

# RAIRO

## INFORMATIQUE THÉORIQUE

JEAN FRANÇON

### **Histoires de fichiers**

*RAIRO – Informatique théorique*, tome 12, n° 1 (1978), p. 49-62.

[http://www.numdam.org/item?id=ITA\\_1978\\_\\_12\\_1\\_49\\_0](http://www.numdam.org/item?id=ITA_1978__12_1_49_0)

© AFCET, 1978, tous droits réservés.

L'accès aux archives de la revue « RAIRO – Informatique théorique » implique l'accord avec les conditions générales d'utilisation (<http://www.numdam.org/legal.php>). 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/>

## HISTOIRES DE FICHIERS (\*)

Jean FRANÇON (1)

Communiqué par R. Cori

---

*Résumé. — On définit une notion précise d'histoire de fichier puis on dénombre certains ensembles d'histoires de fichier qui se présentent naturellement; on utilise ces dénombrements pour calculer un cout moyen d'algorithmes intégré sur un ensemble d'histoires.*

### 1. INTRODUCTION

L'étude des performances des algorithmes relatifs aux structures de données se fait en général des deux façons suivantes : soit expérimentalement, en déroulant un algorithme (ou en simulant son déroulement) sur des données « prises au hasard »; soit théoriquement, en fixant par hypothèse l'ensemble des données admissibles par l'algorithme et en calculant (exactement si possible, ou des bornes, ou des estimations) les nombres d'opérations considérées comme élémentaires effectuées par un algorithme donné pour cet ensemble de données. Il est rare de voir des auteurs étudier une suite d'algorithmes, par exemple une suite d'adjonctions et de suppressions dans une structure de données; quand une telle étude est faite expérimentalement le choix des suites expérimentées est aussi « au hasard »; quand elle est théorique les auteurs imposent généralement des restrictions aux suites considérées, par exemple qu'il s'agit d'une suite formée alternativement d'adjonctions et de suppressions. L'étude la plus poussée en ce sens est celle de Knuth [4].

Le propos de cette note est de définir de façon mathématique une notion d'histoire de fichier, indépendante de la notion de structure de donnée, et d'en étudier quelques conséquences avec pour objectif l'étude des performances d'algorithmes. Une application immédiate est de fournir un cadre précis à la notion de « données prises au hasard » et de fichier « évoluant au hasard ». Une autre application est d'« intégrer » un calcul de performances sur un ensemble d'histoires.

De façon plus précise, nous sommes amenés à définir l'état d'un fichier à un instant donné comme étant un ensemble fini et à considérer que le fichier

---

(\*) Reçu septembre 1977.

(1) Centre de Calcul du C.N.R.S., Strasbourg et Institut de Recherche mathématique avancée, Strasbourg.

évolue au cours du temps sous l'action de quatre opérations : l'adjonction, la suppression, l'interrogation positive (l'article cherché est dans le fichier) et l'interrogation négative (l'article cherché n'est pas dans le fichier). Nous définissons alors une histoire de fichier comme, *grosso modo*, une suite de certaines de ces opérations en spécifiant pour chaque opération un certain ensemble de possibilités, par exemple que toutes les suppressions possibles sont permises, ou seulement la suppression de l'élément ayant la plus petite clé...

En fait, nous n'obtenons pas une seule notion d'histoire de fichier, mais plusieurs, selon les opérations considérées, selon qu'on fixe ou non la taille du fichier de départ ou d'arrivée, selon les ensembles de possibilités considérés pour chaque opération. Cette diversité correspond bien entendu à la diversité des hypothèses généralement faites dans ce domaine, hypothèses toujours « naturelles » en fonction d'un certain point de vue.

Ces définitions, accompagnées d'exemples, font l'objet de la section 2. A la section 3 nous dénombrons les histoires de fichier dans trois cas naturels; les formules obtenues sont remarquablement simples; cette simplicité est à mettre au compte d'un codage des histoires vérifiant certaines conditions par des permutations. L'objet de la section 4 est de mettre en évidence l'utilisation qui peut être faite des dénombrements de la section 3 pour obtenir des formules de performances intégrées sur un ensemble d'histoire. Une courte section de conclusions termine cette note.

## 2. DÉFINITIONS ET EXEMPLES

Dans la pratique, l'état d'un fichier à un instant donné est considéré comme un ensemble dont les éléments sont appelés données, ou articles, ou enregistrements, ou encore par d'autres noms propres à un genre particulier de fichier (par exemple une file de priorité est un fichier dont les éléments sont appelés des priorités, ou des événements dans certaines applications). Chaque élément est identifié par une clé.

Pour les besoins de notre propos, nous appelons *état de fichier* un sous-ensemble d'un certain ensemble  $K$  infini totalement ordonné, appelé *l'ensemble des clés possibles*. Cette définition revient à considérer tout élément d'un fichier comme réduit à sa clé et que toutes les clés d'un fichier sont distinctes. Considérer  $K$  infini et totalement ordonné n'est ici qu'une convention permettant de simplifier l'exposé; étendre les considérations qui suivent au cas d'un ensemble fini de clés possibles est une tâche qui reste à accomplir et qui présente un certain intérêt pour l'étude des structures de données et les algorithmes adaptés à cette situation (*voir* par exemple l'algorithme de tri dans [3], p. 78).

