



Les Rapports techniques ISO sont réexaminés tous les trois ans à partir de la date de leur publication, afin de parvenir à l'accord nécessaire pour la publication d'une Norme internationale.

Représentation machine des symboles de base d'ALGOL dans le jeu de caractères codés ISO à 7 éléments pour l'échange d'information entre matériels de traitement de l'information

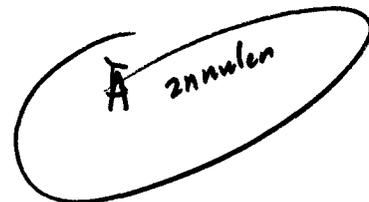
Hardware representation of ALGOL basic symbols in the ISO 7-bit coded character set for information processing interchange

Le Rapport technique 1672 a été établi par le comité technique ISO/TC 97, *Calculateurs et traitement de l'information*. Il résulte de révisions successives du projet de Recommandation ISO n° 1672, sur lequel les comités membres de l'ISO ont voté en 1969. La non-obtention d'un accord général pour la publication du document en tant que Norme internationale a conduit les membres du comité technique ISO/TC 97 à la décision de le publier en tant que Rapport technique. Les difficultés qui ont empêché un accord sur une Norme internationale sont exposées en détail dans l'annexe D.

En octobre 1976, ce document a été soumis au Conseil de l'ISO qui a approuvé sa publication en tant que Rapport technique.

Possibilité d'une future Norme internationale

Le groupe de travail IFIP 2.1, qui a la responsabilité d'ALGOL, est en train de considérer comme possible une révision du langage ALGOL 60. On suggère que toute tentative à venir pour produire une Norme internationale sur ce sujet devrait attendre des nouvelles venant d'IFIP pour savoir si oui ou non une telle version doit être faite.



0 INTRODUCTION

0.1 Le langage de programmation ALGOL 60 (ISO/R 1538) contient 116 «symboles de base». Il a toujours été admis que l'on ne pouvait espérer que l'ensemble des caractères disponibles sur ordinateur coïncide avec ces symboles et, par conséquent, que les représentations machine étaient nécessaires afin que le langage ALGOL puisse être utilisé en pratique.

0.2 Publiée en 1967, la Recommandation ISO/R 646, *Jeux de caractères codés à 6 et 7 éléments pour l'échange d'information entre matériels de traitement de l'information*, indiquait la possibilité d'une représentation machine approuvée sur le plan international et découlant des ensembles de caractères qu'elle décrivait.

0.3 Cependant, l'ISO/R 646-1967 fut remplacée par l'ISO 646-1973, qui normalisait uniquement le jeu de caractères codés à 7 éléments, reléguant l'ensemble à 6 éléments dans un appendice pour information seulement. Puisque la représentation d'ALGOL dans le code à 6 éléments avait été la cause de quelques-unes des difficultés du projet, la proposition fut refondue à partir du seul code à 7 éléments (tout en retenant les équivalents de représentation à 6 éléments comme des variantes quand elles ne provoquaient pas de difficultés).

CDU 681.3.04

Réf. n° : ISO/TR 1672-1977 (F)

Descripteurs : traitement de l'information, échange d'information, jeu de caractères codés, code ISO à sept éléments, ALGOL, symbole.

© Organisation internationale de normalisation, 1977 •

Imprimé en Suisse

Prix basé sur 9 pages

1 OBJET ET DOMAINE D'APPLICATION

Le présent Rapport technique suggère une correspondance entre les symboles ALGOL donnés dans l'ISO/R 1538-1972 (paragraphe 2.1, 2.2 et 2.3) et le jeu de caractères codés ISO à 7 éléments décrit dans l'ISO 646-1973. La version internationale de l'ISO 646 est admise comme référence, et les caractères @ [] { et } sont utilisés.

2 RÉFÉRENCES

ISO 646, *Jeu de caractères codés à 7 éléments pour l'échange d'information entre matériels de traitement de l'information.*

ISO/R 1538, *Langage de programmation ALGOL.*

3 REPRÉSENTATION

3.1 Dans le tableau qui suit, la colonne de gauche indique la représentation des symboles de base d'ALGOL. La colonne de droite indique le symbole équivalent pris dans le jeu de caractères codés à 7 éléments. Certains symboles de base sont représentés par un seul caractère; d'autres le sont par une suite de caractères.

