODUCTION

Avant de coder un algorithme dans un langage de programmation il est nécessaire de faire un choix sur la manière dont seront représentées, structurées et manipulées les données c'est à dire,de choisir une -ou plusieurs- "structures de données" (on entend par là -- une représentation d'un ensemble fini et -- une collection d'algorithmes de base opérant sur cet ensemble). Une bonne connaissance des structures de données existantes et des perfomances des algorithmes de base associés permet de faire le choix le plus approprié de la représentation des données en machine.

Pour mesurer l'efficacité d'un algorithme de base -et plus généralement d'un algorithme quelconque- on peut procéder de manière expérimetale et dérouler l'algorithme sur des -- données "text" (géneralemet prises "au hasard"). On peut aussi, et c'est l'approche qui nous intéressera ici, "analyser" de manière théorique l'algorithme, c'est à dire,calculer -exactement ou estimer asymptotiquement- le nombre d'opérations "elémentaires", (comparaisons, échanges, affectations, opèrations arithmètiques...), l'encombrement mémoire... ou tout autre paramètre significatif du coût d'exécution.

Les structures de données que nous considérons sont des structures dynamiques: leur -- taille varie au cours du temps sous l'effet d'adjonctions et de suppressions. Les listes chainées, les arbres,constituent de telles -- structures. Leur importance pour implanter de manière efficace des opérations sur des - dictionnaires, files de priorité... est bien connue /14/ , /1/ . Mais alors que l'analyse de structures de données statiques dont la taille n'évolue pas avec le temps a fait l'objet de travaux nombreux (analyses de problèmes de tri, analyses de parcours d'arbres considerés comme structures - statiques de représentation de programmes, - d'expressions...), les structures de données dynamiques ont eté jusqu'ici peu étudiées.

La comparaison de deux structures de données ou de deux implantations d'une même structure ne peut habituelllement se limiter à la - comparaison des coûts individuels de chaque algorithme de base; en effet, le plus souvent,une des structures est "meilleure" pour un des algorithmes, moins bonne pour un autre. C'est pourquoi il est naturel de considérer le comportement des structures sous -- l'effet de suites "quelconques" d'algorithmes de base.

Jean FRANGON, dans /11/ , a proposé une - notion d'"histoire de fichiers" qui permet - de modéliser le comportement de structures - (ou fichiers) évoluant "au hasard" et d'obtenir des informations sur le comportement moyen de telles structures. Cette notion - d'"histoire de fichiers" s'est révelée très riche et a donné lieu à plusieurs travaux -- /12/, /5/, /6/, /8/, /3/, /4/, /7/, /10/, -- dont le but du présent article est de donner un aperçu.

2. UN EXEMPLE

Nous présentons tout d'abord la démarche qui

Philippe Flajolet. INRIA. 78150 Rocquencourt (France)
Claude Puech. L.R.I., Université de Paris-Sud

conduit à la notion d'histoire de fichier--- sur un exemple simple.

Considérons un fichier dans lequel les opèrations permises sont l'adjonction d'un nouvel élément dans le fichier et la supression de l'enregistrement (ou clé) de plus petite valeur. On dit qu'un tel fichier est de type - "file de priórité"; on modélise par exemple, ainsi la file d'attente des utilisateurs --- d'un système informatique auquels le système d'exploitation attribue des ressources par - ordre de priorité; on verra dans la suite -- (cf. modèle avec réservoir borné) comment -- traiter le cas d'un ensemble fini d'utilisateurs "autorisés" de priorités données; nous supposerons ici que les priorités (les clés) sont extraites "au hasard" d'un ensemble ordonné deux (par exemple R).

Supposons, pour traiter un cas très simple, que l'on veuille comparer les performances - des listes triées et des listes non triées - pour implanter une telle structure (on verra que l'on peut analyser de manière analogue des implantations plus sophistiquées: tournois...). Si les liste sont des listes chaînées, il est raisonnable de mesurer le coût en temps des opérations d'adjonction et de suppression par le nombre de comparaisons -- entre clés effectuées car ces operations se ramènent essentiellement à de telles compa-- raison et à la mise à jour d'un nombre fixe de pointeurs. Les nombres de comparaisons entre clés necessaires pur faire une adjonction ou une suppression du minimum dans un fichier de taille k (c'est à dire contenant k clés) sont les suivants :

	-Adjonction-			-Suppression du Min-		
	min	max	moyenne	min	max	moyenne
liste non triée	0	0	0	1	k	$\frac{k+1}{2}$
liste triée	1	k+1	$\frac{k}{2}+1$	0	0	0

ce qui traduit de manière chiffrée le résultat bien évident: la structure de liste -- triée est meilleure pour l'extraction du minimum mais moins bonne pour l'insertion.

Pour disposer d'un critère quantitatif unique pour comparer les deux structures, on se propose d'étudier leur comportement sur une suite quelconque d'adjonctions (A) et de suppressions (S). Soit, par exemple, la suite d'opérations: $A(3.5)A(8.2)S_{min}A(2.3)S_{min}S_{min}$ chacune des opérations provoque des comparaisons de clés qui sont comptabilisées ci-dessous:

	A(3.5)	A(8.2)	S_{min}	A(2.3)	S_{min}	S_{min}
liste non triée	0	0	2	0	2	1
liste triée	1	1	0	1	0	0

donc, au total, 5 comparaisons pour une liste non triée, et 4 comparaisons pur une liste triée.

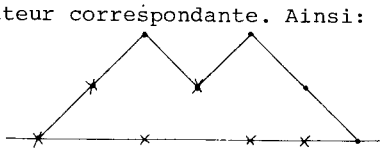
Sachant calculer le coût d'une suite quelconque d'opérations, on souhaite donner un sens à la notion de coût moyen d'une suite d'operations partant d'un fichier vide revenant - à un fichier vide, de longueur n (c'est à dire formée de n opérations). Supposons, comme on l'a dit plus haut, que les clés soient des nombres réels. Le point important est - de se ramener à un ensemble de référence fini. Pour ce faire, on remarque tout d'abord que le coût étudié ne dépend pas des valeurs exactes des clés mais seulement de -- leurs positions relatives; aussi, on considere comme équivalentes toutes les suites -- des mêmes opérations dans lesquelles les ordres relatifs des clés sont les mêmes (suites qui ont donc même coût), et l'on garde - comme représentation canonique d'une classe la suite obtenue à partir d'une suite quelconque de la classe en remplaçant chaque clé par son rang (compté à partir de 0, par exemple) dans l'ensemble des clés présentes dans la structure. Ainsi la classe d'équivalence de la suite d'opérations $A(3.5)\ A(8.2)\ S_{min}\ A(2.3)\ S_{min}\ S_{min}$ est représentée par: $A(0)\ A(1)\ S_{min}\ A(1)\ S_{min}\ S_{min}$ qui est appelée une "histoire" (ici de longueur 6).

L'ensemble des histoires de longueur n est fini: le nombre de possibilités d'adjonction d'une clé dans un fichier de taille k, c'est à dire le nombre de valeurs possibles du --- rang de la clé qu'on insère est en effet égal à k, et les suppressions opèrent toujours sur la plus petite clé. On peut donc calculer le "coût moyen d'une histoire" de longueur n et considérer que c'est le coût-moyen d'une suite de n opérations (le remplacement des valeurs des clés par leur rang, pour l'analyse est analogue à ce que l'on - fait, de façon classique, pour analyser les

algorithmes de tri (cf [K73]), où l'on remplace les n clés par une permutation de 1,2, ...,n).

Calculons, par exemple, le coût moyen des histoires de file de priorité de longueur 6, dans le cas des implantations liste non triée et liste triée envisagé plus haut.

Il est commode de representer graphiquement une histoire par un chemin valué dans le plan: le chemin traduit la suite des opérations (A S) mises en jeu dite schéma de l'histoire, une adjonction correspondant a une montée ("/") une suppression à une descente (" "); la valuation "marque" la suite des rangs des clés sur lesquelles elles opèrent (dite valuation de l'histoire) par des croix de hauteur correspondante. Ainsi:



est la representation de l'histoire A(0)A(1) S_{min}A(0)$S_{min}S_{min}$ de schéma A A S A S S et de valuation 0 00 .

Il y a 5 schémas possibles d'histoires de longueur 6. Pour chacun des schémas on a noté ci-dessous les valuations des histoires associées puis leur coût en nombre de comparaisons entre clés dans le cas des listes non triées et dans le cas des listes triées:

Schémas	Valuations	Coûts liste non triée	Coût liste triée
	000000	3	3
	000000	5	3
	000100	5	4
	010000	5	4
	010100	5	5
	000000	4	3
	010000	4	4
	000000	4	3
	000100	4	4
	000000	6	3
	001000	6	4
	002000	6	5
	010000	6	4
	011000	6	5
	012000	6	6

La distribution des coûts sur les histoires de files de priorité de longueur b est done la suivante:

coût	3	4	5	6
liste non triée	1	4	4	6
liste triée	5	6	3	1

et le coût moyen d'une histoire de file de priorité de longueur 6 est 5 pour une implantation liste non triée, 4 pour une implantation liste triée.