Un fichier évolue au cours du temps sous l'action d'opérations : adjonctions, suppressions, modifications, fusions, éclatements... Éventuellement il est soumis à des opérations qui n'influent pas sur son contenu comme une interrogation ou une réorganisation, mais qui influent dans certains cas sur la structure de données et les algorithmes utilisés.

Nous ne considérons ici que les quatre opérations suivantes :

(i) l'*adjonction* d'un nouvel élément au fichier, symbolisé ici par  $A$  et définie pour un état de fichier  $F \subset K$ , une clé  $x \in K \setminus F$ , par

$$A(x, F) = \{x\} \cup F;$$

(ii) la *suppression* d'un élément du fichier, symbolisée par  $S$  et définie pour  $F \subset K$ ,  $F$  non vide,  $x \in F$  par

$$S(x, F) = F \setminus \{x\};$$

(iii) l'*interrogation* (ou recherche) *positive* (ou avec succès) d'un élément dans le fichier, symbolisée par  $I^+$  et définie pour  $F \subset K$ ,  $F$  non vide,  $x \in F$ , par

$$I^+(x, F) = F;$$

(iv) l'*interrogation* (ou recherche) *négative* (ou sans succès) d'un élément dans le fichier, symbolisée par  $I^-$  et définie pour  $F \subset K$  et  $x \in K \setminus F$  par

$$I^-(x, F) = F.$$

On conviendra de réunir les deux interrogations en une seule notée  $I$  et définie pour toute clé et tout état de fichier.

Toutes ces opérations n'opèrent pas nécessairement de concert sur un fichier donné; par exemple sur la table des symboles d'un compilateur d'un langage sans variables locales n'opèrent que  $A$  et  $I^+$ , avec variables locales que  $A$ ,  $I^+$  et  $S$ ; sur une file de priorités n'agissent que  $A$  et  $S$ .

Les autres opérations signalées ci-dessus peuvent en général être ramenées à des suites d'opérations  $A$ ,  $S$ ,  $I^+$  et  $I^-$ . En particulier la modification d'un élément de fichier (par exemple dans un fichier de personnes la modification de la rubrique « appartenance politique » de M. Untel) peut être identifiée à une interrogation positive; tandis que la modification de la clé d'un élément dans un fichier trié dans l'ordre croissant des clés peut être identifiée à une suppression suivie d'une adjonction.

Remarquons aussi que ce qu'on entend dans la pratique par algorithme de suppression réalise tantôt l'opération  $S$ , tantôt l'opération  $I^-$  (quand on s'aperçoit que l'élément cherché n'est pas dans le fichier); de même un algorithme d'adjonction réalise tantôt l'opération  $A$ , tantôt l'opération  $I^+$  (quand on s'aperçoit que l'élément à ajouter est déjà dans le fichier).

Nous appelons *taille* d'un état de fichier  $F$  son cardinal que nous notons  $|F|$ . Nous disons qu'une suite d'opérations prises dans  $\{A, S, I^+, I^-\}$  est *bien définie pour un état initial*  $F$  si en appliquant la suite d'opérations à  $F$  on n'y rencontre pas de suppression ou d'interrogation positive appliquée au fichier vide. Enfin nous appelons comme Knuth [4] *organisation de données* une structure de données accompagnée d'un ensemble d'algorithmes qui opèrent sur elle.



Nous appelons *histoire de fichier de durée n*, pour  $n$  entier positif, un triplet  $H = (F_0, \Omega, X)$  où  $F_0$  est un état de fichier appelé *état initial* de  $H$ ,  $\Omega$  est une suite  $(\omega_1, \omega_2, \dots, \omega_n)$  d'opérations prises dans  $\{A, S, I^+, I^-\}$  bien définie pour  $F_0$ , appelée *schéma* de  $H$ ,  $X$  est une suite  $(x_1, x_2, \dots, x_n)$  de clés de  $K$  telles que, pour  $i = 1, 2, \dots, n$ , l'opération  $\omega_i(x_i, F_{i-1})$  est définie, en posant  $F_1 = \omega_1(x_1, F_0)$  et pour  $i > 1$   $F_{i-1} = \omega_i(x_{i-1}, F_{i-2})$ ; on appelle  $F_i$  l'*état de fichier* de  $H$  au temps  $i$  et  $F_n$  son *état final*.

Quand on veut étudier une organisation de données on est amené à poser des conditions sur les ensembles de clés auxquelles s'appliquent les opérations qu'on considère. Ces conditions sont souvent exprimées en termes de probabilités : par exemple en prenant pour  $K$  l'intervalle  $[0, 1]$  des réels, on dira que les clés prises comme entrées de l'adjonction sont choisies dans  $K$  suivant une distribution uniforme; ou bien, si  $k_1, k_2, \dots, k_m$  est la suite croissante des clés d'un état de fichier  $F$ , on dira qu'une clé  $k_{m+1}$  ajoutée à  $F$  a même probabilité de tomber dans l'un quelconque des  $m+1$  intervalles

$$[0, k_1[, ]k_1, k_2[, \dots, ]k_{m-1}, k_m[, ]k_m, 1].$$

On trouvera en particulier dans [4] de tels énoncés et une discussion de leurs conditions d'application.