3.2 Pour certains symboles de base d'ALGOL, des représentations distinctes sont indiquées. N'importe laquelle des représentations indiquées devrait être reconnue comme valable, même si un document n'est pas logique avec lui-même quand il indique parfois l'une des représentations et parfois une autre. Cela sauf en ce qui concerne les caractères [] et { } qui peuvent ne pas être autorisés dans certains contextes où ils seraient incompatibles avec l'usage national.

TABLEAU – Représentation des symboles de base

Référence	Représentation machine col/ligne	Référence	Représentation machine col/ligne
<lettre>	lettre correspondante majuscule ou minuscule	,	,
<chiffre>	chiffre correspondant	.	.
true	'TRUE' 'true'	10	@
false	'FALSE' 'false'	:	:
+	+	;	;
-	-	:=	:=
x	*	□	
/	/	step	'STEP' 'step'
÷	% '/'	until	'UNTIL' 'until'
↑	^	while	'WHILE' 'while'
<	**	comment	'COMMENT' 'comment'
≤	< 'LT' 'lt'	((
=	<= 'LE' 'le'))
≥	= 'EQ' 'eq'	[[
>	>= 'GE' 'ge'	(/	(/
≠	> 'GT' 'gt']]
≡	◇ 'NE' 'ne'	/)	/)
⊃	'EQV' 'eqv'	{	{
∨	'IMPL' 'impl'	'	'
∧	'OR' 'or'	,	,
¬	'AND' 'and'	}	}
go to	'NOT' 'not'	'	'
if	'GOTO' 'goto'	begin	'BEGIN' 'begin'
then	'IF' 'if'	end	'END' 'end'
else	'THEN' 'then'	own	'OWN' 'own'
for	'ELSE' 'else'	Boolean	'BOOLEAN' 'boolean' 'Boolean'
do	'FOR' 'for'	integer	'INTEGER' 'integer'
	'DO' 'do'	array	'ARRAY' 'array'
		real	'REAL' 'real'
		switch	'SWITCH' 'switch'
		procedure	'PROCEDURE' 'procedure'
		string	'STRING' 'string'
		label	'LABEL' 'label'
		value	'VALUE' 'value'

ANNEXE A

RÈGLES SUGGÉRÉES POUR LES DISPOSITIONS TYPOGRAPHIQUES

Conformément à l'ISO/R 1538 (paragraphe 2.3), « Les dispositions typographiques, telles que l'espace entre deux mots ou le changement de ligne, n'ont aucune signification dans le langage de référence ». Dans cette représentation machine, les règles équivalentes suivantes sont suggérées :

- a) la tabulation horizontale (0/9) est interprétée comme un espace (2/0);
- b) la tabulation verticale (0/11) ou page suivante (0/12) est interprétée comme un retour à la ligne (0/10);
- c) la suppression (7/15) est toujours ignorée aussi bien dans les symboles de base qu'entre ceux-ci, excepté dans le cas décrit en i) ci-après;
- d) l'espace (2/0) est toujours ignoré, sauf à l'intérieur des chaînes où il représente le symbole espace (␣) et excepté dans le cas décrit en i) ci-après;
- e) l'utilisation de l'espace arrière (0/8) est indéfinie;
- f) l'utilisation du retour chariot (0/13) sans l'emploi du retour à la ligne (0/10) est indéfinie;
- g) le retour à la ligne (0/10) est ignoré partout entre les symboles de base, mais n'est pas autorisé à l'intérieur d'un symbole de base;
- h) les autres caractères dans les colonnes 0 et 1 peuvent être significatifs tant que leurs fonctions de contrôle sont concernées, mais sont ignorés tant que leur effet dans le texte ALGOL est impliqué, excepté dans le cas décrit en i) ci-après;
- i) les caractères constituant le séparateur de chaîne (quand les versions à caractères multiples sont utilisées) ne doivent être séparés par aucun autre caractère.