On peut démontrer plus généralement à l'aide des techniques exposées plus loin, qu'une histoire de file de priorité de longueur 2n a pour coût moyen $\frac{n(n+2)}{3}$ dans le cas d'une implantation liste non triée, $\frac{n(n+5)}{6}$ dans le cas d'une implantation liste triée, ce qui montre que la liste triée est, en moyenne, deux fois plus rapide quand n est grand.

3. LA NOTION D'HISTOIRE DE FICHIERS

Nous allons donner maintenant les definitions et hypothèses permettant de donner un sens précis à la notion d'histoire de fichier dans un cadre plus général.

Une structure de donnée (ou fichier) dynamique évolue avec le temps sous l'effet d'adjonctions, suppressions, modifications, fusions, éclatements, interrogations, réorganisations... Nous ne considérerons ici que des structures de données soumises aux opérations suivantes:

- A: adjonction d'un enregistrement (ou clé) dans le fichier.
- S: suppression d'une clé du fichier.
- Q: recherche d'une clé, ou interroga-

tion; nous serons parfois amenés à distinguer les recherches avex succès (Q^+) des recherches sans succès (Q^-).

Certaines des opérations mentionnés plus haut peuvent se ramener à ces opérations de base; d'autres demandent des analyses différentes.

Selon les modes d'utilisations de ces operations, on distingue classiquement les types de données suivants.

- Dictionnaires (D): accès aux clés par valeur sans restriction sur les opérations; ce type est classiquement implanté en memoire centrale sous forme de listes, triées ou non, ou d'arbres, equilibrés ou non; il est utilisé pour tenir à jour des tables d'identificateurs, des ensembles d'objets dans des fichiers...

- File de Priorité (F.P.); accès aux clés par valeur; opérations: adjonctions sans restriction, suppressions de la clé de valeur minimale; les files de priorité peuvent s'implanter comme les dictionnaires, mais il est avantageux de tirer partie des restrictions d'accès lors des suppressions en utilisant les structures d'arbre tournoi, tas, arbre pagode /13/ ou arbre binomial /16/ - elles sont utilisées lorsqu'on doit gérer sous forme de file d'attente (par exemple, comme on l'a vu plus haut, dans un système d'exploitation d'ordinateur) un ensemble d'individus obeissant à un système de priorités; on les retrouve dans les algorithmes de plus court chemin dans les graphes.

- Liste Linéaire (L.L.); accès aux clés par position; les seules opérations permises sont A et S sans restriction; ce type permet de tenir à jour des tableaux variables; il peut être implanté sous forme de listes chainées ou de tableaux, mais aussi, de manière plus efficace sous forme de tournois de position /17/.

- Table de Symboles (T.S.); cas particulier des dictionnaires où les suppressions; opèrent toujours sur la dernière clé insérée; seules les recherches avec succès sont autorisées; ces structures s'implantent comme des dictionnaires; l'absence d'interrogation sans succès est typique de l'organisation de la table des symboles de langages de type Algol, Pascal dont tous les identificateurs sont declarés; l'insertion est effectuée à l'entrée dans un bloc, la suppression à la sortie de ce bloc.

- Pile (P.): accès aux clés par position; opérations permises: adjonction et suppression restreintes à la clé qui se trouve en première position dans la structure (sommet de la pile); ces structures très simples s'implantent sous forme de listes chainées ou de vecteurs.

D'autres types de données (queues, structures de partition ...) sont intéressantes mais ne seront pas étudiées ici.

En supposant la structure de données initialement vide, une suite d'opérations de longueur n est une suite de la forme; $O_1(n_1)$, $O_2(n_2),\ldots,O_n(n_n)$ où pour i = 1,2,...,n, O_i est A,S,Q^+ ou Q^- et x_i une clé; l'opération O_i opére sur la structure vide. Nous supposerons que les opérations $O_1,O_2,\ldots,O_n$ sont telles qu'on ne rencontre pas de suppression ni d'interrogation positive appliquée au fichier vide (une telle condition est facile à formaliser).

Il est nécessaire de preciser de quels ensembles sont extraites les clés aux quelles s'appliquent les opérations qu'on considère. Nous envisagerons ici deux modèles differents:

1) le modèle avec réservoir borné; les clés sont les élements d'un ensemble fini ordonné, appelé réservoir initial, de cardinal N; une opèration d'adjonction au temps i (c'est à dire après la i^{cme} opération) consiste à enlever une clé du reservoir ($A(x_{i+1})$ n'est donc défini que si x_{i+1} est élement du reservoir au temps i...) et à l'ajouter à la structure; une opèration de suppression consiste, au contraire, à enlever une clé de la structure et à la remettre dans le réservoir; une opération d'interrogation ne modifie ni le reservoir, ni l'ensemble des clés de la structure (la clé peut cependant conduire à une reórganisation).

Dans le cadre de ce modèle, on appelle histoire de longueur n une suite d'operations de longueur n, verifiant les contraintes précédentes, et telle que le réservoir initial soit l'ensemble {1,2,...,N} (si ce n'est pas le cas ou renumérote arbitrairement les clés du réservoir par 1,2,...,N.

2) le modèle "markovien"; ici les clés sont les éléments d'un ensemble infini ordonné dense (par exemple R); si les seules opérations "élémentaires" portant sur les clés sont des comparaisons, deux suites d'opérations "isomorphes pur l'ordre" (c'est à dire que les uites des opérations elles mêmes sont identiques ainsi que les ordres relatifs des clés sur lesquelles elles portent) ont même coût en nombre de comparaisons; elles sont dites équivalentes.

À toute suite d'opérations $0_1(c_1), 0_2(c_2), \ldots, 0_n(c_n)$ on associe son histoire qui est $0_1(r_1), 0_2(r_2), \ldots, 0_n(r_n)$ où l'on a conservé la suite des opérations $0_1, 0_2, \ldots, 0_n$ (dite schéma de l'histoire) et où $r_1, r_2, \ldots, r_n$ (dite valuation de l'histoire) est la suite des rangs dans la structure, des clés sur lesquelles opèrent $0_1, 0_2, \ldots, 0_n$; par convention, ces rangs seront comptés à partir de 0. Deux suites d'opérations équivalentes ont même histoire et on peut considerer les histoires comme des représentants canoniques des classes d'équivalence de suites d'opérations.

Le modèle à reservoir borné et le modèle markovien seront les deux seuls modèles que nous envisagerons ici. On pourrait en étudier d'autres, par exemple, le "modèle de Kneth" pour lequel nous renvoyons à /12/.

Pour chaque type de donnée on peut définir une fonction de possibilités qui pour chaque opération licite sur la structure et pour chaque taille de fichier indique le nombre de façons différentes (du point de vue du modèle) de réaliser l'opération considérée.
Par exemple, dans le modèle à réservoir borné, lorsqu'on cherche à adjoindre une clé dans un fichier de taille k, il reste N-k clés dans le réservoir, et donc le "nombre de possibilités d'adjonction à niveau k" est N-k. On a rassemblé dans le tableau ci-dessous les valeurs des différentes fonctions de possibilité.

Types de données	D.	F.P.	L.L.	T.S.	P.	
pas(A,k)	k+1	k+1	k+1	k+1	1	Modèle markovien
pas(s,k)	k	1	k	1	1	
pas(Q^+,k)	k	0	0	k	0	
pas(Q^-,k)	k+1	0	0	0	0	
pas(A,k)	N-k	N-k	N-k	N-k	1	Modèle à réservoir borné (formule valables si $0 \leqslant k \leqslant N$; sinon pas(0,k)=0).
pas(s,k)	k	1	k	1	1	
pas(Q^+,k)	k	0	0	k	0	
pas(Q^-,k)	N-k	0	0	0	0	

pos (0,k) est le nombre de possibilités de au niveau k.

Pour simplifier les écritures nous posons pos (A,k) = a_k, pos (S,k) = s_k, pos (Q^+,k) = q_k^+, pos (Q_k^-, k) = q_k^-.

Notons qu'un type de données est, pour chaque modèle, entièrement défini par le quadruplet $(a_k, s_k, q_k^+, q_k^-)_{k \geqslant 0}$ appelé système de possibilités. Etant donné un systeme de possibilités, il est facile de calculer le nombre d'histoires ayant un schéma donné (le schèma d'une histoire $0_1(c_1), 0_2(c_2), \ldots, 0_n(c_n)$ est $0_1, 0_2, \ldots, 0_n$): le schéma de l'histoire suffit à définir la suite des tailles de la structure, dite suite des niveaux de h; ainsi pour une histoire de schéma A A S Q^+. A Q^- S S la suite des niveaux est (0,1,2,1,1, 2,2,1,0); le nombre d'histoires ayant pour schéma A A S Q^+A Q^- S S est donc $a_o\ a_1\ s_2\ q_1^+\ a_1\ q_2^-\ s_2\ s_1$; de façon générale, pour un schema $0_1, 0_2, \ldots, 0_n$ dont la suite des niveaux est $(k_0, k_1, \ldots, k_n)$, le nombre d'histoires admettant ce shema est pos $(0_1, k_0)$ pos $(0_2, k_1), \ldots$ pos $(0_n, k_{n-1})$.