Nous préférons ici formuler ces conditions de la façon suivante : à tout couple  $(\omega, F)$  où  $\omega$  est une opération et  $F$  un état de fichier, on associe un ensemble  $E \subset K$  tel que  $E \cap F = \emptyset$  si  $\omega \in \{A, I^-\}$  et  $E \subset F$  si  $\omega \in \{S, I^+\}$ , et on ne considère que les opérations  $\omega(x, F)$  avec  $x \in E$ ;  $E$  est appelé *ensemble d'entrée de  $\omega$  pour  $F$*  et l'entier  $\text{Card } E$  est appelé le *nombre de possibilités de  $\omega$  pour  $F$* .

Voici quelques exemples courants d'ensembles d'entrée.

*Exemple 1* : Si  $K$  est l'intervalle  $[0, 1]$  des réels, pour tout état de fichier  $F$  l'ensemble d'entrée de  $(A, F)$  et  $(I^-, F)$  est  $K \setminus F$ , celui de  $(S, F)$  et  $(I^+, F)$  est  $F$ .

*Exemple 2* :  $K$  étant l'ensemble des mots sur l'alphabet  $\{0, 1\}$ , les ensembles d'entrée sont définis comme dans l'exemple 1.

*Exemple 3* : Pour  $K$  quelconque (infini, totalement ordonné), pour un état de fichier  $F$  de taille  $r > 0$  dont la suite croissante des clés est  $(k_1, k_2, \dots, k_r)$ , l'ensemble d'entrée de  $A$  et  $I^-$  est  $\{k'_0, k'_1, \dots, k'_r\}$  avec

$$k'_0 < k_1 < k'_1 < \dots < k_r < k'_r$$

et pour  $r = 0$  l'ensemble d'entrée est  $K$ ; l'ensemble d'entrée de  $S$  et  $I^+$  est  $F$ . Nous appelons *fichier de genre liste* un fichier pour lequel toutes les opérations ont de tels ensembles d'entrée; ces conditions sont généralement employées pour les tables, les listes, triées ou non, ou encore les arbres binaires de recherche.

*Exemple 4* : Mêmes conditions que dans l'exemple 3 sauf que l'ensemble d'entrée de la suppression est réduit à un seul élément et qu'on ne considère pas d'interrogations.

Nous appelons *fichier de genre file de priorités* un fichier pour lequel toutes les opérations ont de tels ensembles d'entrée; c'est le cas généralement considéré pour les files de priorités (seules opérations : adjonction et suppression de la plus petite clé) et pour les fichiers où on effectue des suppressions « par l'âge » (*voir* [4]), comme celle du « premier entré », ou du « dernier entré ».

### 3. PROPRIÉTÉS DE QUELQUES ENSEMBLES D'HISTOIRES

Quand on veut étudier une organisation de données on est amené à dénombrer certains ensembles d'histoires. Nous allons considérer des histoires telles que, si  $p(\omega)$  est le nombre de possibilités d'une opération  $\omega$  sur un état de fichier  $F$ , on ait les inégalités – appelées ci-après *conditions C* :

$$p(A) \leq |F| + 1,$$

$$p(S) \leq |F|,$$

$$p(I^+) \leq |F|,$$

$$p(I^-) \leq |F| + 1.$$

Ces conditions sont vérifiées par les fichiers des genres liste et file de priorités.

Le but de cette section est de dénombrer des ensembles d'histoires vérifiant les conditions *C* et telles que toutes les histoires d'un même ensemble ont même état initial et même durée. Il est clair que le nombre d'histoires ayant même état initial et pour schéma une suite  $(\omega_1, \omega_2, \dots, \omega_n)$  donnée est égal au produit  $p(\omega_1) p(\omega_2) \dots p(\omega_n)$ ; on a donc à sommer de tels produits sur des ensembles de schémas. De tels dénombrements semblent *a priori* difficiles; cependant les conditions *C* vont nous permettre d'établir une bijection entre certains ensembles d'histoires et certains ensembles de permutations. Les dénombrements cherchés sont alors facilités et les résultats obtenus sont d'une simplicité surprenante.

Quelques définitions sont nécessaires avant d'énoncer un résultat de combinatoire des permutations et la bijection annoncée.

Pour  $n$  entier positif soit  $\mathfrak{S}_n$  l'ensemble des permutations de

$$[n] = \{ 1, 2, \dots, n \};$$

pour  $\sigma \in \mathfrak{S}_n$  posons par convention  $\sigma(0) = \sigma(n+1) = 0$ ; pour  $i \in [n]$  on dit que la lettre  $\sigma(i)$  est :

- (i) *pic* de  $\sigma$  si  $\sigma(i) > \sigma(i+1)$  et  $\sigma(i) > \sigma(i-1)$ ;
- (ii) *creux* de  $\sigma$  si  $\sigma(i) < \sigma(i+1)$  et  $\sigma(i) < \sigma(i-1)$ ;
- (iii) *double montée* de  $\sigma$  si  $\sigma(i-1) < \sigma(i) < \sigma(i+1)$ ;
- (iv) *double descente* de  $\sigma$  si  $\sigma(i-1) > \sigma(i) > \sigma(i+1)$ .

Soit  $P(\sigma)$  [resp.  $Q(\sigma)$ ,  $DM(\sigma)$ ,  $DD(\sigma)$ ] l'ensemble des pics (resp. creux, doubles montées, doubles descentes) de  $\sigma$ ; le quadruplet

$$(P(\sigma), Q(\sigma), DM(\sigma), DD(\sigma))$$

est appelé le *type* de  $\sigma$  et noté  $\tau(\sigma)$ .