ANNEXE B

PROPOSITION DE SOUS-CHAÎNES DE CARACTÈRES SYSTÈMES

B.1 Il y a une difficulté en ALGOL, définie dans l'ISO/R 1538 (paragraphe 2.6.1) :

⟨chaîne propre⟩ ::= ⟨toute séquence de symboles de base ne contenant ni 'ni'⟩ | ⟨vide⟩

Ainsi, apparemment, seuls les symboles de base d'ALGOL sont autorisés dans une chaîne. Cependant, dans beaucoup d'applications pratiques, il est important de pouvoir transmettre tous les caractères autorisés en incluant non seulement les caractères non-ALGOL visibles, mais également les caractères de commande non visibles.

B.2 La difficulté est accrue dans les procédures d'entrées/sorties décrites dans les parties II A et II B de l'ISO/R 1538 qui traitent des symboles de base d'ALGOL dans une chaîne et qui, en plus, demandent le compte des symboles de base. La chaîne ':= ' (voir ISO/R 1538, paragraphe 3.2.2) est alors ambiguë. Contient-elle un symbole de base ou deux ?

B.3 Pour surmonter cette difficulté, nous proposons qu'une chaîne puisse contenir, en plus des symboles de base, des sous-chaînes de caractères système définies comme suit :

⟨liste de caractères⟩ ::= ⟨col⟩/⟨ligne⟩ | ⟨liste de caractères⟩, ⟨col⟩/⟨ligne⟩

⟨col⟩ ::= ⟨entier sans signe⟩

⟨ligne⟩ ::= ⟨entier sans signe⟩

⟨sous-chaîne de caractères⟩ ::= "⟨liste de caractères⟩"

Si $col < 8$ et $ligne < 16$, alors le symbole indiqué est celui que l'on trouve dans la colonne et la ligne correspondantes de l'ISO 646. D'autres caractères peuvent être indiqués, par un accord préalable dans les limites d'un contexte donné, par des valeurs de colonnes ou de lignes dépassant les limites indiquées ci-dessus.

B.4 Dans cette proposition, les caractères qui constituent une chaîne sont à considérer comme étant si possible composés et travaillant de gauche à droite. Le retour à la ligne peut ne pas apparaître dans un caractère composé [voir annexe A, règle g)], mais peut apparaître entre les caractères composés ou à l'intérieur d'une sous-chaîne de caractères; il est alors ignoré.

B.5 La chaîne propre {:=} serait alors, de façon inambiguë, un symbole de base et non deux. Une chaîne contenant deux points suivie de égal sera représentée par {"3/10,3/13"} ou comme $\left. \begin{matrix} \vdots \\ \vdots \end{matrix} \right\}$ (où une nouvelle ligne suit immédiatement les deux points sans espace).

À l'intérieur d'une sous-chaîne de caractères, chaque ⟨col⟩/⟨ligne⟩ compte comme s'il était un symbole de base afin d'affecter des entrées aux symboles de base d'une chaîne. Le caractère " et les virgules à l'intérieur d'une liste de caractères ne seront pas considérés. Ainsi la chaîne

{B, "3/15, 0/12" 'TRUE'}

contient cinq symboles numérotés comme suit :

- 1 B
- 2 ,
- 3 ?
- 4 page suivante
- 5 true

Les caractères tirés d'une liste de caractères ne sont jamais à considérer comme composés et ne doivent jamais être considérés comme ayant une signification quelconque, ainsi {"2/7,2/15,2/7"} représente "/" que l'on ne doit pas considérer comme représentant '÷'. {"7/13"} signifie '}' et le caractère ne ferme pas la chaîne pour autant.

