


Une Histoire du Langage APL

par Sylvain Baron



Cet article reprend le contenu de notre conférence faite à la dernière réunion de l'AFAPL à Paris, le 15 juin 1995. Depuis un certain temps, nous étions quelques uns, connaissant et pratiquant l'APL depuis ses débuts (1969-1970 environ), à nous être aperçus, ce qui est bien normal, d'une disparition progressive de souvenirs partagés. Ce que nous connaissions tous d'expérience, et donc de mémoire, par notre pratique incessante des congrès, des rencontres et des lectures d'articles maintenant introuvables, n'était plus connu. Souvent même des faits étaient gravement déformés dans des écrits plus récents qui risquaient de servir de mémoire - incorrecte - aux nouveaux venus au langage. C'est pourquoi, sans faire une œuvre narrative d'historien, ni sans être complet très certainement, nous donnons ci-dessous, comme repère, une chronologie de faits s'appuyant tous sur des documents écrits, dont tous ceux qui sont cités sont en notre possession et peuvent être consultés sur simple demande.

Nous devons aussi signaler que nous nous restreignons dans cet article aux dates et aux faits marquant l'évolution du **langage** APL en tant que notations, avec mention explicite, quand il le faut, des matériels et des systèmes d'exploitation qui lui ont permis de fonctionner sur un ordinateur, et donc d'être « un langage de programmation ». Nous ne donnerons ici aucune indication historique sur l'évolution de l'**usage** d'APL. Nous ne le ferons ni en ce qui concerne le nombre de programmeurs ou d'utilisateurs finals, ni en ce qui concerne les types d'applications, non pas que ce soit un travail plus difficilement réalisable, mais parce que nous entrerions là dans un domaine qui, à notre avis, se doit d'être accompagné d'analyses et d'explications. Nous pensons que ce volet serait plus à même de faire l'objet d'une table ronde lors d'une réunion de notre association.

Enfin, il est difficile de ne pas citer de noms. Dans ce cas, les oubliés sont plus nombreux que les cités. Nous connaissons des dizaines de contributions écrites citées par des dizaines d'auteurs, dont très souvent Ken Iverson, mentionnant les contributions de chacun, soit dans la reconnaissance d'idées cachées et ensuite exploitées, soit dans l'innovation dans le domaine des fonctions primitives, des opérateurs ou de l'architecture du langage, aussi bien par réflexion théorique que par contrainte dans la programmation des systèmes APL. Nous espérons néanmoins que nos citations de noms restent celles qui s'imposent et que l'essentiel sera ainsi correctement situé.

I - La période académique

En 1950, Ken Iverson obtient un « B.A. degree in Mathematics » à la Queen's University, à Kingston (CANADA), puis, en 1951, un « M.A. degree in Mathematics », à l'université de Harvard, aux États-Unis, suivi en 1954 d'une thèse (Ph. D.) en mathématiques appliquées, toujours à Harvard.

En 1955, le professeur Howard Aiken (constructeur en 1940 du premier ordinateur -le MARK 1- avec le financement d'IBM) prend Ken Iverson comme « Assistant Professor » à l'université de Harvard. On lui demande d'enseigner un cours de Traitement automatique de données établi par le professeur Aiken.

En 1956, débute la rédaction du livre « Automatic Data Processing » avec F.Brooks. C'est au cours de cette période que naissent les principes et une part considérable des notations qui conduiront au langage APL. En 1960, devant l'ampleur du manuscrit, l'éditeur John Wiley suggère de couper le livre en deux, afin de réserver le premier livre aux notations d'Iverson. A ce moment, en 1960, Ken Iverson quitte l'université pour entrer chez IBM, au Centre de Recherche Thomas Watson, à Yorktown Heights, dans l'état de New York.

L'éditeur poursuit son travail et les deux livres sont publiés les années suivantes:

En 1962, A Programming Language
 Iverson K.E.
 John Wiley&Sons,

et en 1963, Automatic Data Processing
 Brooks, F.P. Jr. and Iverson, K.E.
 John Wiley&Sons

Lorsqu'il entre chez IBM en 1960, Ken Iverson fait la connaissance de Adin D. Falkoff entré en 1955 au Thomas Watson Research Center d'IBM. Les recherches de Falkoff se prêtent bien aux notations d'Iverson qui sont pleinement utilisées dans un article de grande importance publié en 1962 : « Algorithm for parallel-search Memories », Adin Falkoff, Journal of the ACM, Oct.1962,9,n°4.

1964 est une année fertile : Ken Iverson donne une conférence « Formalism in Programming Languages » ACM, Volume 7, number 2, February 1964 (Reproduite en 1981 dans « A source Book in APL », APL Press, Palo Alto) d'une richesse exceptionnelle.