En considérant tous les schémas possibles on peut calculer (on verra plus loin une autre façon de faire le calcul) le nombre d'histoires de longueur n, de niveau initial k, de niveau final l que nous noterons $H_{k,l,n}$ ($H_{k,l,n}$ dans le cas du modèle à réservoir borné si l'on veut faire apparaitre la taille du reservoir). Nous noterons $H_{k,l,n}$ l'ensemble des histoires verifiant les mêmes conditions. Comme on le verra les histoires partant du niveau 0 et retournant au niveau 0 font un rôle particulier et, pour simplifier, nous noterons $H_{0,0,n}$ par H_n et $H_{0,0,n}$ par H_n

4. COÛT MOYEN D'UNE HISTOIRE

Nous avons défini, dans les deux modèles envisagés, pour tout n, un ensemble fini de référence, l'ensemble H_n des histoires de longueur n, qui est formé, comme on l'a vu, soit de toutes les suites "licites d'opérations du longueur n, soit d'un échantillon "représentatif". Nous sommes en mesure de faire des analyses "en moyenne" des coûts des structures de donnée permettant d'implanter chaque type.

Nous considérons des fonctions de coût c définies pour toute opération $0(x)$, et additives sur les opérations, c'est à dire telles que le coût d'une suite $0_1(x_1), 0_2(x_2), \ldots, 0_n(x_n)$ d'opérations est: $C(0_1(x_1)) + \ldots + C(0_n(x_n))$. (C peut être comme dans l'exemple du §2 un nombre de comparaisons entre clés, ou le nombre de mises à jour d'une certaine variable, ou le nombre de cases-mémoire utilisées...).

À partir du coût d'une suite d'operations, et donc, en particulier, du coût d'une histoire, on peut définir le coût moyen d'une histoire ou coût integré par:

$$\overline{K_n} = \frac{1}{H_n} \sum_{h \in H_n} c(h) \quad (\text{où } H_n \text{ est le cardinal de } H_n)$$

expression qui dépend , bien entendu, à la fois de la structure de donnée et du modèle choisi.

Remarque 1: Le coût moyen que nous calculons prend en compte toutes les histoires de H_n affectées du même "poids", l'ensemble H_n dependant du modèle envisagé. Rappelons, en les interpretant, de manière un peu différente, les deux modèles envisagés: dans le modèle à reservoir borné, lors d'une adjonction, toutes les clés restant dans le réservoir ont la même "probabilité" d'être adjointes, et lors d'une suppression toutes les clés de la structure la même probabilité d'être supprimées; cette dernière hypothèse reste vraie pour lessuppressions dans le modèle markovien, mais pour les adjonctions dans un fichier de taille k contenant les clés $c_1, c_2, \ldots, c_k (c_1 < c_2 < \ldots < c_k)$, l'hypothèse est que chacun des k+1 intervalles $]-\infty, c_1[$ $]c_1, c_2[, \ldots,]c_k, +\infty[$ a la même probabilité de contenir la clé qu'on adjoint.

Remarque 2: Nous mettons l'accent ici sur l'analyse en moyenne mais on verra plus loin que les méthodes utilisées permettentaussi d'obtenir, par le biais de leurs moments, des informations sur les distributions de coûts.

Plutôt que de calculer les coûts intégrés à l'aide de la formule de definition, il est plus intéressant d'utiliser la formule suivante, dite formule du coût intégré:

$$K_n = \frac{1}{H_n} \sum_k [\, c(A,k) NA_{n,k} + c(S,k) NS_{n,k} + c(Q^+,k) NQ_{n,k} + c(Q^-,k) NQ^-_{n,k} \,]$$

où, pour toute opération 0; $c(0,k)$ est le "coût unitaire" de o au niveau k, $NO_{n,k}$ est le nombre total (pour toutes les histoires considerées) d'opérations 0 à niveau k, dit nombre de passages de l'opérations o à nive au k.

La formule du coût intégré est facile à démontrer si l'on suppose que le coût d'une opérations 0 sur une structure de taille k nedépend que de 0 et de k (ce coût ètant alor le coût unitaire de 0 au niveau k). Mais cette propriété très forte n'est vérifiée par aucune structure usuelle; par exemple pour un arbre binaire de recherche, le coût d'une insertion dépend à la fois de la forme de l'arbre (laquelle dépend des adjoctions et suppressions antérieures), et de la position de la clé insérée dans l'arbre. Toutefois, on peut démontrer /12/, /8/ que, pour une classe très générale de structures de données, dynamiques, dites structures stationnaires la formule du coût intégré reste vraie en prenant pour coût unitaire d'une opération 0 au niveau k le coût "moyen" d'effectuer l'opération o sur un fichier de taille k); les structures de liste simple, liste triée, arbres binaires de recherche, arbres tournois ... et de nombreuses structures sur lesquelles on n'impose pas des conditions d'équilibrage sont des structures stationnaires.

La formule du coût intégré permet de ramener l'évaluation des coûts intégrés à:

- -évaluer les coûts unitaires.
- -évaluer les nombres de passages à niveau k.
- -calculer les H_n c'est à dire dénombrer

Tableau 1. *Coût évalué en nombre de comparaisons des principales structures stationnaires; dans le cas des files binomiales,* $k = b_i 2^i$, $\nu(k) = b_i$ *et* $\sigma(k) = \frac{1}{k} \; i b_i 2^i$.

Type de donnée	Implantation	CA_k	CS_k	CQ_k^+	CQ_k^-
Dictionnaire	liste triée	$(k+2)/2$	$(k+1)/2$	$(k+1)/2$	$(k+2)/2$
	liste non triée	0	$(k+1)/2$	$(k+1)/2$	k
	arbre binaire de recherche	$2(K_{k+1}-1)$	$2(1+\frac{1}{k})H_k-3$	$2(1+\frac{1}{k})H_k-3$	$2(H_{k+1}-1)$
File de priorité	liste triée	$(k+2)/2$	0		
	arbre binaire de recherche	$2(H_{k+1}-1)$	0		
	tournois binaire	$H_{k+1}-1/2$	$2(H_k-2+1/k)$		
	pagode	$2(1-1/k+1))$	$2(H_k-2+1/k)$		
	queue binomial	$1+(k)-\nu(k+1)$	$\sigma(k)+\nu(k)-1-\nu(k-1)$		
Liste Lineaire	liste	$(k+2)/2$	$(k+1)/2$		
	tournois de position	$2(H_{k+1}-1)$	$2(1+\frac{1}{k})H_k-3$		

les histoires.

- Les coûts unitaires ont des coûts moyen qui se calculent par des méthodes "standard" /1/ . Le tableau 1 donne par exemple les coûts unitaires d'adjoctions, suppression, interrogaion pour différentes implantations classiques des files de priorité et des dictionnaires, dans le cas du modèle markovien. Les coûts sont là mesurés en nombres de comparaisons:

- L'évaluation des nombres de passage à niveau k se ramène à des dénombrements d'histoires. En effet, une histoire dans laquelle se produit l'opération 0 à niveau k,se décompose en une histoire partant du niveau 0 et allant au niveau k, l'opération 0 à la suite de laquelle la structure a pour taille l (l = k,k-1 ou k+1) et une histoire par tant du niveau l et retournant au niveau 0. On en déduit:

$$NA_{k,n} = \sum_{0 \leq i < a} H_{0,k,i} \cdot a_k \cdot H_{k+1,0,n-i-1}$$

$$NS_{k,n} = \sum_{0 \leq i < n} H_{0,k,i} \cdot s_k \cdot H_{k-1,0,n-i-1}$$

$$NQ^+_{k,n} = \sum_{0 \leq i < n} H_{0,k,i} \cdot q^+_k \cdot H_{k,0,n-i-1}$$

$$NQ^-_{k,n} = \sum_{0 \leq i < n} H_{0,k,i} \cdot q^-_k \cdot H_{k,0,n-i-1}$$

Il nous reste maintenant à voir comment dénombrer les histoires.

5. DÉNOMBREMENTS D'HISTOIRES ET FRACTIONS CONTINUES

Les dénombrements d'histoires peuvent se faire:

- soit par des méthodes "combinatoires géométriques"
- soit par des méthodes "algébriques".

Les méthodes géométriques developpées dans /11/, /12/ sont fondées sur une correspondance fondamentale (due à J. Frangon et G. Véemot) entre histoires et permutations qui ramène les décomptes d'histoires à des décomptes de classes de permutation, qu'on peut par ailleurs dénombrer directement. Nous ne développerons pas ici ces méthodes et renvoyons aux références citées.

La découverte par l'un des auteurs /5/ d'un lieu entre théorie des histoires et théorie des fractions continues a permis d'aborder par des méthodes algebriques les problèmes de dénombrement et de résoudre un certain nombre de problèmes non résolus par l'approche géométrique.

