



UNION INTERNATIONALE DES TÉLÉCOMMUNICATIONS

**UIT-T**

SECTEUR DE LA NORMALISATION  
DES TÉLÉCOMMUNICATIONS  
DE L'UIT

**Z.315**

**LANGAGE HOMME-MACHINE**

---

**SPÉCIFICATION DE LA SYNTAXE DU  
LANGAGE D'ENTRÉE (DE COMMANDE)**

**Recommandation UIT-T Z.315**

(Extrait du *Livre Bleu*)

---

## NOTES

1 La Recommandation Z.315 de l'UIT-T a été publiée dans le fascicule X.7 du Livre Bleu. Ce fichier est un extrait du Livre Bleu. La présentation peut en être légèrement différente, mais le contenu est identique à celui du Livre Bleu et les conditions en matière de droits d'auteur restent inchangées (voir plus loin).

2 Dans la présente Recommandation, le terme «Administration» désigne indifféremment une administration de télécommunication ou une exploitation reconnue.

© UIT 1988, 1993

Droits de reproduction réservés. Aucune partie de cette publication ne peut être reproduite ni utilisée sous quelque forme que ce soit et par aucun procédé, électronique ou mécanique, y compris la photocopie et les microfilms, sans l'accord écrit de l'UIT.

### SPÉCIFICATION DE LA SYNTAXE DU LANGAGE D'ENTRÉE (DE COMMANDE)

#### 1 Considérations générales

Le texte qui suit décrit les éléments du langage d'entrée. Des diagrammes syntaxiques du langage d'entrée figurent au § 4 dont les numéros internes correspondent à ceux du § 2. Si des éléments d'entrée sont utilisés dans la sortie, une référence à ces éléments est insérée dans la description du langage de sortie (Recommandation Z.316). Les aspects relatifs à la procédure sont pris en considération dans la Recommandation Z.317. Il convient de noter que certaines options peuvent être appliquées dans certains domaines de la syntaxe, avec un risque de confusion syntaxique. La mise en œuvre de telles options doit être décidée en fonction des caractéristiques particulières du système en question.

#### 2 Structure des commandes

##### 2.1 *Commande*

La commande commence par le code de commande, qui définit la fonction devant être accomplie par le système. Si d'autres informations sont nécessaires, le code de commande peut être suivi d'une zone paramètre qui en est séparée par un : (deux-points). La zone paramètre comprend un ou plusieurs blocs de paramètres (voir les § 2.3 et 2.9.1). Une commande est toujours complétée par un caractère d'exécution (voir la Recommandation Z.317).

##### 2.2 *Code de commande*

Le code de commande peut compter jusqu'à trois identificateurs séparés par le caractère – (tiret) (par exemple: zone fonctionnelle – type d'objet – opération). Si les codes de commande se présentent sous la forme d'abréviations mnémoniques isolées, il est recommandé qu'ils se composent du même nombre de caractères.

##### 2.3 *Bloc de paramètres*

Un bloc de paramètres contient l'information nécessaire à l'exécution de la fonction spécifiée dans le code de commande. L'information contenue dans un bloc de paramètres figure sous la forme d'un ou plusieurs paramètres propres à la commande. S'il s'agit de plusieurs paramètres, ils doivent être séparés les uns des autres par une , (virgule). Tous les paramètres d'un bloc doivent être de la même sorte, c'est-à-dire soit des paramètres définis par position, soit des paramètres définis par nom.

##### 2.4 *Paramètres*

Un paramètre identifie et contient un élément d'information, et peut être défini par son nom ou sa position. On peut omettre les paramètres non obligatoires, comme indiqué aux § 2.4.1 et 2.4.2.

###### 2.4.1 *Paramètre défini par position*

Un paramètre défini par position est constitué par une valeur de paramètre qui peut être précédée d'un nom de paramètre, dont il est séparé par un signe = (signe égal). Les paramètres doivent figurer dans un ordre prédéterminé à l'intérieur du bloc de paramètres. Dans les cas où l'on n'a pas à indiquer une valeur de paramètre, le paramètre est omis, en laissant le séparateur approprié ou l'indicateur nécessaire pour terminer une commande. Cela indique la position du paramètre dans le bloc de paramètres. L'omission d'un paramètre peut signifier qu'on fait référence à sa valeur de paramètre assignée à cette fin.