Soit  $\mathbf{P} = (P, Q, DM, DD)$  une *partition ordonnée* de  $[n]$ , c'est-à-dire un quadruplet de parties de  $[n]$ , éventuellement vides, deux à deux disjointes et de réunion  $[n]$ . Nous définissons l'application  $\gamma = \Gamma(\mathbf{P}) : [n] \rightarrow \mathbf{Z}$  (ensemble des entiers relatifs) par

$$\begin{aligned} \gamma(1) &= 1 && \text{et pour } i = 1, 2, \dots, n-1, \\ \gamma(i+1) &= \gamma(i) - 1 && \text{si } i \in P, \\ \gamma(i+1) &= \gamma(i) + 1 && \text{si } i \in Q, \\ \gamma(i+1) &= \gamma(i) && \text{si } i \in DM \cup DD. \end{aligned}$$

D'autre part nous appelons *contraction* sur  $[n]$  une application  $c : [n] \rightarrow [n]$  telle que

$$\begin{aligned} c(1) = c(n) &= 1 && \text{et pour } i = 1, 2, \dots, n-1, \\ |c(i+1) - c(i)| &\leq 1. \end{aligned}$$

Nous sommes alors en mesure d'énoncer le théorème suivant, démontré dans [2] :

**THÉORÈME 1 :** *Soit  $T_n$  l'ensemble des couples  $(\Pi, f)$  vérifiant les conditions suivantes :*

- (i)  $\Pi = (P, Q, DM, DD)$  est une *partition ordonnée* de  $[n]$  telle que  $n \in P$  et  $\gamma = \Gamma(\Pi)$  est une *contraction* sur  $[n]$ ;
- (ii)  $f : [n] \rightarrow [n]$  est une *application dominée* par  $\gamma$ , c'est-à-dire qu'on a  $f(i) \leq \gamma(i)$  pour  $i \in [n]$ .

Alors il existe une bijection  $\theta : T_n \rightarrow \mathfrak{S}_n$  telle que pour tout  $(\Pi, f) \in T_n$  on ait  $\tau(\theta(\Pi, f)) = \Pi$ .

Cette bijection  $\theta$  a la propriété suivante : si  $j$  est un entier tel que  $j \in P$  et  $f(j) = 1$  [resp.  $f(j) = \gamma(j)$ ] alors, en posant  $\sigma = \theta(\Pi, f)$ , on a  $\sigma^{-1}(j) < \sigma^{-1}(k)$  [resp.  $\sigma^{-1}(j) > \sigma^{-1}(k)$ ] pour  $k = j+1, j+2, \dots, n$ .

Soit  $\mathfrak{S}'_n$  l'ensemble des permutations  $\sigma$  de  $\mathfrak{S}_{n+1}$  telles que  $\sigma(n+1) = n+1$ ; cet ensemble est évidemment de cardinal  $n!$ . On a le :

**COROLLAIRE 2 :** *Soit  $T'_n$  l'ensemble des couples  $(\Pi, f) \in T_{n+1}$  tels que  $f(i) \neq \gamma(i)$  pour  $i \in P \cup DD$ , avec  $\Pi = (P, Q, DM, DD)$  et  $\gamma = \Gamma(\Pi)$ . Alors  $\theta$  est une bijection de  $T'_n$  sur  $\mathfrak{S}'_n$ .*

Soit  $A_{2n}$  l'ensemble des permutations  $\sigma$  de  $\mathfrak{S}_{2n+1}$  qui sont sans doubles montées ni doubles descentes et telles que  $\sigma(2n+1) = 2n+1$ ; on sait depuis [1] que l'entier  $\text{Card } A_{2n}$  est le nombre sécant  $A_{2n}$  de série génératrice

$$\sum_{n \geq 0} A_{2n} \frac{x^{2n}}{(2n)!} = \frac{1}{\cos x}.$$

**COROLLAIRE 3 :** Soit  $T_n''$  l'ensemble des couples  $(\Pi, f) \in T'_{2n+1}$  tels que  $\Pi = (P, Q, \emptyset, \emptyset)$ ; alors  $\theta$  est une bijection de  $T_n''$  sur  $A_{2n}$ .

Soit  $I_{2n}$  l'ensemble des involutions (c'est-à-dire des permutations  $\sigma$  telles que  $\sigma^2$  est l'identité) de  $\mathfrak{S}_{2n}$  sans point fixe; on sait que  $\text{Card } I_{2n}$  est égal au produit  $1.3.5 \dots (2n-1)$ .

**COROLLAIRE 4 :** Soit  $T_n'''$  l'ensemble des couples  $(\Pi, f) \in T_n''$  tels que  $f(i) = \gamma(i)$  pour tout  $i \in Q$ , avec  $\Pi = (P, Q, DM, DD)$  et  $\gamma = \Gamma(\Pi)$ ; alors il existe une bijection  $\theta'$  de  $T_n'''$  sur  $I_{2n}$ .