Le théoreme fondamental (prop 7 de /6/) établit une relation très générale entre histoires et fractions continues. Il s'agit d'une indentité entre la série caractéristique non commutative des histoires et une fraction continue. Cette identité, très ri

che, a de nombreux corollaires. Elle permet entre autres, d'obtenir, en particularisant:

(i) le nombre H_n d'histoires d'un type donné

(ii) le nombre $H_n^{[\leq h]}$ d'histoires de hauteur inférieure ou égale a h.

(iii) les nombres $H_{k,l,n}$ d'histoires de niveau initial k et niveau final l.

(iv) les nombres $H_{n/e/}$ d'histoires de longueur n de coût totale.

Les résultats (i) et (ii) permettent, comme on l'a vu, grâce à la formule du coût intégré de calculer le coût moyen d'une histoire. Quant au (iv) il permet d'aborder le problème de la distribution des coûts.

Signalons aussi qu'en dehors du contexte de l'analyse de structures de données dynamiques le lien entre fractions continues et histoires a également des conséquences combinatoires intéressantes /6/, /9/.

Plûtot que de donner la forme générale du théoreme reliant histoires et fractions continues, nous donnerons ici les formes particulières qui permettent d'obtenir les quantités (i),(ii),(iii),(iv).

(i) Le calcul du nombre d'histoires d'un type donné se fait en utilisant le theoreme:

Théorème Le nombre H_n d'histoires de longueur n pour un type de donnée de système de possibilités (a_k, s_k, q_k, q_k^-) vérifie:

$$\sum_{n\geq 0} H_n z^n = \cfrac{1}{1-q_0 z - \cfrac{a_0 s_1 z^2}{1-q_1 z - \cfrac{a_1 s_2 z^2}{1-q_2 z - \cfrac{a_2 s_3 z^2}{\cdots}}}}$$

(où $q_k = q_k^+ + q_k^-$)

Dans le cas du modèle markovien, on obtient pour les séries génératrices ordinaires $H(z) = \sum H_n z^n$ du nombre d'histoires de type dictionnaires, file de priorité, table de symboles, liste liniaire, pile, les developpements en fraction continue suivants:

$$DICT_{H(z)} = \cfrac{1}{1-1z- \cfrac{1^2z^2}{1-3z- \cfrac{2^2z^2}{1-5z- \cfrac{3^2z^2}{\cdots}}}}$$

$$FP_{H(z)} = \cfrac{1}{1- \cfrac{1z^2}{1- \cfrac{2z^2}{1- \cfrac{3z^2}{\cdots}}}}$$

$$TS_{H(z)} = \cfrac{1}{1-0.z- \cfrac{1z^2}{1-1z- \cfrac{2z^2}{1-2z- \cfrac{3z^2}{\cdots}}}}$$

$$LL_{H(z)} = \cfrac{1}{1- \cfrac{1^2z^2}{1- \cfrac{2^2z^2}{1- \cfrac{3^2z^2}{\cdots}}}}$$

$$PILE_{H(z)} = \cfrac{1}{1- \cfrac{z}{1- \cfrac{z}{1- \cfrac{z}{\cdots}}}}$$

Indiquons brièvement comment on en déduit les coefficients H_n:

- le cas des piles est trivial car la fraction continue vérifie l'equation fonctionelle

$$H(z) = \frac{1}{1-z^2 H(z)} \quad \text{d'où } H(z) = \frac{1-\sqrt{1-4z^2}}{2z^2}$$

et donc, comme on s'y attendant (les histoires de pile étant en bijection avec les arbres)

$$PILE_{H_{2n}} = \frac{1}{n+1}\binom{2n}{n}$$

- les fractions continues correspondant aux dictionnaires et aux files de priorité sont un cas particulier de la fraction continue de Gauss qui exprime la rapport deux séries hyporgéométriques contigües (cf.[W 67]), ce qui permet d'obtenir

$$FP_{H_{2n}} = 1.3.5. \ldots .(2n-1) \quad D_{H_n} = n!$$

- l'identification des fractions continues relatives aux listes linéaires et aux tables de symboles s'effectue grâce au théorème d'addition de Stieljes-Rogers (cf /18/) d'où l'on déduit:

$$\sum_{n\geq 0} LL_{H_{2n}} \frac{z^{2n}}{(2n)!} = \frac{1}{\cos z} \quad ;$$

$$\sum_{n\geq 0} TS_{H_n} \frac{z^n}{n!} = e^{e^z - 1 - z}$$

et donc:

$LL_{H_{2n}} = E_{2n}$ et $TS_{H_n} = \beta_n$, où E_{2n} est le 2n"ième nombre d'Euler et β_n le n"ième nombre de Bell 2-associé.

Dans le cas du modèle à réservoir borné le nombre de possibilités d'adjoction devient nul dès que le fichier contient toutes les clés du réservoir. Aussi les fractions continues se reduisent elles à des fractions rationnelles. On obtient, par exemple, pour les dictionnaires, les files de priorité et les liste linéaires les fractions rationnelles:

$$D_H/N/(z) = \cfrac{1}{1-Nz-\cfrac{N.1z^2}{1-Nz-\cfrac{(N-1)2z^2}{\cdots\cfrac{}{1-Nz-\cfrac{1.Nz^2}{1-Nz}}}}}$$

$$FP_H/N/(z) = \cfrac{1}{1-\cfrac{Nz^2}{1-\cfrac{(N-1)z^2}{1-\cfrac{(N-2)z^2}{\cdots\cfrac{}{1z^2}}}}}$$

$$LL_H/N/(z) = \cfrac{1}{1-\cfrac{N\,1z^2}{1-\cfrac{(N-1)2z^2}{\cdots\cfrac{}{1-\cfrac{2(N-1)z^2}{1-1Nz^2}}}}}$$

d'où l'on déduit:

$$D_H/N/(z) = \sum_j \frac{\binom{N}{j}}{1-2jz}$$

(et aussi:

$$\sum_{n\geq 0} D_H{}_n/N/\frac{z^n}{n!} = \left(\frac{e^{2z}+1}{2}\right)^N)$$

$$LL_H/N/ = \sum_j \frac{\binom{N}{j}}{1-(N-2j)z}$$

(et aussi

$$\sum_{n\geq 0} LL_H{}_n/N/\frac{z^n}{n!} = ch^N\, z)$$

$$FP_H/N/(z) = \frac{He_N(z)}{He_{NH}(z)}$$ (où $He_m(z)$ est le polynôme d'Hermite

$$\sum_r \frac{m!}{r!(m-2r)!}\left(-\frac{z^2}{2}\right)^r$$

et donc, par exemple:

$$LL_H{}_n/N/ = \frac{1}{2^N}\sum_j \binom{N}{j}(N-2j)^N$$

(ii) La correspondance entre histoires et fractions continues permet aussi de dénombrer très simplement les histoires de longueur n et de hauteur inférieure ou ègale à h (c.a.d. telles que la taille du fichier ne dépasse jamais h), que nous noterons $H_n^{[h]}$

Théorème : La série génératrice des histoires de hauteur inférieure ou égale a h est une fraction rationnelle qui est la $h^{ième}$. réduite de fraction continue des histoires de hauteur quelconque:

$$\sum_{n\geq 0} H_n^h z^n = \cfrac{1}{1-q_0z-\cfrac{a_0s_1z^2}{\cdots\cfrac{}{1-q_hz}}}$$

En utilisant les propriétés "classiques" des réduites des fractions continues, on obtient

Corollaire : Les histoires de hauteur inférieure ou égale à h ont une série génératrice rationnelle donée par:

$$H^{[h]}(z) = \frac{Ph(z)}{Qh(z)}$$

où P_h et Q_h sont les polynômes definis par les relations de récurrence:

$$\begin{cases} P_h = (1-q_hz)\,P_{h-1} - a_{h-1}s_hz^2P_{h-2} \\ P_{-1} = 0 \quad , P_0 = 1 \end{cases}$$

$$\begin{cases} Q_h = (1-q_hz)\,Q_{h-1} - a_{h-1}s_hz^2Q_{h-2} \\ Q_{-1} = 1 \quad , Q_0 = 1-q_0z \end{cases}$$

Les polynômes qui apparaissent ainsi dans l'expression de la série génératrice $H^{[h]}(z)$ sont des polynômes "bien connus'. En effet, les polynômes réciproques des Q_h, c'est à dire les polynômes $\overline{Q}_h$ définis par :

$$\overline{Q}_h(z) = z^{h+1}Q_h\left(\frac{1}{z}\right)$$ (Q_h est de degré h+1) sont des polynômes orthogonaux classiques de fonction génétracie simple (nous supposons ici que le modèle étudié est le modèle markovien pour lequel la fractions continue des histoires de hauteur quelconque est bien "infinie"). Le tableuau 2 donne pour chacun des types de données le nom de la famille de polynômes orthogonaux $\{\overline{Q}_h\}$ associée et sa fonction generatrice ordinaire ou exponentielle.