B.6 EXEMPLES

- a) {***} signifie '↑x' et non 'x↑' ou 'xxx'
- b) {◇=} signifie '≠' et non '◇'

ANNEXE C

REPRÉSENTATION DES MOTS SYMBOLES DE BASE

Quelques mises en œuvre d'ALGOL 60 n'utilisent pas d'indications spéciales pour les mots symboles de base, mais utilisent des représentations dans lesquelles ils ne peuvent être distingués des identificateurs. Cet usage peut être adopté par accord préalable dans les limites d'un contexte donné, à la condition que les règles suivantes soient respectées :

- a) aucun mot symbole de base ne peut être utilisé comme identificateur;
- b) les caractères tabulation horizontale (0/9), espace (2/0), tabulation verticale (0/11), page suivante (0/12), retour à la ligne (0/10) ne peuvent être utilisés à l'intérieur d'un identificateur ou à l'intérieur d'un symbole de base (en particulier, GOTO ne peut être écrit GO TO);
- c) si un mot symbole de base est précédé ou suivi par une lettre, un chiffre, ou un mot symbole de base, alors l'un des caractères indiqués en b) ci-dessus doit être utilisé comme séparateur.

ANNEXE D

DIFFICULTÉS AYANT EMPÊCHÉ UN ACCORD SUR UNE NORME INTERNATIONALE

Les difficultés soulevées par le présent projet ont été dans une large mesure d'ordre historique. Si ALGOL 60 était maintenant défini pour la première fois et si le jeu de caractères codés ISO à 7 éléments était disponible pour la représentation machine, il y aurait sans doute peu de difficultés à aboutir à un compromis qui satisferait les différents points de vue; mais, en fait, de nombreuses réalisations étaient disponibles chacune avec sa propre représentation machine, cela bien avant que les jeux de caractères ISO n'apparaissent.

Ainsi, chacun se retrancha derrière ses positions dont beaucoup impliquaient le sentiment, d'ailleurs sincère, que la tradition particulière par laquelle ils avaient connu ALGOL était la seule naturelle et que les autres traditions étaient un peu perverses. Ces difficultés de tradition seraient difficiles à résoudre, même si presque tous les matériels utilisaient le jeu de caractères ISO. La situation est encore compliquée, cependant, par l'existence de grandes quantités d'équipements utilisant d'autres jeux de caractères. Là encore les gens semblent être profondément retranchés derrière les jeux de caractères qu'ils connaissent et qu'ils utilisent.

Si une organisation internationale refuse d'établir un compromis avec ces autres ensembles de caractères, il va de soi que l'on n'arrivera à rien en ne voulant pas regarder la réalité. Et si l'on arrive vraiment à un compromis, alors qu'advient-il de la normalisation internationale ? Y aurait-il un quelconque avantage à produire un code de caractères comme Norme internationale si l'on devait considérer cette norme comme de peu d'importance dès qu'elle serait mise à l'épreuve ?

De plus, si l'on acceptait une quelconque variante, quel argument pourrait-on avancer pour refuser de considérer la demande de quelqu'un d'autre ? Et si l'on change le code établi pour prendre en compte un nouveau code qui se trouve être préféré par un quelconque constructeur, n'est-ce pas injuste pour un autre constructeur qui a pris la peine de mettre en œuvre le code ISO ?

Sur les 116 symboles de base d'ALGOL, il y en a 44 qui ne présentent aucune difficulté. Il y a les 26 lettres majuscules de A à Z, les chiffres de 0 à 9, les symboles () , . = + - / . Ils sont universellement disponibles et se calquent directement d'une représentation à l'autre.

Vingt-neuf autres symboles découlent directement du code ISO mais sont moins universels. Ce sont les 26 lettres minuscules de a à z et les symboles

! ; :=

le dernier découlant directement de l'utilisation de deux caractères pour un seul symbole. S'il est seulement prévu une case pour chaque lettre, cela est un désavantage mineur car, à vrai dire, les sous-ensembles ALGOL n'autorisent pas l'emploi simultané de plus de 26 lettres à la fois. Par contre, l'absence des 3 autres symboles rend l'emploi d'ALGOL impossible. Le présent Rapport technique ne suggère aucune variante, mais il pourrait être avantageux d'autoriser

... , :=

permettant ainsi la représentation normalisée d'ALGOL avec un petit sous-ensemble du code ISO.

Les 24 autres symboles sont fournis par des mots anglais «particularisés». La particularisation consiste en un mécanisme quelconque qui fait que ces mots peuvent clairement être distingués d'identificateurs constitués des mêmes lettres.