Puis, dans l'IBM Systems Journal (Vol.3, N°3, 1964), Falkoff, Iverson et Sussenguth publient, en notations d'Iverson « A formal description of SYSTEM/360 », dans le même journal où F.Brooks, G.Blaauw et G.Amdahl, les pères de l'IBM 360, décrivent la structure de ce système. Cet article a eu à l'époque un fort impact, car il décrivait, en 20 pages à peu près, le système 360 dont la description en anglais prenait des milliers de pages. On doit aussi à ce travail une primitive : le tirage au hasard (?).

Enfin, c'est en 1964 qu'est né le projet d'implantation des notations sur ordinateur et qu'un progrès considérable fut fait au cours d'un travail de plusieurs mois par Falkoff, Iverson et McDonnell pour la mise au point de la « boule » APL, conçue pour le terminal à boule amovible qu'IBM venait de créer : l'IBM Selectric 1050. Ce travail eu pour effet essentiel de diminuer le nombres de symboles et d'adopter une typographie linéaire.

En 1965, Larry Breed est recruté pour le développement du système. Il fait détacher auprès de lui son ami Roger Moore qui venait d'entrer chez I.P.SHARP Associates (IPSA) pour réaliser la partie time-sharing du système APL\360. Larry Breed et Phil Abrams débute par une implantation des notations sur IBM 7090 (le système est connu sous le nom d'IVSYS: Iverson System)

En 1966, Dick Lathwell, Alan Rose et Paul Berry rejoignent l'équipe et, à l'automne, le système APL\360 (version nommée XM1) fonctionne pour la première fois. Entre temps, il avait fallu trouver un nom au langage comme au système. L'acronyme APL fut proposé par Adin Falkoff en juillet 1966 et immédiatement accepté (cf. l'introduction d'Eugene McDonnell au « Source Book of APL » déjà cité).

En 1967, un APL\1130 est développé pour les ordinateurs « universitaires » IBM-1130. A.Falkoff y avait vu la possibilité d'utiliser cette version d'APL pour prolonger la vie de ces machines et donc, grâce à cette « bonne » raison, la possibilité de faire sortir l'APL du laboratoire. Cette même année, Mike Jenkins code le domino (division et inversion de matrice) et Trenchard More (futur théoricien des tableaux généralisés) se joint au groupe.

En 1968, l'APL\360 dans sa version XM6 est terminé et Roger Moore retourne chez I.P. SHARP qui obtient l'autorisation de commercialiser un service APL en time-sharing. Pour ce faire I.P. SHARP acquiert un ordinateur IBM 360/50.

En 1969, l'impatience est grande au sein de l'équipe de Falkoff de voir IBM développer un système de fichiers pour APL et, surtout, de commercialiser APL. Larry Breed quitte IBM pour la toute nouvelle STSC (aujourd'hui Manugistics) qui partage l'IBM 360/50 d'I.P. SHARP pour commercialiser ce time-sharing APL aux États-Unis. Larry Breed (STSC) et Eric Iverson (IPSA) développèrent un système de fichiers commun pour les deux sociétés. Alan Rose quitte aussi IBM pour STSC. Cette même année, Jim Brown rejoint en tant qu'externe le groupe de Falkoff à IBM où il prépare sa thèse sur une généralisation d'APL.

Fin 1969 voit enfin le début de la commercialisation de l'APL\360 XM6 (sans système de fichiers) par IBM.

II - La période informatique

1 - L'unité

L'apparition de systèmes APL sur différents types d'ordinateurs (chez IBM d'abord, puis peu à peu chez les autres constructeurs) a eu pour effet de voir s'accroître considérablement le nombre des utilisateurs et donc de rendre plus difficiles les évolutions du langage et des systèmes. Car on se devait alors de prendre en compte « la base de clientèle installée » et donc d'offrir des évolutions « compatibles vers le haut ». Mais on est encore pour APL dans une période où le langage reste le fait d'un très petit nombre de personnes qui fonctionnent par consensus. Cette caractéristique, liée à la qualité des personnes, fait que les évolutions sont parfaitement mûries et parfaitement en phase avec un produit déjà étonnamment réussi au sortir de sa longue période académique (1956-1969).

Notons qu'en 1970 toute l'équipe APL quitte la Division Recherche de Yorktown Heights qui clot le Projet APL et s'installe au centre scientifique IBM de la « Data Processing Division » récemment ouvert à Philadelphie.

En 1971, Dick Lathwell débute une investigation de l'idée des variables partagées qui étaient incidemment apparues comme un moyen d'échange entre processus dans le livre de 1962 et l'article de l'IBM System Journal de 1964. Les variables partagées conduiront à la variété des communications par processeurs auxiliaires que nous connaissons dont TSIO (Time Sharing Input Output) fut le premier exemple pour résoudre la question de la communication avec le système d'exploitation OS-MVT (Operating System for Multiple Variable Tasks) des machines IBM 360.