Tableau 2. *Nom de la famille de polynômes orthogonaux associés et fonction génératrice pour chaque type de données.*

DICT	Laguerre	$\Sigma \overline{Q_{k-1}} \frac{t^k}{k!} = \frac{1}{1+t} \exp\left(\frac{t}{1+t} z\right)$
F.P.	Hermite	$\Sigma \overline{Q_{k-1}} \frac{t^k}{k!} = \exp\left(-\frac{t^2}{2} + tz\right)$
L.L.	Meixner	$\Sigma \overline{Q_{k-1}} \frac{t^k}{k!} = \frac{1}{\sqrt{1+t}} \exp(\text{arctg } t.z)$
T.S.	Charbier	$\Sigma \overline{Q_{k-1}} \frac{t^k}{k!} = (1+t)^{z+1} e^{-t}$
P	Tchebicheff	$\Sigma \overline{Q_{k-1}} t^k = \frac{1}{1-zt+t^2}$

(iii) La connaissance de la série génératrice H(z) des histoires et des polynômes -- Q_k dénominateurs des réduites de la fraction continue associée permet aussi de dénombrer les histoires partant d'un niveau quelconque et arrivant à un niveau quelconque.

En effet (les notations sont celles de la -- fin du §3):

Théorème : (modèle markovien)

soit $H_{k,l}(z) = \sum_{m \geq 0} H_{k,l,n} z^n$. Alors

$$H_{kl}(z) = \frac{Q_{\mu-1}(z)}{a_0 a_1 \cdots a_{k-1} s_1 s_2 \cdots s_l \cdot a^{k+l}} (Q_{\lambda-1}(z) H(z) - P_{\lambda-1}(z))$$

où $\mu = \min(k,l)$ et $\lambda = \max(k,l)$

En particulier

$$H_{0k}(z) = \frac{1}{s_1 s_2 \cdots s_k z^k} (Q_{k-1}(z)H(z) - P_{k-1}(z))$$

$$H_{k0}(z) = \frac{1}{a_0 a_1 \cdots a_k z^k} (Q_{k-1}(z)H(z) - P_{k-1}(z))$$

Rappelons que les $H_{0,k,n}$ et $H_{k,0,n}$ intervien nent dans les formules donnant le nombre d'o pérations d'un type donné à niveau k et done dans la formule du coût-integré.

(iv) Enfin, la correspondance entre histoires et fractions continues permet d'obtenir des dénombrements plus fins : ceux des - histoires de coût donné.

Soit c une fonction de coût vérifiant les -- propriétés énoncées au début du §4. Si σ_k -- est le nombre de possibilités de l'opération 0 à niveau k, notons $c_q(\sigma_k)$ le polynôme en $u\, u^{c_0} + u^{c_1} + \ldots + u^{c_{\sigma_k - 1}}$ où $c_0, c_1, \ldots, c_{\sigma_k - 1}$ -- sont les coûts de chacune des σ_k possibili-- tés. La fractions continue permet de "mar-- quer" à l'aide de la variable supplémentaire u le coût c de chacune des opérations possi -- bles (par u^c) de la même manière que l'on -- marquait précedemmment la longueur n des his toires (par z^n) à l'aide de la variable z. On obtient:

Théorème:

$$\Sigma H_{n/\ell/} z^n u^\ell = \cfrac{1}{1 - c_u(q_0) - \cfrac{c_u(a_0)c_u(s_1)z^2}{1 - c_u(q_1) - \cfrac{c_u(a_1)c_u(s_2)z^2}{1 - c_u(q_2) - \cfrac{c_u(a_2)c_u(s_3)z^2}{\cdots}}}}$$

Voyons, par exemple, ce que donne ce théorème dans le cas de l'exemple étudié au §2. Le fichier considéré était de type file de - priorité donc $q_k = 0$, $a_k = k+1$, $s_k = 1$ puis qu'on s'était dans le modèle markovien.

Considérons le cas d'une implantation sans forme de liste triée, le coût de la suppression du minimum est 0 donc $c_u = (s_k) = 1$ le coût d'une adjonction à niveau k est 1', ou 2,..., on k + 1 (les coûts sont comptés en - nombres de comparaisons de clés) donc $c_u(s_k) = u + u^2 + \ldots + u^{k+1} = u^{[k+1]}$ où $[k] = 1+u+u^2+\ldots+u^{k-1}$ (quand u=1, [k] vaut k; on -- note traditionnellement la variable par q et on dit que $[k] = 1+q+\ldots+q^{k-1}$ est un q- -analogue de k; nous avons utilisé ici u, q ayant été employé avec un autre sens). D'après le théorème ci-dessus, on a donc pour la distribution des coûts:

$$\Sigma H_{n,/1/}\, z^n\, u^l = \cfrac{1}{1-\cfrac{u[1]z^2}{1-\cfrac{u[2]z^2}{1-\cfrac{u[3]z^2}{\ddots}}}}$$

Si l'on considère maintenant un dictionnaire implanté sous forme de liste triée, le coût d'une recherche sans succès peut être 1,2,... ou k + 1 (dans un fichier de taille k), le coût d'une recherche avec succès 1, 2... ou k, le coût d'une adjonction 1,2,... ou k + 1, le coût d'une suppression 1,2,... ou k; donc, $c_u(q_k^+) = c_u(q_{k-1}^-) = c_u(s_k) = c_u(s_{k-1}) = u[k]$, et:

$$\Sigma\, H_{n,/1/}\, z^n\, u^l = \cfrac{1}{1-u([0]+[1])z - \cfrac{u^2[1]^2z^2}{1-u([1]+[2])z - \cfrac{u^2[2]^2z^2}{1-u([2]+[3])z - \cfrac{u^2[3]^2z^2}{\ddots}}}}$$

Les fractions continues, qui se reduisent, quand u=1, à des fractions continues de Gauss, seront utilisées au §7 pour calculer les moments de la distribution des coûts.

Elles sont également intéressants du fait de leur interprétation combinatoire (cf /9/).

6. CALCULS DE COÛTS INTÉGRÉS: RÉSULTATS EXACTS ET ÉVALUATIONS ASYMPTOTIQUES

A l'aide de la formule du coût intégré 1, des formules qui donnent le nombre de passages à niveaux et des dénombrements d'histoires que nous venons de faire, nous sommes théoriquement en mesure de calculer les coûts intégrés des histoires.

Il est toutefois plus commode d'exprimer, au préalable, sous une autre forme les $H_{k,l,n}$, forme qui permet d'obtenir une expression élégante des coûts intégrés sous forme d'intégrale.

A chacun des types de donée on associe une forme linéaire sur l'ensemble des polynômes définie par $\langle x^n \rangle = H_n$ (où H_n est le nombre d'histoires du type considerè de longueur n) et par $\langle \lambda f + \mu q \rangle = \lambda \langle 1 \rangle + \mu \langle q \rangle$, et un produit scalaire défini par $\langle f | q \rangle = \langle f.q \rangle$. Vis à vis de ce produit scalaire, les polynômes $\overline{Q_k}$, polynômes réciproques des dénominateurs des réduites de la fraction continue associée au type de données consideré (ou, en d'autres termes au système de possibilités (a_k, s_k, q_k, q_k^-), vérifient des realtions d'orthogonalité:

Théorème (W 67)

$$\langle x^l | \overline{Q}_{k-1} \rangle = \langle \overline{Q}_{l-1} | \overline{Q}_{k-1} \rangle = 0 \quad \text{pour } 0 \leqslant l < k$$

$$\langle x^k | \overline{Q}_{k-1} \rangle = a_0, a_1, \ldots, a_{k-1}\ s_0, s_1, \ldots, s_k$$

Les relations permettent d'obtenir de nouvelles expressions des $H_{0,k,n}$ et $H_{k,l,n}$:

$$\cfrac{1}{1-u([0]+[1])z - \cfrac{u^2[1]^2z^2}{1-u([1]+[2])z - \cfrac{u^2[2]^2z^2}{1-u([2]+[3])z - \cfrac{u^2[3]^2z^2}{\ddots}}}} \qquad (1)$$

Théorème: Le nombre d'histoires finissant à hauteur k et ayant longueur n vaut:

$$H_{0,k,n} = \frac{1}{s_1 s_2 \ldots s_k} \langle x^n\, \overline{Q}_{k-1}(x) \rangle \quad \square$$

De manière analogue:

Théorème: Le nombre d'histoires commençant à hauteur k, finissant à hauteur ℓ, et ayant longueur n vaut:

$$H_{k,\ell,n} = \frac{1}{a_0 a_1 \ldots a_{k-1} s_1 s_2 \ldots s_\ell} \langle \overline{Q}_{k-1}(x) \overline{Q}_{\ell-1}(x)\ x^n \rangle \quad \square$$