Dans le langage de référence, la particularisation consiste à souligner les mots manuscrits ou frappés, mais, cependant, dans le texte imprimé, les caractères gras sont habituellement utilisés. Pour l'entrée dans un ordinateur, le souligné a parfois été utilisé et peut donner l'apparence que l'on est très proche du langage de référence, mais cela n'est pas disponible sur la plupart des équipements.

Le souligné est disponible dans le code ISO mais seulement en utilisant l'espace arrière, ce qui, de façon notoire, est une caractéristique sujette à caution sur certains équipements. Le souligné est également désavantageux en rendant les symboles de base interminables. Le symbole procedure est constitué de 27 caractères (9 lettres, 9 espaces arrière, 9 soulignés) et nécessite la reconstruction de la ligne si tant est que l'on soit capable de dire, uniquement à partir de l'impression, quel programme est représenté.

Le souligné est donc à rejeter en faveur d'une autre particularisation qui est très utilisée, à savoir la mise entre apostrophes des mots considérés. Cela a pour avantage supplémentaire que le début et la fin de chaque symbole de base sont clairement indiqués, alors que les mots particularisés par le souligné peuvent être mis bout à bout en laissant le soin au compilateur de les reconnaître.

Il semble que l'on soit assez d'accord pour admettre que les apostrophes forment une méthode acceptable de particularisation si tant est que l'on doive utiliser la particularisation. En effet, certaines personnes sont persuadées que cette ponctuation supplémentaire n'est pas nécessaire et que les mots symboles de base pourraient être distingués des identificateurs simplement par le contexte. Cela ne peut être fait sans l'introduction de nouvelles règles, qui ont été rassemblées dans l'annexe C comme variante autorisée, mais certains considèrent cette représentation comme contraire à l'esprit d'ALGOL et souhaiteraient la voir abandonnée.

Sur les 17 symboles restants, 4 ont provoqué peu d'argumentation :

- x représentable par \star est à peu près universellement utilisé en informatique;
- \leq et \geq seraient peut-être mieux représentés par \leq et \geq , mais, ayant décidé de ne pas utiliser le souligné, la représentation \leq et \geq est généralement acceptée;
- \supset représentable par 'IMPL' n'a provoqué aucun commentaire. Le symbole est, dans tous les cas, si peu utilisé qu'il a peu de chances de provoquer de fortes réactions.

Le cas des 13 autres symboles est plus délicat :