###### 2.4.2 *Paramètre défini par nom*

Un paramètre défini par nom se compose d'un nom de paramètre, suivi d'une valeur de paramètre dont il est séparé par un signe = (signe égal). Ces paramètres peuvent être placés selon un ordre arbitraire dans le bloc de paramètres. Dans les cas où l'on n'a pas à indiquer de valeur de paramètre, on omet également le nom de paramètre, le séparateur = (signe égal) et le séparateur, (virgule) qui suivent le paramètre. Cette omission peut signifier qu'on fait référence à la valeur par défaut. On peut aussi indiquer la valeur par défaut en spécifiant une valeur par paramètre

assignée à cette fin. Dans les cas où une valeur de paramètre sous-entend le nom de paramètre, on peut omettre ce dernier ainsi que le séparateur = (signe égal).

## 2.5 *Nom de paramètre*

Un nom de paramètre indique sans ambiguïté la nature et la composition de la valeur subséquente du paramètre. Il définit ainsi la valeur de paramètre et l'interprétation qui doit lui être donnée. Il s'agit en fait d'un identificateur. Il existe des noms de paramètre simples et des noms de paramètre composés. Le nom de paramètre simple indique une seule valeur de paramètre et un nom de paramètre composé indique une valeur de paramètre dans une liste ou un tableau de types de paramètre similaires.

### 2.5.1 *Nom de paramètre simple*

Un nom de paramètre simple se compose d'un seul identificateur.

### 2.5.2 *Nom de paramètre composé*

Un nom de paramètre composé se compose d'un ou plusieurs identificateurs séparés chacun par un - (tiret).

#### 2.5.2.1 *Indice*

Un indice se compose d'un ou plusieurs chiffres.

## 2.6 *Valeur de paramètre*

Une valeur de paramètre contient l'information nécessaire pour spécifier le ou les objets, ou la ou les valeurs appropriées; elle se compose d'une ou plusieurs unités d'information. Dans le cas où il n'y a pas de groupement d'informations (voir le § 2.9), une valeur de paramètre se réduit à un argument de paramètre. Voir le § 2.10 pour ce qui concerne l'interrogation d'une base de données.

## 2.7 *Argument de paramètre*

Un argument de paramètre contient l'information nécessaire pour spécifier l'objet ou la valeur approprié(e). Un argument de paramètre est la forme sous laquelle se présente une valeur de paramètre quand il n'y a pas de groupement de l'information (voir le § 2.9). Il existe des arguments de paramètre simples et des arguments de paramètre composés.

### 2.7.1 *Argument de paramètre simple*

Un argument de paramètre simple se compose d'une seule unité d'information.

### 2.7.2 *Argument de paramètre composé*

Un argument de paramètre composé se compose de deux unités d'information au moins, séparées par un - (tiret).

## 2.8 *Unité d'information*

L'unité d'information représente la plus petite quantité d'information du langage, du point de vue de la syntaxe. Une unité d'information peut être un nombre, un identificateur, un nom symbolique, une chaîne de texte ou une expression arithmétique. Un nombre a toujours une base par défaut (par exemple hexadécimale) que l'on peut "forcer", si nécessaire, en introduisant la base désirée comme spécifié dans la Recommandation Z.314. Toutefois, la base par défaut pour un nombre introduit par clavier ne peut être "forcée" par une autre base.

## 2.9 *Groupement d'informations*

On a recours à un groupement d'informations lorsqu'on désire accélérer et faciliter les opérations d'entrée. On réalise un tel groupement en rassemblant dans la même commande plusieurs ensembles d'informations du même type.

### 2.9.1 *Groupement des blocs de paramètres*

Si plusieurs blocs de paramètres doivent être inclus dans une même commande, ils seront séparés par un : (deux-points).

## 2.9.2 Groupement des arguments de paramètres

Le groupement des arguments de paramètres permet de saisir plusieurs arguments de paramètres dans un même paramètre d'une même commande.

### 2.9.2.1 Groupement des arguments de paramètres simples

Il est possible d'indiquer plusieurs arguments de paramètre simples dans la même valeur de paramètre, en les séparant par un & (perluète). *Exemple 1:* 5&9 désignent les arguments de paramètre de paramètre simples 5 et 9.

Dans le cas d'une séquence d'arguments de paramètre simples consécutifs (valeur d'accroissement implicite = 1), il est possible d'indiquer les arguments en inscrivant l'argument de paramètre simple le plus bas et l'argument de paramètre simple le plus élevé et en les séparant par && (perluète, perduète)<sup>1)</sup>. *Exemple 2:* 5&&9 désigne les arguments de paramètre simples 5, 6, 7, 8 et 9.

Une valeur d'accroissement explicite peut être spécifiée après l'argument de paramètre supérieur séparé par + + (signe plus, signe plus). *Exemple 3:* 5&&9+ + 2 désigne les arguments de paramètre simples 5, 7 et 9.