En quoi les realtions d'orthogonalité sont elles utiles pour évaluer les coûts intégrés? Nous allons le montrer brièvement dans un cas simple: celui des piles, sous l'hypothèse markovienne. Comme, dans ce cas là, $a_k = s_k = 1$, $q_k = 0$, les polynômes $\overline{Q}_k$ verifient la relation: $\overline{Q}_k = z\,\overline{Q}_{k-1} - \overline{Q}_{k-2}$, avec $\overline{Q}_{-1} = 1$, $\overline{Q}_o = z$, d'où l'on tire immediatement que la série generatrice oridnaire associée $\overline{Q}(z,t) = \sum_{k \geqslant 0} \overline{Q}_{k-1}(z)\ t^k$ vaut:

$$\overline{Q}(z,t) = \frac{1}{1-zt\ \ t}$$

et donc:

$$\overline{Q}_{k-1}(z) = \sum_{i \geqslant 0} (-1)^i \binom{k-1}{i} z^{k-2i}$$

est un polynôme de Tchebicheff (résultat que nous avions annoncé au (iii) du §5). Calculons $\langle \overline{Q}(x,t) \rangle$ de deux manières différentes:

- d'une part

$$\langle \bar{Q}(x,t) \rangle = \left\langle \frac{1}{1-xt+t^2} \right\rangle = \frac{1}{1+t^2} \left\langle \frac{1}{1-x\frac{t}{1+t^2}} \right\rangle =$$

$$= \frac{1}{1+t^2} \Sigma \left\langle x^n \left(\frac{t}{1+t^2}\right)^n \right\rangle$$

et donc:

$$\langle \bar{Q}(x,t) \rangle = \frac{1}{1+t^2} \Sigma H_n \left(\frac{t}{1+t^2}\right)^n$$

- d'autre part

$$\langle \bar{Q}(x,t) \rangle = \sum_{k\geq 0} \bar{Q}_{k-1}(x) t^k = \sum_{k\geq 0} \bar{Q}_{-1}(x) \bar{Q}_{k-1}(x) t^k$$

par définition de $\bar{Q}_{-1}$, et donc, d'après les relations d'orthogonalité:

$$\bar{Q}(x,t) = 1$$

Posant $t = \frac{1-\sqrt{1-4z^2}}{2z}$, on en déduit:

$$\sum_{n\geq 0} H_n z^n = \frac{1-\sqrt{1-4z^2}}{2z}$$

résultat qui avait obtenu plus haut directement à partir de la fraction continue. Mais le méthode de démonstration ci-dessus permet d'obtenir bien d'autres résultats: ainsi en calculant de deux manières différentes ----- $\langle \bar{Q}(x,u)\bar{Q}(x,t) \rangle$ on obtient d'une part

$$\sum_{k,\ell} \langle \bar{Q}_{k-1}(x)\ \bar{Q}_{\ell-1}(x) u^k t \rangle = \frac{1}{1-ut}$$

et d'autre part

$$\frac{1}{1+t^2} \sum_{k,n} \langle \bar{Q}_{k-1}(x) x^n \rangle u^k \left(\frac{t}{1+t^2}\right)^n =$$

$$= \frac{1}{1+t^2} \sum_{k,n} H_{0,k,n}\ u^k \left(\frac{t}{1+t^2}\right)^n$$

d'après l'expression vue plus haut, de $H_{0,k,n}$ comme un produit scalaire. En identifiant les deux expressions et en posant

$t = \frac{1-\sqrt{1-4z^2}}{2z}$ on obtient:

$$\sum_{n,k\geq 0} H_{0,k,n}\ t^k z^n = \sum_{n,k\geq 0} H_{k,0,n}\ t^k z^n =$$

$$= \frac{1-\sqrt{1-4z^2}}{2z^2 - zt(1-\sqrt{1-4z^2})}\ , \text{ et donc}$$

$$H_{0,k}(z) = H_{k,0}(z) = z^k [B(z)]^{k+1} \quad \text{où}$$

$$B(z) = \frac{1-\sqrt{1-4z^2}}{2z^2} \quad (\text{et } H_{a,b}(z) = \sum_{n\geq 0} H_{a,b,n} z^n)$$

O_2 (on l'a vu à la fin du §4) le nombre d'adjonctions à niveau k (nombre total sur toutes les histoires) vaut:

$$NA_{k,n} = \sum_{0\leq i<n} H_{0,k,i} \cdot a_k \cdot H_{k+1,0,n-i-1} =$$

$$= \sum_{0\leq i<n} H_{0,k,i} \cdot H_{k+1,0,n-i-1}\ , \text{ donc:}$$

$$NA_k(z) = \sum_{n\geq 0} NA_{k,n} z^n = z\, H_{0,k}(z) \cdot H_{k+1,0}(z) =$$

$$= z^{2k+2} (B(z))^{2k+3}$$

Finalement, si $KA_n = \sum_k C(A,k) NA_{n,k}$ (partie du coût intégré relative aux adjonctions):

$$KA(z) = \Sigma KA_n z^n = \sum_{k\geq 0} CA_k\ z^{2k+2} (B(z))^{2k+3} =$$

$$= z^2 (B(z)^3\ CA(z^2 B^2(z)) \quad \text{où}$$

$$CA(t) = \sum_{k\geq 0} CA_k\ t^k$$

est la série génératrice des coûts unitaires d'adjonction pour une implantation de pile.

On peut traiter de la même manière les suppressions et obtenir:

Théorème: Les fonctions génératrices des coûts unitaires CA(t), CS(t) et des coûts intégrés KA(z), KS(z) pour les files sont liées par la transformation linéaire:

$$\begin{cases} KA(z) = z^2 B^2(z) CA(z^2 B^2(z)) \\ KS(z) = B(z) CS(z^2 B^2(z)) \end{cases}$$

où

$$B(z) = \frac{1-\sqrt{1-4z^2}}{2z^2}$$

Pour les autres types de données, on obtient par des méthodes analogues:

Théorème (métodes markovien) Pour des suites de références ω_ℓ dépendant du type de donnée les séries génétracies triples d'histoires:

$$\Sigma H_{k,\ell,n}\ \omega_\ell u^k v^\ell \frac{z^n}{n!} = \Xi(u,v,z)$$

ont des expressions donées par le tableau suivant:

	$\Xi(u,v,z)$	ω_ℓ
DICT	$\frac{1}{1-z(1+u)(1+v)-uv}$	1
FP	$e^{z^2/2}\, e^{zu+uv+vz}$	$\frac{1}{\ell!}$
LL	$\frac{1}{(1-uv)\cos z-(u\ v)\sin z}$	1
TS	$e^{e^z(1+u)(1+v)-z-u-v1}$	$\frac{1}{\ell!}$

Les séries génératrices des coûts s'obtiennent à partir des séries génératrices des coûts unitaires à l'aide de transformations intégrales comme il est indiqué dans les deux théorèmes suivants, (pour simplifier, on suppose ici que $CA_k=CS_k=CO_k=C_k$ on trouvera dans /8/ des enoncés plus généraux).

Théorème (modèle markovien) Soit $\{C_k\}_{k\geqslant 0}$ une famille de coûts unitaires d'opérations: C_k représente le coût d'une opération effectuée à niveau k . Soit C(u) la série génératrice des coûts unitaires $C(u) = \sum_k C_k\ \omega_k\ u^k$ où $\{\omega_k\}$ est une suite de référence qui dépend de la structure ($\omega_k = 1$ ou $\omega_k = \frac{1}{k!}$). La série exponentielle des coûts intégrés ------

$\hat{K}(z) = \Sigma K_n \frac{z^{n+1}}{n+1!}$ est reliée à la série des coûts unitaires C(u) par une transformation intégrale

$\hat{K}(z) = \mathcal{L}(C(u)\ ;\ z)$,

où la transformation $\mathcal{L}$ est donnée par la table suivante

	$\mathcal{L}$	ω_k
DICT	$\frac{2}{2-z}\int_0^{(\frac{z}{2-z})^2} \frac{C(u)\,du}{(1-u)((\frac{z}{2-z})^2-u)}$	1
FP	$2\, e^{z^2/2}\int_0^{z^2/4} e^{-u}C(u)\ \ z^2-4u\ du$	$\frac{1}{k!}$
LL	$\frac{2}{\cos z}\int_0^{tg^2\frac{z}{2}} C(u)\ \frac{du}{(1-u)^2 tg^2 z-4u}$	1
TS	$2\, e^{e^z-z-1}\int_0^{(e^{z/2}-1)^2} e^{-u}C(u)\ \frac{du}{(e^z+1-u)^2-4uz}$	$\frac{1}{k!}$

Théorème (modèle à réservoir borné) Pour les types de données liste linéaire et dictionnaires sur population de taille N, la série binômiale des coûts unitaires C(u) = $= \sum_{k\geqslant 0} C_k \binom{N}{k} u^k$, et la série exponentielle des coûts intégrés

$\hat{K}(z) = K_n \frac{z^{n+1}}{n+1}$ sont liées par la relation

$\hat{K}(z) = ch^N z\ \mathcal{L}^{/N/}[C(u);z]$ pour les listes linéaires

$\hat{K}(z) = (\frac{e^{2x}+1}{2})^N \mathcal{L}^{/N/}[C(u);z]$ pour les dictionnaires,

où la transformation $\mathcal{L}^{/N/}$ est définie par

$$\mathcal{L}^{/N/}[C(u);z] = \int_0^{th^2 z/2} \frac{C(u)}{(1\ u)^N (10u)} \frac{du}{\sqrt{(1\ u)^2 th^2 z-4u}}$$