- \div représentable par $\%$. ALGOL requiert obligatoirement deux symboles de division différents. Il n'est donc pas possible d'utiliser $/$ à la fois pour la division réelle et la division entière ainsi que cela est fait dans FORTRAN. $\%$ a l'avantage, à la fois par sa nature oblique, d'indiquer un signe de division et d'avoir, par son apparence, la forme approximative de \div mais certains arguments ont été soulevés en faveur de $//$.
 - \uparrow représentable par $\hat{\ } ou $\star\star$. La notation $\star\star$ est utilisée à la fois par FORTRAN et PL/I, et il y a un accord général pour qu'il soit autorisé comme une variante possible. $\hat{\ }$ est plus délicat. L'ISO/R 646 donnait ce caractère comme flèche vers le haut ou accent circonflexe. Aussi, il semble évident de le choisir pour représenter la flèche vers le haut d'ALGOL. Cependant, l'ISO 646 donne $\hat{\ }$ comme tête de flèche vers le haut ou accent circonflexe, ce qui rend le choix moins évident. La plus grande opposition à ce symbole provient du fait que, dans le code le plus employé (EBCDIC), la position de $\hat{\ }$ est utilisée pour le caractère \neg qui est également un symbole ALGOL. Pour les utilisateurs qui ont le symbole \uparrow de l'ISO/R 646, il semble absurde de ne pouvoir l'employer. Pour les utilisateurs d'EBCDIC, il semble encore plus absurde d'utiliser \neg pour signifier \uparrow .$
 - \neq représentable par \diamond . Il y a une certaine logique dans ce choix, puisque si \geq signifie «plus grand ou égal» et \leq signifie «inférieur ou égal», alors \diamond devrait signifier «inférieur ou supérieur», ce qui logiquement signifie «non égal». Cependant, cela n'a été que rarement mis en œuvre, et les utilisateurs d'EBCDIC préféreront \neg comme en PL/I, si \neg est disponible.
 - \equiv représentable par 'EQV'. Dans les premiers projets, 'EQUIV' avait été suggéré et cela provoqua peu de commentaires, mais il fut réalisé que cela pouvait provoquer une confusion avec la fonction *equiv* définie dans le paragraphe 1.2.3.2 de la partie II B de l'ISO/R 1538. Un vote, portant sur un certain nombre d'actions à adopter éventuellement, élimina toutes les suggestions sauf 'EV' et 'EQV' qui se rattachaient au premier choix. 'EQV' a été choisi du fait de son utilisation prépondérante actuelle.
 - \wedge et \vee représentables par 'AND' et 'OR'. Il est admis que ces représentations devraient être disponibles; la question est de savoir si oui ou non des variantes devraient être utilisées en plus des représentations ci-dessus. Un choix évident pour \wedge serait $\&$ qui est utilisé en PL/I, mais il semblerait illogique d'avoir un symbole pour *and* et non pour *or*. Le symbole PL/I est la barre verticale, mais elle présente les désavantages suivants :
 - i) elle est trop difficile, en manuscrit, à distinguer de 1, l et l;
 - ii) elle coïncide avec les variantes métasyntaxiques de la syntaxe formelle d'ALGOL.
- Au moment de la discussion, les membres du sous-comité 5 du Comité technique ISO/TC 97 ne se doutaient pas que la barre oblique inversée de l'ISO 646 avait été introduite afin de permettre la représentation de \wedge et \vee par $/$ et \backslash . Cela est digne de considération, mais semblerait quelque peu étrange dans des pays dont le système national avait adopté des symboles utilisés en tant que variantes pour la position 5/12.
- \neg représentable par 'NOT'. Rien ne pourra satisfaire les utilisateurs EBCDIC à part \neg , mais cela correspond à $\hat{\ }$ dans le code ISO. Une solution possible serait d'autoriser que $\hat{\ }$ signifie soit \uparrow soit \neg , le sens, dans chaque cas, étant déterminé par le contexte. Il n'y aurait pas d'ambiguïté sauf dans les chaînes.
 - \textcircled{E} représentable par \textcircled{E} . Dans un précédent rapport, la représentation E avait été donnée comme en FORTRAN. Cela conduisait à une ambiguïté dans ALGOL puisque 105 est un nombre valide mais que E5 est un identificateur autorisé. La particularisation suggérait que l'on prenne 'E', mais, puisque ce symbole est une partie d'une donnée et peut être utilisé aussi bien dans les données que dans les programmes, il faut considérer comme essentiel que 10 soit représenté par un seul caractère. À la réunion de Berlin du groupe de travail préparatoire, quand on pensait qu'il était essentiel que le jeu de caractères à 6 éléments soit inclus, il fut décidé que l'apostrophe était le seul symbole qui pourrait faire l'affaire. Il fut montré que ' pourrait très bien tenir ce rôle en plus de celui de la particularisation, sans ambiguïté sauf dans les chaînes. Cela était néanmoins déplaisant et les conséquences dans une chaîne assez tortueuses. À la réunion du sous-comité 5, à Washington, l'abandon de la représentation à 6 éléments conduisit à son remplacement par \textcircled{E} avec beaucoup de soulagement.
 - [] représentable par [] ou (/). L'utilisation de [] est manifeste là où elle est disponible. Toutefois, les normes nationales peuvent se surajouter à ces dispositions. () ne pourront être utilisées car ALGOL requiert la distinction entre les indices et d'autres parenthèses. (/) ont été assez largement mis en œuvre, mais certaines personnes auraient préféré (:).

' ' représentable par { } . Les normes nationales peuvent, là aussi, être surajoutées à ces dispositions et, dans ce cas, les variantes '(')' doivent être prises à la place. Dans ALGOL, on doit pouvoir distinguer entre les délimiteurs ouvrant et fermant une chaîne, et les suggestions concernant " " ne le permettent pas.