Tous ces corollaires sont démontrés dans [2] ainsi que d'autres résultats, par exemple que le nombre de schémas permettant de définir les histoires de  $\mathfrak{S}_n$  (défini ci-dessous) est le nombre de Catalan :

$$\frac{1}{n+1} \binom{2n}{n}.$$

Revenons aux histoires de fichier et considérons l'ensemble  $\mathfrak{S}_n$  des histoires de fichier de genre liste, d'états initial et final vides et de durée  $n$  (donc telles que le nombre de possibilités de toute adjonction et de toute interrogation négative dans un état de fichier  $F$  est égal à  $|F|+1$ , de toute suppression et de toute interrogation positive est égal à  $|F|$ ). Dorénavant nous identifierons  $\mathfrak{S}$  et  $T'_n$  en vertu du :

**THÉORÈME 5 :** Il existe une bijection de  $\mathfrak{S}_n$  sur  $T'_n$ .

*Preuve :* Soit  $H = (\emptyset, \Omega, X) \in \mathfrak{S}_n$  avec  $\Omega = (\omega_1, \omega_2, \dots, \omega_n)$  un schéma et  $X = (x_1, x_2, \dots, x_n)$  une suite de clés. Posons

$$\begin{aligned} P &= \{i \in [n] : \omega_i = S\} \cup \{n+1\}, \\ Q &= \{i \in [n] : \omega_i = A\}, \\ DM &= \{i \in [n] : \omega_i = I^-\}, \\ DD &= \{i \in [n] : \omega_i = I^+\}. \end{aligned}$$

Le quadruplet  $\Pi = (P, Q, DM, DD)$  est évidemment une partition ordonnée de  $[n+1]$  et, pour  $\gamma = \Gamma(\Pi)$ , on a  $\gamma(i) = 1 + |F_{i-1}|$ , en posant

$$F_i = \omega_i(x_i, F_{i-1}), F_0 = \emptyset,$$

pour  $i = 1, 2, \dots, n$ .

Pour  $i \in [n]$  considérons l'ensemble  $E_i$  des entrées de  $\omega_i$ ; par hypothèse on a  $\text{Card } E_i = \gamma(i)$  si  $\omega_i \in \{A, I^-\}$  et  $\text{Card } E_i = \gamma_i - 1$  si  $\omega_i \in \{S, I^+\}$ ; en identifiant de façon arbitraire  $E_i$  à  $\{1, 2, \dots, \text{Card } E_i\}$  on peut identifier la clé  $x_i$  à un entier  $f(i)$  avec  $1 \leq f(i) \leq \gamma(i)$  si  $i \in Q \cup DM$  et  $1 \leq f(i) \leq \gamma(i) - 1$  si  $i \in P \cup DD$ . On a ainsi une bijection de  $\mathfrak{S}_n$  sur  $T'_n$ .

Q.E.D.

Du théorème 5 et du corollaire 2 on déduit immédiatement le :

**COROLLAIRE 6 :** *On a l'égalité*

$$\text{Card } \mathfrak{S}_n = n!$$

*Exemple :* Pour  $n = 4$  les quantités intéressantes sont données par le tableau suivant :

Schémas									
IIII	IIAS	IAIS	IASI	AIIS	AISI	ASII	ASAS	AASS	TOTAL
Suite des tailles du fichier									
00000	00010	00110	00100	01110	01100	01000	01010	01210	
Nombre de possibilités de l'opération									
1111	1111	1131	1111	1331	1311	1111	1111	1221	
Nombre d'histoires pour ce schéma									
1	1	3	1	9	3	1	1	4	24 = 4!

Le théorème 5 et le corollaire 2 permettent de « coder » tout ensemble d'histoires vérifiant les conditions  $C$  et de durée  $n$  par un ensemble de permutations de  $\mathfrak{S}'_n$ . Nous allons maintenant en déduire le dénombrement d'autres ensembles particuliers d'histoires. Tout d'abord un raffinement du corollaire 6 : pour  $n$  et  $k$  entiers, soit  $\mathfrak{S}_{n,k}$  l'ensemble des histoires de genre liste, de durée  $n$ , d'état initial vide et dont l'état final est de taille  $k$ .

**THÉORÈME 7 :** *On a l'égalité*

$$\text{Card } \mathfrak{S}_{n,k} = \binom{n}{k} n!$$

Nous allons en donner deux preuves : la première en utilisant le codage d'une histoire par une permutation, la seconde par un calcul direct à partir d'une relation de récurrence.

*Preuve 1 :* Par le théorème 5  $\mathfrak{S}_{n,k}$  est en bijection avec l'ensemble des couples  $(\Pi, f) \in T'_{n+k}$  tels que, si  $\Pi = (P, Q, DM, DD)$ , on ait

(i)  $n+1 \in P, n+2 \in P, \dots, n+k \in P;$

(ii)  $f(n+1) = f(n+2) = \dots = f(n+k) = 1.$

Par le corollaire 2 et la propriété de la bijection  $\theta$  (théorème 1), cet ensemble est en bijection avec l'ensemble des permutations de  $\mathfrak{S}'_{n+k+1}$  pour lesquelles les lettres  $n+1, n+2, \dots, n+k+1$  forment une sous-suite croissante de pics. La donnée d'une permutation de cet ensemble est équivalente à la donnée d'une permutation des lettres  $1, 2, \dots, n$  et des positions où insérer dans cette permutation les lettres  $n+1, n+2, \dots, n+k+1$  de façon à ce que deux de ces lettres ne soient pas contiguës et que  $n+k+1$  soit la dernière; il y a  $\binom{n}{k}$  choix possibles pour ces positions et  $n!$  choix possibles d'une permutation de  $1, 2, \dots, n$ , ces choix étant indépendants.