Dans le cas des files de priorité (modèle markovien) l'expression intégrale peut être simplifiée par application d'une transformation de Laplace-Borel, et on obtient le résultat très simple:

Théorème (modèle markovien) Pour le type de données file de priorité, la série génératrice des coûts intégrés définie par

$\bar{K}(t) = \Sigma K_{2n} \frac{t^n}{n!}$

sont liées par la relation

$\bar{K}(t) = \frac{1}{\sqrt{1-2t}}\ \ \bar{C}(\frac{t}{1-t})$ où

$\bar{C}(u) = \sum_{k\geqslant 0} (CA_k + CS_{k\ 1})\, u^{k+1}$

Applications:

Il est clair à ce point qu'on dispose d'un mode de calcul effectif des coûts intégrés de toutes les structures stationnaires dont les coûts unitaires possèdent des séries génératrices de forme simple. Or, à l'exception des files binomiales, les coûts unitaires des structures usuelles sont des combinaisons linéaires des fonctions de k suivantes:

$1,\ k,\ k^2\ ;\ \frac{1}{k}\ ;\ \frac{1}{k+1};\ H_K = 1+\frac{1}{2}+\ldots+\frac{1}{k}$,

(voir au tableau 1 en fin de 4 les coûts unitaires d'adjonction, suppression, recherche, pour différentes implantations des types de données considerés)

Coûts intégrés des structures de dictionnaires.

	liste non triée	liste triée	arbre binaire de recherche
nombre de comparaisons	$\frac{1}{10}n^2+\frac{3n}{10}-\frac{7}{15}$	$\frac{n}{12}-\frac{3n}{4}+\frac{1}{6}$	$2nH_n-\frac{5}{2}n+0(\log^2 n)$

dont les séries génératrices ont les formes simples:

$$\frac{1}{1-x}\ ,\ \frac{1}{(1-x)^2}\ ,\ \frac{2-x}{(1-x)^3}\ ;\ \ell n\frac{1}{1-x}\ ;\ \frac{1}{x}\ \ell n\frac{1}{1-x}-x\ ;$$

$$\frac{1}{1-x}\ \ell n\frac{1}{1-x}$$

Les calculs, parfois fastidieux, se ramènent essentiellement à des déterminations de primitives de fonctions élémentaires -produits de logarithmes et de fonctions algébriques- puis aux développements en séries de Taylor. On obtient, par exemple, pour les structures de dictionnaires (modèle markovien);

Pour les files de priorité (modèle markovien) on obtient les coûts integrés du tableau 3

Les calculs de coûts intégrés de files de priorité implantées sous forme de files binomiales /16/ pour une description de cette structure) par les méthodes ci-dessus n'aboutissent pas, à cause de la forme particulière des coûts unitaires dans ce cas: si $k=\Sigma b_i 2^i$ est la décomposition binaire de k, $v_2(k)=\min\{i\,|\,b_i\neq 0\}$ et $\sigma(k)=\frac{1}{k}\Sigma i b_i 2^i$ les coûts unitaires d'adjonction et de suppression valent: ([B 78]) $CA_k=v_2(k)$, $CS_{k+1}=\sigma(k)-v_2(k)$, de sorte que $\overline{C}(u)=\Sigma(CA_k+CS_{k-1})u^k=\Sigma\sigma|k)\ u^k$ n'a pas d'expression simple.

On peut, toutefois obtenir une stimation asymptotique du coût intégré grâce au théorème suivant /3/, /4/:

Théorème Pour des coûts unitaires C_k "reguliers" les coûts intégrés de files de priorité vérifient:

$$\overline{K}_{2n}=n\int_0^{1/2} C_{[nt]}\frac{dt}{\sqrt{1-2t}}\ (1+0(\tfrac{1}{n}))$$

Ce théorème permet, dans tous les cas où nous avions sû calculer précédemment le coût intégré, d'obtenir très rapidement un equivalent assymptotique de ce coût ($c_k=1\Rightarrow\overline{K}_{2n}\sim n$ $C_k=k\Rightarrow\overline{K}_{2n}\sim\frac{n^2}{3}$, $C_k=1+\frac{1}{2}+\ldots+\frac{1}{k}$ $\overline{K}_{2n}\sim n\log_2 n$)

Il permet de plus d'analyser des implantations plus elaborées des files de priorité, comme les files binomiales. En effet, après

Tableau 3. *Coûts intégrés des files de priorité; par convention n?=1.3.5...2n-1*

Implantation	Coût integré
liste non triée	$\frac{n(n+2)}{3}$
liste triée	$\frac{n(n+5)}{6}$
arbre binaire de recherche	$\frac{2!}{n?}\left\{\sum_{1\leq i<n}\frac{i?}{i!}\left[\frac{i(2^{n-i}-i)}{2i-1}H_{n-i}+\frac{2^{n-1}-1}{n-i}\right]+\frac{2^n-1}{n}\right\}-2n$ $=n\ln n+0(n)$
tournoi binaire	$\frac{n!}{n?}\left\{\sum_{1\leq i<n}\frac{i?}{i!}\left[\frac{3i(2^{n-i}-i}{2i-1}H_{n-i}+5\frac{2^{n-i}-1}{n-i}\right]+5\frac{2^n-1}{n}\right\}-\frac{9n}{2}$ $=\frac{3}{2}n\ln n+0(n)$
pagode	identique aux arbres binaires de recherche

une étude fine de (k) (rappelons que C_k = = (k)) on peut prouver:

Théorème ([Ch 81]) Le coût intégré des histoires de file de priorite implantées sous forme de files binomiales vaut:

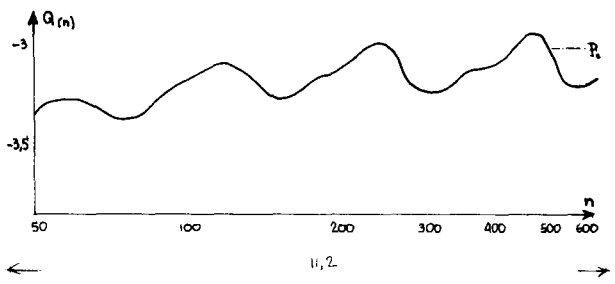
$$\overline{K}_{2n} = n \log_2 n + n\, P(\log_2 n) + o(n)$$

où P est une fonction continue, de période 1 dont on peut calculer les coefficients de -- Fourier.

Ce résultat est conforme aux valeurs des --- coûts intégrés que l'on peut calculer numériquement puisque, comme on le voit sur la figure ci-dessous la quantité

$$Q(x) = \frac{\overline{K}_{2n}}{n} - \log_2 n$$

au comportement quasi-périodique.



Signalons qu'on peut aussi obtenir une estimation asymptotique des coûts integrés des structures de type dictionnaire:

Théorème /3/ Pour des coûts unitaires "reguliers" les coûts integrés de dictionnaires verifient:

$$K_{2n} = n \int_0^{1/4} \frac{(1+2t)(CA_{[tn]} + CS_{[tn]}) + 2t\, CQ_{[tn]}}{\sqrt{1-4t}}\, dt \quad (1+0\,\tfrac{1}{n})$$

7. DISTRIBUTION DE COÛTS

Nous avons indiqué au §5(iv), comme dernière application de la correspondance entre histoires et fracitons continues, comment denombrer les histoires en fontion de leur coût; nous avons montré que, par exemple, pour des files de priorité implantées sous forme de - listes triées (modéle markovien) le nombre $H_{n/\ell/}$ d'histoires de longueur n de coût ℓ -- (compté en nombre de comparaisons de clés) est tel que:

$$\Sigma\, H_{n,/\ell/}\, z^n u^\ell = \cfrac{1}{1 - \cfrac{u[1]z^2}{1 - \cfrac{u[2]z^2}{1 - \cfrac{u[3]z^2}{\ddots}}}}$$

Nous allons voir maintenant, sur ce même --- exemple, comment on peut calculer la variance et les moments d'ordre supérieur de la - distribution des coûts (la moyenne de cette distribution n'est autre que le coût intégré des histoires qui vaut ici, on l'a vu à la - fin de §6, $\frac{n(n+5)}{6}$).

Remarquons tout d'abord que si l'on note:

$$\Sigma\, H_{n/\ell/}\, z^n u^\ell = \Sigma\, H_n(u)\, z^n$$

$H_{2n}(u)$ est un polynôme en u, de valuation n et de degré $\frac{n(n+1)}{2}$ car le nombre de comparaisons entre clés dans une adjonction à niveau k est au moins un et au plus k 1; les histoires de coût n étant celles pour lesquelles - toutes les adjonctions se font avec des clés de rang 0, il y en a autant que de schémas - de files de priorité ou encore d'histoires de pile c'est à dire

$$\frac{1}{n+1}\binom{2n}{n} \text{ et } H_{2n,/n/} = \frac{1}{n+1}\binom{2n}{n} \;;$$

par contre

$$H_{2n,\left/\frac{n(n+1)}{2}\right/} = 1:$$

il y a une seule histoire de coût maximal, l'histoire A(0),A(1),...,A(n-1)$S_{min}S_{min}$,..., S_{min} = Le fait que le coût considéré est quadratique dans le plus mauvais cas provient - de ce qu'ils s'agit d'un paramètre cumulatif (c'est le cas de toutes les fonctions de coût que nous considérons ici puisque nous supposons que le coût d'une suite d'opérations est la somme du coût des opérations).