En 1971, Jim Brown obtient sa thèse de l'université de Syracuse (New-York) « A Generalization of APL », Ph.D. Thesis. (voir : J.A. Brown, The origins of APL2, APL94 Conf. Proc., Quote-Quad 25.1, Sept. 1994).

En 1973, enfin, est faite l'annonce du système de fichiers pour l'APL d'IBM : le processeur auxiliaire TSIO et le nouvel APL-SV (APL Shared Variables). Alex Morrow en est le patron du développement. Cette version d'APL introduit des nouveautés majeures en plus des variables partagées. Toutes les anciennes fonctions I-beam sont remplacées par des fonctions et des variables système normalisées avec le symbole « Quad », incluant des nouveautés comme Quad-FX et Quad-CR (sans compter les quatre Quads : SVO, SVC, SVQ et SVR). Le domino est étendu à la résolution des systèmes surabondants par la méthode des moindres carrés. C'est aussi dans APL-SV qu'apparaît la primitive Exécute.

Mais déjà la communauté APL résonne, depuis 1972 environ, des bruits d'annonce d'une extension du langage aux tableaux généralisés (tableaux dont les éléments sont eux-mêmes des tableaux) et l'équipe APL d'IBM souffre du carcan des interdictions d'annonce et de débat que connaissait cette compagnie au temps de son quasi-monopole. C'est pourquoi la publication en 1973 de l'article « General Arrays, Operator and Functions » (Z. Ghandour, J. Mezei, IBM Journal of Research and Development, Vol.17, No. 4, July 1973) est lu avec attention par tous. C'est un fourre-tout très intéressant qui ne prend pas parti mais indique une foule de primitives et de constructions s'attaquant à la plupart des problèmes qu'implique l'extension d'APL aux tableaux généralisés.

En 1974, se produit un événement lourd de conséquence pour le développement d'APL : la fermeture du centre de Philadelphia et l'éclatement de l'équipe entre la côte Ouest (Palo Alto, Californie) pour les développements (Alex Morrow, Jim Brown) et la côte Est (Yorktown Heights, New-York) pour la recherche (Iverson, Falkoff).

En 1974, arrive l'annonce de la commercialisation d'APL/CMS réalisé par l'équipe de Bob Creasy (avec Tony Hassit et Len Lyon) permettant d'avoir une machine virtuelle APL sous le système VM.

En 1975, le premier micro-ordinateur, l'IBM 5100, était vendu avec clavier APL et le langage APL natif à côté du BASIC. Cette même année, IBM annonçait un nouveau système de time-sharing et un nouveau système APL : VSAPL commun à VSPC, VM/CMS et TSO. Tous ces systèmes sont mentionnés pour mémoire car le langage APL lui-même ne variait pas.

2 - Les variante

L'approche des tableaux généralisés (ou « nested arrays » de Trenchard More), l'importance donnée au sujet où tout est tableau, l'engouement pour la notation vectorielle des enclos (strand notation) impliquant que l'enclos d'un scalaire n'est pas un enclos mais un scalaire et bien d'autres choses (dont l'extension de l'usage des crochets) irritaient Iverson qui pourtant n'avait aucune prise sur les développements en cours à Palo Alto. Son effort portait sur l'idée d'opérateur telle qu'elle commençait à être reconnue depuis 1974 et sur la possibilité d'en faire un outil puissant pour rationaliser APL au moins dans trois domaines : le traitement des enclos, la suppression des notations anormales car multi-symboles (indexation et indication d'axes) et la définition directe de fonctions. Cette exigence primordiale souvent formulée en privé dès 1974-1975 a mis plus de dix ans à aboutir. On va donc voir, sur le strict plan du langage (concepts et notations) deux lignes se développer, la ligne majoritaire (en nombre d'implantations) étant bien sûr celle de l'APL2 d'IBM en raison du poids de cette société à cette période.

En 1976, Ken Iverson publie un rapport de recherche: « Two combinatoric operators » dans le but d'illustrer l'axe de développement de la notion d'opérateurs. Le produit interne monadique est proposé pour une fonction « Déterminant » généralisée et le produit externe monadique permet de faire une «table» des valeurs d'une fonction (table d'addition, table de multiplication, etc.). Ce texte est repris au congrès APL d'Ottawa de la même année.

En 1977, dans un séminaire interne IBM, Jim Brown fait une présentation d'APL2. La photocopie de ce rapport « confidentiel » n'en circule pas moins chez les grands clients. L'APL2 est prêt. Son développement a commencé dès 1973. Un an d'effort suffirait à pouvoir le commercialiser. Il faudra attendre cinq ans (1982) pour avoir un IUP (Installed User Program) et sept ans (1984) pour qu'il soit un produit programme.