*Preuve 2 :* On vérifie facilement les égalités

$$\text{Card } \mathfrak{S}_{n,k} = 0 \quad \text{pour } k < 0 \text{ et } k > n,$$

$$\text{Card } \mathfrak{S}_{n,n} = n! \quad (\text{valable pour } n = 0 \text{ par convention}),$$

$$\begin{aligned} \text{Card } \mathfrak{S}_{n,k} &= k(\text{Card } \mathfrak{S}_{n-1,k-1} + \text{Card } \mathfrak{S}_{n-1,k}) \\ &+ (k+1)(\text{Card } \mathfrak{S}_{n-1,k+1} + \text{Card } \mathfrak{S}_{n-1,k}) \quad \text{pour } 0 \leq k < n. \end{aligned}$$

On vérifie alors trivialement que la solution de cette relation de récurrence est celle de l'énoncé.

Q.E.D.

Remarquons qu'il serait intéressant d'avoir des preuves directes, en termes d'histoires, des relations suivantes qui se déduisent trivialement du théorème précédent : pour  $k > 0$  :

$$\text{Card } \mathfrak{S}_{n,k} = n(\text{Card } \mathfrak{S}_{n-1,k-1} + \text{Card } \mathfrak{S}_{n-1,k}),$$

$$\text{Card } \mathfrak{S}_{n,k} = \frac{n-k+1}{k} \text{Card } \mathfrak{S}_{n,k-1},$$

$$\text{Card } \mathfrak{S}_{n,k} = \text{Card } \mathfrak{S}_{n,n-k},$$

$$\text{Card } \bigcup_k \mathfrak{S}_{n,k} = 2^n n!.$$

Soit  $\mathfrak{H}'_{n,k}$  l'ensemble des histoires de  $\mathfrak{H}_{n,k}$  sans interrogations (seules opérations permises :  $A$  et  $S$ ). Du corollaire 3 on déduit immédiatement le :

THÉORÈME 8 : *On a l'égalité*

$$\text{Card } \mathfrak{H}'_{2n,0} = A_{2n},$$

où  $A_{2n}$  est le nombre sécant.

On a évidemment aussi

$$\text{Card } \mathfrak{H}'_{n,n} = n!$$

$$\text{Card } \mathfrak{H}'_{n,k} = 0 \quad \text{pour } n+k \text{ impair et pour } k > n,$$

$$\text{Card } \mathfrak{H}'_{n,k} = k \text{ Card } \mathfrak{H}'_{n-1,k-1} + (k+1) \text{ Card } \mathfrak{H}'_{n-1,k+1}, \quad k > 0.$$

Mais nous n'avons pas trouvé d'expression simple pour  $\text{Card } \mathfrak{H}'_{n,k}$  dont voici la table des premières valeurs :

$n \quad k$	0	1	2	3	4	5	6	7
1	0	1						
2	1	0	2					
3	0	5	0	6				
4	5	0	28	0	24			
5	0	61	0	180	0	120		
6	61	0	662	0	1 320	0	720	
7	0	1 385	0	7 266	0	10 920	0	5 040

Soit maintenant  $\mathfrak{F}_{n,k}$  l'ensemble des histoires du genre file de priorités, de durée  $n$ , d'état initial vide et d'état final de taille  $k$  (donc les seules opérations sont  $A$  et  $S$  et  $S$  n'a qu'une possibilité).

THÉORÈME 9 : *On a les égalités*

$$\text{Card } \mathfrak{F}_{n,k} = 0 \quad \text{pour } n+k \text{ impair et pour } k > n,$$

$$\text{Card } \mathfrak{F}_{n,k} = \frac{n!}{2^j j!} \quad \text{pour } n-k \text{ pair et } 0 \leq k \leq n \text{ avec } j = \frac{n-k}{2}.$$

*Preuve :* Le corollaire 4 fournit  $\text{Card } \mathfrak{F}_{n,0}$ ; comme il est assez compliqué de donner la preuve générale en utilisant le codage d'une histoire par une permutation, voici une preuve directe. On vérifie facilement les égalités

$$\text{Card } \mathfrak{F}_{n,k} = 0 \quad \text{pour } k < 0 \text{ et } k > n,$$

$$\text{Card } \mathfrak{F}_{n,k} = k \text{ Card } \mathfrak{F}_{n-1,k-1} + \text{Card } \mathfrak{F}_{n-1,k+1} \quad \text{pour } n > 0 \text{ et } k > 0,$$

et on peut poser  $\text{Card } \mathfrak{H}_{0,0} = 1$ . On vérifie alors trivialement que la solution de cette récurrence est celle de l'énoncé.