Remarquons aussi que $H_{2n}(1)$ est le nombre total d'histoires de files de priorité de longueur 2n et vaut donc 1.3.5...(2n-1).

Calculer les moments de la distribution des coûts revient à calculer les valeurs au point 1 des dérivées successives de $H_{2n}(u)$.

On sait en effet que la distribution a pour:

-moyenne $\mu_{2n} = \dfrac{H'_{2n}(1)}{H_{2n}(1)}$

-variance $\sigma_{2n} = \dfrac{H''_{2n}(1)}{H_{2n}(1)} - \left[\dfrac{H'_{2n}(1)}{H_{2n}(1)}\right]^2 + \left[\dfrac{H'_{2n}(1)}{H_{2n}(1)}\right]$

...

La série génératrice $\Sigma\, H_n(u) z^n$ n'a pas d'expression analytique à l'aide de fonctions élémentaires (Touchard /15/ a montré qu'elle s'exprimait en termes de fonctions elliptiques). Aussi doit on essayer de déduire les propriété des polynômes $\bar{Q}_k(z,u)$ (polynômes réciproques des dénominateurs des réduites de la fraction continue), à savoir relations de récurrence et propietés d'orthogonalité, pour obtenir des propriétés des $H_{2n}(u)$.

Soit $\tilde{H}_{2n}(u) = u^{-n} H_{2n}(u)$. On a:

$$\Sigma\, \tilde{H}_{2n}(u) z^{2n} = \cfrac{1}{1 - \cfrac{[1]z^2}{1 - \cfrac{[2]z^2}{1 - \cfrac{[3]z^2}{\ddots}}}}$$

La série génératrice eulerienne des polynômes reciproques des dénominateurs de cette fraction continue

$$K(t,z,u) = \sum_{k\geq 0} \tilde{\bar{Q}}_{k-1}(z,u) \frac{t^k}{[1][2]\ldots[k]}$$

$$([r] = 1+u+u^2+\ldots+u^{r-1})$$

vérifie l'équation aux différences:

$$\frac{K(ut,z,u)-K(t,z,u)}{(u-1)t} = (z-t)K(t,z,u) \qquad (2)$$

(cette équation est une conséquence de la relation de récurrence vérifiée par les $\bar{Q}_{k-1}$; ces polynômes se réduisent aux polynômes d'Hermite quand u = 1; et alors l'équation aux différences se réduit à:

$\dfrac{\partial K}{\partial t} = (z-t)K$ qui a pour solution

$e^{-\frac{t^2}{2}+tz}$ qui est bien la série génératrice exponentielle des polynômes d'Hermite).

En ordonnant par rapport à z, notons:

$$K(t,z,u) = \Sigma\, \tilde{\bar{Q}}_{k-1}(z,u) \frac{t^k}{k!} = \Sigma\, R_n(t,u) z^n$$

En calculant de 2 manières différentes $\langle K(t,z,u)\rangle$ (comme on l'avait fait au §6), où la forme linéaire est ici définie sur l'espace des polynômes en z à coefficients polynômes en un par $\langle z^n\rangle = H_n(u)$, et en utilisant les propriétés d'orthogonalité des $\tilde{\bar{Q}}_{k-1}$, on obtient:

$$\Sigma\, R_n(t,u) H_n(u) = 1 \qquad (3)$$

En développant au voisinage de 1 l'équation au differences (2) on obtient

$\dfrac{\partial K}{\partial h}(t,z,1)$, $\dfrac{\partial^2 K}{\partial u^2}(t,z,1)\ldots$

d'où en extrayant le coefficient de z^n

$\dfrac{\partial R_n}{\partial u}(t,1)$, $\dfrac{\partial^2 R_n}{\partial u^2}(t,1)\ldots$

valeurs que l'on rapporte dans les équations obtenues en dérivant (3) par rapport à u et en faisant u = 1... équations qui permettent alor de déterminer successivement $\tilde{H}_{2n}(1), \tilde{H}'_{2n}(1), \tilde{H}''_{2n}(1)\ldots$

Revenant aux $H_{2n}(\psi)$ on obtient alors:

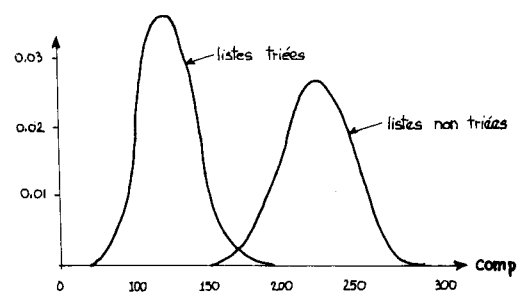
Theorème: La distribution du nombre de comparaisons entre clés dans les histoires de files de priorité sous forme de liste triée a pour caractéristiques:

$$\psi_{2n} = \frac{n(n+5)}{6} \qquad \sigma^2_{2n} = \frac{n(n-1)(n+3)}{45}$$

Nous renvoyons à /10/ où l'on trouvera ditaillée prévue qui n'a été qu'esquissié ci-dessus ainsi que d'autres analyses de distributions de coûts pour des implantations simples de files de priorité et de dictionnaires. On y montre en pariculier que les caracteristiques de la distribution du nombre de comparaisons pour les files de priorité implantées sous forme de liste non triée sont:

$$\mu_{2n} = \frac{n(n\ 2)}{3} \ , \ \sigma^2_n = \frac{n(n-1)(2n+1)}{45}$$

La figure ci-dessus montre les positions des deux distributions pour n = 25



BIBLIOGRAFIA

/1/ AHO A.,HOPOROFT J., ULLMAN S.,: *The Design And Anlysis of Efficient Computer Algorithms*, Addison-Wesly (1974).

/2/ BROWN M.R.: "Implementation and analysis of binomial queues algorithms," *SIAM J. Comput.*, vol. 7, n° 3 (1978), pp. 298--319.

/3/ CHENO L.: "Profils limites d'histoires sur les dictionnaires et les files de-- priorité; application aux files binomiales", Thése de 3eme Cycle, Orsay (1981).

/4/ CHENO L., FLAJOLET P., FRANÇON J., ---- PUECH C., VUILLEMIN J.: "Dynamic data-- structure Finite files, Limiting profiles and variance analysis", in 18th --- Allerton Conf., Monticello II, (1980), pp. 223-232.

/5/ FLAJOLET P.: *Analyse d'Algorithmes de - Manipulation d'Arbres et de Fichiers*,-- Thèse d'Etat, Faculté des Sciences d'Orsay (1979), et Cahiers du BURO (à paraître)

/6/ FLAJOLET P.: "Combinatorial aspects fo continued fractions" in Discrete Math. 32, (1980), pp. 125-161.

/7/ FLAJOLET P., FRANÇON J.,"Histoires de fichier en réservoir borné" in III Journées algorithmiques de Nice, (1980).

/8/ FLAJOLET P., FRANÇON J.,VUILLEMIN J.: "Sequence of operations analysis of dynamic data structures" in Journal of Algorithms 1 (1980), p. 207-216.

/9/ FLAJOLET P., PUECH C., VUILLEMIN J.: "Eulerian distributions of the symmetric group and the continued fraction of Heine", en préparation.

/10/ FLAJOLET P., PUECH C., VUILLEMIN J.: "On the distribution of costs in dyna-- mic data structures". en préparation.

/11/ FRANÇON J.: *Histoires de fichiers*, ---- RAIRO Info. Théor. 12 (1978), 49-67.

/12/ FRANÇON J., *Combinatoire des structures de données*, Thèse Fac. Sc. Strasbourg, juin 1979.

/13/ FRANÇON, VIENNOT, VUILLEMIN: "Description et analyse d'une représentation -- performante des files de priorité", Rapport Informatique Université Paris--Sud (1978).

/14/ KNUTH D.: *The Art of Computer Programming*, Addison-Wesley, vol. 1, 3 (1968--73).

/15/ TOUCHARD J.: "Sur un problème de configurations et sur les fractions conti -- nues", *Can. J. of Math.* 4 (1952) pp. 2--25.

/16/ VUILLEMIN J.: "A data-structure for manipulating queues", CACM, vol 21, n° 4 pp. 309-315, (avril 78).

/17/ VUILLEMIN J.: "A unifying look at data structures" CACM, vol 23, n° 4, pp.229--239 (1980).

/18/ WALL H.: *Analytic Theory of Continued Fractions*, Chelsea Pub. Co., New-York (1967) réed.