Si nécessaire, on peut aussi avoir recours à d'autres combinaisons des possibilités indiquées ci-dessus. *Exemple 4:* 5&&7&9 désigne les arguments de paramètre simples 5, 6, 7 et 9. *Exemple 5:* 5&&9+ +2&10 désigne les arguments de paramètre simples 5, 7, 9 et 10.

### 2.9.2.2 Groupement des arguments de paramètre composés

Il est possible d'indiquer par & (perluète) plusieurs arguments de paramètre composés à l'intérieur de la même valeur de paramètre. *Exemple 1:* 5-1&6-3 désigne les deux arguments de paramètre composés 5-1 et 6-3.

Si, à l'intérieur d'un groupe, les arguments de paramètre composés diffèrent seulement par la dernière unité d'information, le premier argument composé est spécifié complètement, tandis que tous les arguments composés suivants sont représentés uniquement par leur dernière unité d'information séparés par &- (perluète, tiret). *Exemple 2:* 7-1&-3 désigne les deux arguments de paramètre composés 7-1 et 7-3.

Si, à l'intérieur d'un groupe, les arguments de paramètre composés diffèrent seulement par la dernière unité d'information et si le groupe constitue une séquence ininterrompue (valeur d'accroissement implicite = 1), il est possible d'indiquer les arguments en inscrivant l'unité d'information la plus haute et l'unité d'information la plus basse, séparées par &&- (perluète, perduète, tiret)<sup>1)</sup>. *Exemple 3:* 7-1&&-3 désigne les trois arguments de paramètre composés 7-1, 7-2, 7-3. *Exemple 4:* 7-1&-3&&-5 désigne les quatre arguments de paramètre composés 7-1, 7-3, 7-4 et 7-5.

Une valeur d'accroissement explicite peut être spécifiée après l'unité d'information la plus haute séparée par + + (signe plus, signe plus).

Si cela est nécessaire, on peut aussi utiliser une combinaison quelconque des possibilités indiquées ci-dessus. *Exemple 5:* 5-1&&-3&8-2&-5&-6 désigne les six arguments de paramètre composés 5-1, 5-2, 5-3, 8-2, 8-5 et 8-6. *Exemple 6:* 5-1&&-7 + + 2&8-1&-3 désigne les six arguments de paramètre composés 5-1, 5-3, 5-5, 5-7, 8-1 et 8-3.

## 2.10 Interrogation d'une base de données

L'interrogation d'une base de données s'exprime sous la forme d'information de projection et de sélection. L'information de projection peut être représentée par un paramètre. Son nom identifie la fonction de projection. Son groupe d'argument(s) de paramètre identifie le(s) champ(s) approprié(s) des enregistrements de données à afficher. L'information de sélection peut être représentée par un paramètre dont le nom identifie la fonction de sélection et la valeur identifie un argument de sélection (ou un groupe d'arguments de sélection). Un argument de sélection comprend une ou plusieurs conditions qui doivent toutes être remplies. Une condition est spécifiée par un identificateur et un argument de paramètre (ou un groupe d'arguments de paramètre), séparés par un opérateur de relation. L'identificateur spécifie le nom du champ de l'enregistrement à sélectionner. L'omission de l'information de sélection signifie que l'interrogation n'est pas conditionnelle.

Les mots "projection" et "sélection" n'ont été choisis qu'à titre d'exemple. D'autres mots comme "select" et "where" peuvent être utilisés.

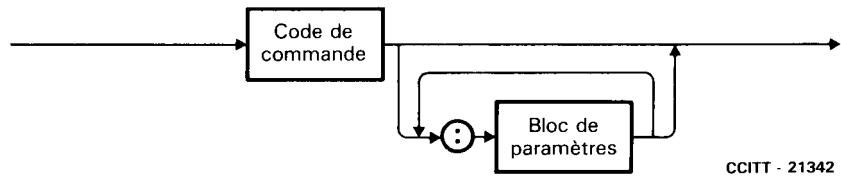
---

<sup>1)</sup> L'interprétation des séparateurs && (perluète, perduète) et &&- (perluète, perduète, tiret) n'est pas exclusive. Il existe d'autres interprétations. Une des variantes envisagées supposerait que la syntaxe ne contienne aucun moyen de désigner un accroissement spécifique. Autrement dit, la relation entre la valeur supérieure et la valeur inférieure dans la séquence est une relation sémantique qui dépend de la fonction pour laquelle la séquence est spécifiée.