Q. E. D.

Mentionnons pour terminer un résultat facile à démontrer (donc laissé au lecteur) et qu'on pourrait appeler « propriété d'invariance des histoires de genre liste par renversement du temps ». Soit  $\mathfrak{H}_{n,k,k'}$  l'ensemble des histoires de genre liste, de durée  $n$ , d'état initial de taille  $k$  et d'état final de taille  $k'$ ; soit  $H = (F_0, \Omega, X) \in \mathfrak{H}_{n,k,k'}$  de schéma  $\Omega = (\omega_1, \omega_2, \dots, \omega_n)$ , avec  $X = (x_1, x_2, \dots, x_n)$  et d'état final  $F_n$ . Posons  $\tilde{A} = S, \tilde{S} = A, \tilde{I}^+ = I^+, \tilde{I}^- = I^-$ . Considérons l'histoire  $\tilde{H} = (F_n, \tilde{\Omega}, \tilde{X})$  avec  $\tilde{\Omega} = (\tilde{\omega}_n, \tilde{\omega}_{n-1}, \dots, \tilde{\omega}_1)$  et  $\tilde{X} = (x_n, x_{n-1}, \dots, x_1)$ ; l'application  $H \rightarrow \tilde{H}$  est appelée *renversement du temps*.

**THÉORÈME 10 :** *Le renversement du temps est une bijection de  $\mathfrak{H}_{n,k,k'}$  sur  $\mathfrak{H}_{n,k',k}$ .*

On a aussi une propriété analogue pour les histoires de genre liste sans interrogation.

Du théorème 10 on déduit l'identité, pour  $n$  et  $n'$  entiers,

$$\sum_{k''} \text{Card } \mathfrak{H}_{n,k,k''} \text{Card } \mathfrak{H}_{n',k'',k'} = \text{Card } \mathfrak{H}_{n+n',k,k'};$$

en particulier on a

$$\sum_{k''} \text{Card } \mathfrak{H}_{n,0,k''} \text{Card } \mathfrak{H}_{n',k'',0} = \text{Card } \mathfrak{H}_{n+n',0,0} = (n+n')!$$

qui n'est rien d'autre que l'identité classique de van der Monde en vertu du théorème 7.

#### 4. PERFORMANCES D'ALGORITHMES INTÉGRÉES SUR DES HISTOIRES

Quand on veut comparer deux organisations de données on compare généralement les coûts des algorithmes qui réalisent les opérations effectuées sur les structures de données, ces coûts étant pris en un sens quelconque, par exemple temps d'exécution, nombre de comparaisons de clés...

Appelons *coût intégré sur une histoire* (ou *coût d'une histoire*) la somme des coûts des algorithmes qui réalisent les opérations du schéma de cette histoire; appelons *coût intégré sur un ensemble d'histoires* (ou *coût d'un ensemble d'histoires*) la somme des coûts de cet ensemble. Les résultats de la section précédente permettent de comparer expérimentalement des coûts intégrés sur un ensemble d'histoires en définissant un échantillon (en un sens statistique précis) de l'ensemble d'histoires et en mesurant les performances des algorithmes qui réalisent les opérations des schémas de cet échantillon.

La comparaison de coûts intégrés sur un ensemble d'histoires peut aussi se faire théoriquement par un calcul de coût intégré maximal ou de coût intégré moyen comme nous allons le voir sur le cas des histoires de genre liste sous des hypothèses simplificatrices.

Si  $\omega$  est une opération,  $x$  une clé,  $F$  un état de fichier, notons  $c(\omega(x, F))$  le coût (en général nombre réel non négatif) de l'algorithme qui réalise l'opération  $\omega$ . Si  $\mathfrak{H}$  est un ensemble d'histoires, notons  $c(\mathfrak{H})$  son coût.

Faisons maintenant une hypothèse — appelée dorénavant *hypothèse H* — permettant d'exprimer le coût intégré en fonction des quantités étudiées classiquement qui sont les coûts moyens d'adjonction, suppression et interrogation dans une structure de données de taille  $k$  obtenue après  $k$  adjonctions. Notons  $\overline{CA}(k)$  [resp.  $\overline{CS}(k)$ ,  $\overline{CI}^+(k)$ ,  $\overline{CI}^-(k)$ ] ce coût moyen d'une adjonction (resp. suppression, interrogation positive, interrogation négative). Par exemple, quand le coût est le nombre de comparaisons de clés, on a

(i) pour une liste non triée,  $\overline{CA}(k) = 1$  pour tout  $k$ ,

$$\overline{CS}(k) = \overline{CI}^+(k) = \frac{k}{2}, \quad \overline{CI}^-(k) = k;$$

(ii) pour une liste triée,

$$\overline{CA}(k) = \overline{CI}^-(k) = \frac{k(k+3)}{2(k+1)}, \quad \overline{CS}(k) = \overline{CI}^+(k) = \frac{k+1}{2};$$

(iii) pour une table triée (recherche par dichotomie possible) :

$$\overline{CA}(k) = \overline{CI}^-(k) = L+2 - \frac{2^{L+1}}{k+1}, \quad \overline{CS}(k) = \overline{CI}^+(k) = L+1 - \frac{2^{L+1} - L+2}{k},$$

avec  $L = \lceil \log_2 k \rceil$  (voir [3] p. 410).

Pour  $i \in [n]$ ,  $z \in \mathfrak{H}_n$ , soit  $F_i$  l'état de fichier au temps  $i$  de l'histoire  $z$  et soit  $k_i$  sa taille; soit  $(y_1, y_2, \dots, y_{k_i})$  la suite des clés de  $F_i$  dans l'ordre où elles ont été ajoutées au fichier (si une clé est ajoutée puis supprimée, puis ajoutée, etc. on considère sa dernière adjonction); l'hypothèse *H* est la suivante : le coût  $c(\omega_i(x_i, F_i))$  relatif à  $z$  est le même que le coût de  $\omega_i(x_i, C_i)$  où  $C_i$  est l'état de fichier obtenu après  $k_i$  adjonctions à partir du fichier vide des clés  $y_1, y_2, \dots, y_{k_i}$  dans cet ordre.