␣ représentable par un intervalle. Cela est probablement le plus délicat de tous les points litigieux.

Les arguments en faveur de l'utilisation de l'intervalle réel sont les suivants :

- a) c'est le choix raisonnable le plus naturel à faire. Qu'est-ce qui pourrait être plus évident ?
- b) si l'on insiste pour prendre un symbole visible, les programmeurs tendent à oublier de l'inclure. Personne, par contre, n'oubliera d'inclure un intervalle réel quand il est requis;
- c) ce que nous recherchons, c'est une représentation machine utile. Le purisme du langage de référence n'est pas à considérer;
- d) introduire la notion que, dans une représentation machine, un intervalle qui est significatif est seulement une petite irrégularité. Cela ne change pas l'indépendance d'écriture d'ALGOL à l'extérieur des chaînes, et ces chaînes ne sont, de toute manière, qu'une très faible partie d'ALGOL;
- e) dans l'utilisation de l'annexe C, les intervalles sont parfois reconnus comme séparateurs;
- f) les programmeurs qui se sont habitués à l'intervalle réel en changeraient à contre-cœur

Les arguments en faveur de l'utilisation d'un symbole visible sont les suivants :

- a) il est difficile de trouver une méthode satisfaisante pour inclure un intervalle réel et de le distinguer d'un « nul » dans une liste de caractères autorisés ou dans un document d'équivalences (comme, par exemple, dans le présent Rapport technique);
- b) l'un des avantages d'ALGOL est son indépendance vis-à-vis du support, et n'importe quel morceau de papier servira à enregistrer un programme. Il n'est pas nécessaire d'utiliser un code spécial. Cela devrait rester vrai pour la représentation machine aussi bien que dans le langage de référence;
- c) les programmeurs qui utilisent l'intervalle réel dans la représentation machine en arrivent à le considérer comme valide également dans le langage de référence et tendent à être troublés à l'idée que leur langage de référence authentifié par un programme est non valide;
- d) pour être sûr de la sortie que l'on obtiendra, il est nécessaire d'indiquer à une perforatrice combien d'espaces doivent être inclus et, dans une liste de programmes, de compter combien ont été inclus. Ces choses sont plus simples à faire avec un symbole espace visible que sans;
- e) puisque ALGOL n'est pas lié à une largeur de ligne particulière, le compte du nombre d'intervalles dans une chaîne peut dépendre de la détermination du nombre d'intervalles qui apparaissent à la fin de la ligne entre le dernier symbole non blanc et la fin de l'enregistrement. Cela est impossible à faire à partir d'une liste. Il devrait toujours être possible de dire si un programme est, ou n'est pas, correctement enregistré en examinant la liste, sans avoir à se référer à la source.
- f) le jeu de caractères ISO contient la tabulation horizontale (0/9) qui, en rapport avec l'annexe A, règle a), est à interpréter dans ALGOL comme un intervalle. Si les intervalles n'ont pas de signification même dans les chaînes, cela ne nuira en rien, mais, si les intervalles sont significatifs, alors la possibilité d'une tabulation horizontale entraîne que le compte des intervalles, à partir d'une impression, est non seulement difficile mais impossible;
- g) quand des programmes sont imprimés sous la forme d'un livre (qui se distingue de la sortie à partir d'un ordinateur), les caractères utilisés ont des tailles différentes et les imprimantes ont leurs propres conventions d'espacement. Il n'y a qu'à regarder du FORTRAN imprimé pour voir les choses affreuses qui résultent de ce fait. En désespoir de cause, les auteurs ont parfois été amenés à adopter un symbole d'espace visible (tel que "b", puisque FORTRAN dispose seulement des lettres majuscules) comme étant la seule possibilité de rendre clair ce qu'ils voulaient énoncer;
- h) les programmeurs qui se sont habitués à l'espace visible en changeraient à contre-cœur

Dans un précédent projet, le symbole _ (5/15) a été adopté pour indiquer ␣, mais un vote postal a maintenant conduit à son remplacement par un intervalle réel (2/0).