En 1978, Ken Iverson publie « Operators and Functions », Research Report 7091, IBM Corp. March 1978. Cette fois tout y est. Ce texte contient la solution ou la piste de toutes les solutions et toutes les idées d'extensions d'APL auxquelles on pouvait rêver. Le texte comprend 9 pages et 17 paragraphes, dont voici les titres (volontairement laissés en anglais) :

1. Syntax and order of execution
2. Power and identity operators
3. Commutator
4. Scalar operators and identity functions
5. Domain and valence operator
6. Uniform functions
7. Nuclear axis operator
8. Composition and duality
9. Reduction and scan
10. Scalar representation
11. Derivative operator
12. Difference operator
13. The variant operator
15. Boolean operator
16. Set functions
17. Indexing

En regardant le contenu du document, on note que chaque paragraphe est équilibré et fait de 4% à 6% du texte, à l'exception du paragraphe « 8. Composition and duality » qui représente 21% du texte. Les tableaux généralisés (structures d'enclos) sont traités complètement, grâce aux opérateurs de composition et de dualité, au paragraphe « 10. Scalar representation » qui représente 6% du texte total, ni plus, ni moins, exemples compris.

En 1979, Ken Iverson présente « The derivative operator » et « The role of operators in APL » au congrès APL. Cette même année, le 29 octobre, à Detroit, il reçoit la plus haute récompense américaine en informatique : The ACM Turing Award. En cette fin d'année, Ken Iverson quitte IBM et entre au Canada, à Toronto, chez I.P. SHARP Associates.

En 1980, le système SHARP APL offre les opérateurs de composition et de dualité ainsi que les deux primitives enclose et disclose. Les implications en sont exposées dans la communication « Operators and enclosed arrays » B. Bernecky and K. Iverson, I.P. SHARP, 1980

En 1981, STSC donne accès à NARS (Nested Arrays Research System) système proche d'APL2, du moins dans son esprit si ce n'est dans le choix des symboles pour les primitives.

En 1982, IBM distribue enfin APL2 mais seulement comme IUP (Installed User Program) c'est à dire sans garantie d'un plein avenir et d'une pleine maintenance.

Cette même année apparaît Dyalog APL sous UNIX (de Dyadic Systems Ltd) qui est aussi très proche d'APL2.

En 1983, Ken Iverson publie « Rationalized APL » (I.P. SHARP Research Report, March 1983). C'est un livre de 48 pages (21cm x 27cm) qui expose ce que serait un APL rationnel et les développements (primitives et opérateurs) nécessaires pour y parvenir.

C'est un texte qui reprend, en le développant et en l'améliorant, le texte fondamental de 1978 « Operators and functions ». C'est à partir de ce texte qu'I.P. SHARP a continué ses développements du langage jusqu'à son rachat par REUTER en 1987. Notons que la version UNIX de SHARP APL (connu sous le nom de SAX et toujours vendue par SOLITON) va beaucoup plus loin que la version IBM. Elle a plus d'opérateurs et ils s'appliquent aux fonctions utilisateurs et aux fonctions dérivées comme aux fonctions primitives ce qu'I.P. SHARP n'a pas eu le temps de faire sur la version d'IBM.

En 1984, APL2 devient un produit programme d'IBM. Jim Brown publie « The principles of APL2 », Technical report 03.247, San Jose, IBM, March 1984. C'est un excellent texte qui présente clairement les règles qui gouvernent le langage (légèrement réaménagées depuis l'IUP) et les principes qui ont motivé les choix de conception. En 1988, Jim Brown publie avec S.Pakin et R.Polivka « APL2 at a Glance » chez Prentice Hall, ouvrage distribué avec l'APL2 et déjà traduit en allemand et en russe.

En ce qui concerne « Operators and functions » et « Rationalized APL », l'aventure culmine en 1987 quand l'ACM publie au congrès de 1987 « A DICTIONNARY OF THE APL LANGUAGE » de Ken Iverson, dernier Dictionnaire écrit par Ken en symboles APL. Quelques années après on retrouvera les mêmes idées écrites en langage **J**, mais ceci est (peut être) une autre histoire.

Nous en resterons là pour cette présentation des étapes du langage APL. Nous attendrons volontairement quelques années pour que l'histoire immédiate des systèmes APL pour micro-ordinateurs (MacIntosh ou IBM et compatibles sous DOS, puis sous WINDOWS ou sous OS2) puisse devenir l'objet d'une approche synthétique. Néanmoins, il est très intéressant de s'apercevoir que, déjà, la collection complète de notre revue « **Les Nouvelles d'APL** » contient, tant par ses articles que par ses publicités, un panorama presque complet de cette histoire à écrire. Je conseillerais donc vivement à tout lecteur intéressé de bien garder pour l'avenir les numéros successifs de notre revue qui l'aideront, un jour, à revoir comment les choses se sont passées.