Cette hypothèse est discutée par Knuth [4] qui en formule une forme plus restrictive et plus précise. Knuth ne considère que les opérations d'adjonction et suppression; cependant les interrogations ne modifient pas, sauf exception, les structures de données et on peut donc étendre l'étude de Knuth aux histoires avec interrogation. Knuth fournit un certain nombre de cas où cette hypothèse est valable.

Sous l'hypothèse  $H$  on a

$$c(\mathfrak{S}_n) = \sum_k [NA(k)\overline{CA}(k) + NS(k)\overline{CS}(k) + NI^+(k)\overline{CI}^+(k) + NI^-(k)\overline{CI}^-(k)],$$

où  $NA(k)$  [resp.  $NS(k)$ ,  $NI^+(k)$ ,  $NI^-(k)$ ] est le nombre d'opérations  $A$  (resp.  $S$ ,  $I^+$ ,  $I^-$ ) dans un état de fichier de taille  $k$  au cours des histoires de  $\mathfrak{S}_n$ .

Ces dernières quantités sont faciles à calculer à partir du théorème 7. Soit  $NA(k, i)$ , pour  $0 \leq i \leq n-1$  le nombre d'histoires de  $\mathfrak{S}_n$  pour lesquelles la  $(i+1)$ -ième opération est une adjonction dans un fichier de taille  $k$ . On a évidemment :

$$NA(k) = \sum_i NA(k, i)$$

et

$$\begin{aligned} NA(k, i) &= \text{Card } \mathfrak{S}_{i, 0, k}(k+1) \text{Card } \mathfrak{S}_{n-1-i, k+1, 0} \\ &= \text{Card } \mathfrak{S}_{i, 0, k}(k+1) \text{Card } \mathfrak{S}_{n-1-i, 0, k+1} \end{aligned}$$

par le théorème 10; d'où on tire, par le théorème 7 :

$$NA(k, i) = (k+1) \binom{i}{k} i! \binom{n-1-i}{k+1} (n-1-i)!$$

pour  $n, k, i$  entiers (avec les conventions sur les coefficients binomiaux :

$$\binom{n}{k} = 0 \text{ pour } k < 0, \text{ pour } n < 0 \text{ et pour } k > n \geq 0).$$

On obtient de façon analogue  $NS(k)$ ,  $NI^+(k)$  et  $NI^-(k)$ . En posant

$$\bar{c}(\mathfrak{S}_n) = \frac{c(\mathfrak{S}_n)}{\text{Card } \mathfrak{S}_n},$$

on peut énoncer le :

**THÉORÈME 11** : *Le coût moyen  $\bar{c}(\mathfrak{S}_n)$  d'une histoire de  $\mathfrak{S}_n$  est donné par la formule*

$$\begin{aligned} \bar{c}(\mathfrak{S}_n) &= \frac{1}{n!} \sum_k [NA(k)\overline{CA}(k) \\ &\quad + NS(k)\overline{CS}(k) + NI^+(k)\overline{CI}^+(k) + NI^-(k)\overline{CI}^-(k)], \end{aligned}$$

avec

$$\begin{aligned} NA(k) &= (k+1) \sum_i \binom{i}{k} i! \binom{n-1-i}{k+1} (n-1-i)!, \\ NS(k) &= k \sum_i \binom{i}{k} i! \binom{n-1-i}{k-1} (n-1-i)!, \\ NI^+(k) &= k \sum_i \binom{i}{k} i! \binom{n-1-i}{k} (n-1-i)!, \\ NI^-(k) &= (k+1) \sum_i \binom{i}{k} i! \binom{n-1-i}{k} (n-1-i)!. \end{aligned}$$

## 5. CONCLUSION

Au point où nous sommes arrivés les tâches qui restent à accomplir sont claires :

(i) appliquer le théorème 11 à différentes fonctions de coûts, essayer de simplifier les formules obtenues, en faire une étude asymptotique;

(ii) même tâches pour d'autres ensembles d'histoires, par exemple  $\mathfrak{S}_{n,k}$ ,  $\bigcup_k \mathfrak{S}_{n,k}$ ,  $\mathfrak{S}'_n$ ,  $\mathfrak{F}_n$ ;

(iii) utiliser les résultats obtenus pour faire des études comparatives d'organisations de données.

Ce programme de travail est en cours de réalisation.

## REMERCIEMENTS

Ce travail a profité des remarques critiques de nombreuses personnes, en particulier Philippe Flajolet, Dominique Foata, Gérard Viennot et Jean Vuillemin.

## BIBLIOGRAPHIE

1. D. ANDRÉ, *Sur les permutations alternées*, J. Math. Pures Appl., vol. 7, 1881, p. 167-184.
2. J. FRANÇON et G. VIENNOT, *Sur la détermination des permutations par leurs pics, creux, doubles montées et doubles descentes* (à paraître).
3. D. E. KNUTH, *The Art of Computer Programming*, vol. 3, Reading, Mass., Addison-Wesley, 1973.
4. D. E. KNUTH, *Deletions that Preserve Randomness*, STAN-CS-76-584, Stanford University, décembre 1976.