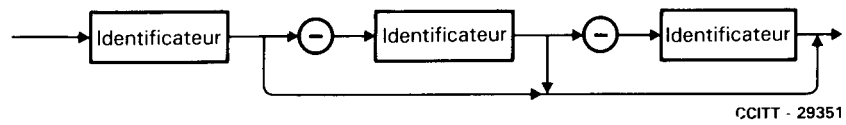


## 4 Définition de la structure du langage d'entrée (commande) sous forme de diagrammes syntaxiques

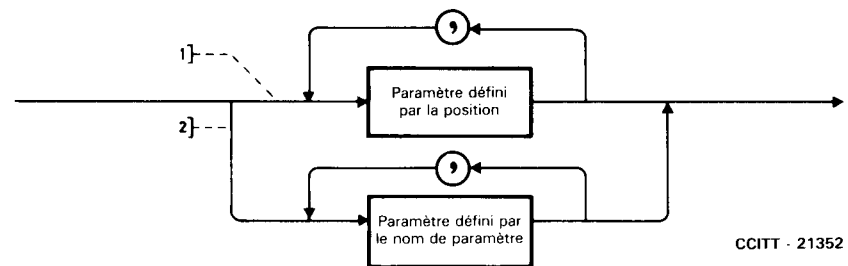
### 4.1 Commande



### 4.2 Code de commande



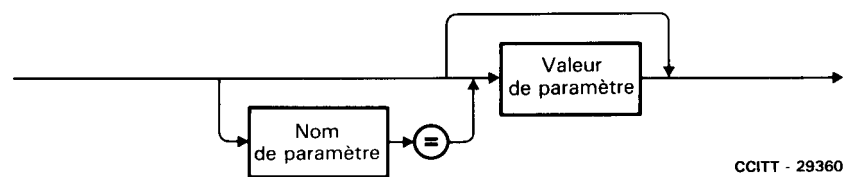
### 4.3 Bloc de paramètres



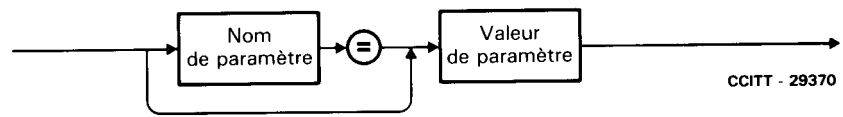
- 1) La branche supérieure n'est valable que pour des blocs de paramètres définis par position.
- 2) La branche inférieure n'est valable que pour des blocs de paramètres définis par nom.

### 4.4 Paramètres

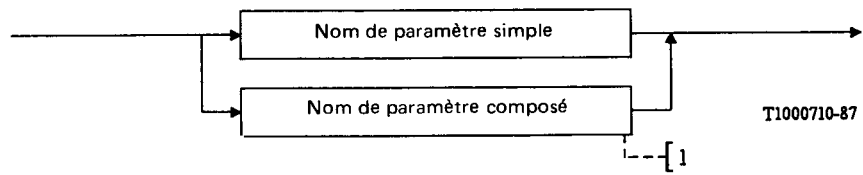
#### 4.4.1 Paramètre défini par position



#### 4.4.2 Paramètre défini par nom

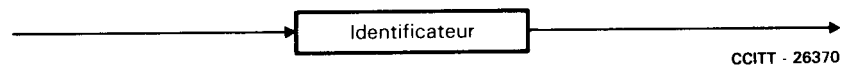


#### 4.5 Nom de paramètre

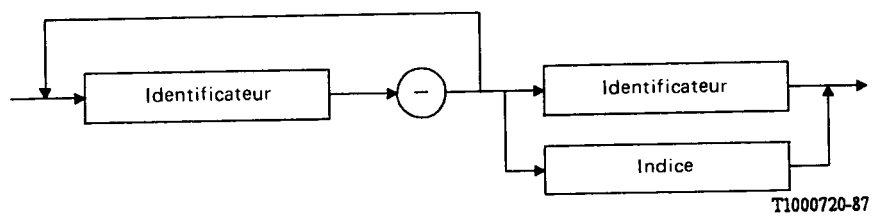


1) Facultatif.

#### 4.5.1 Nom de paramètre simple

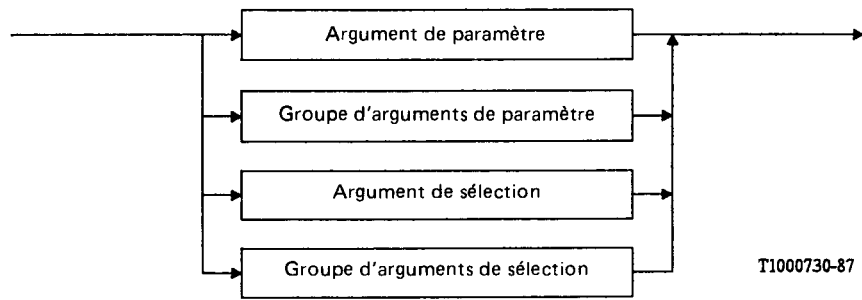


#### 4.5.2 Nom de paramètre composé

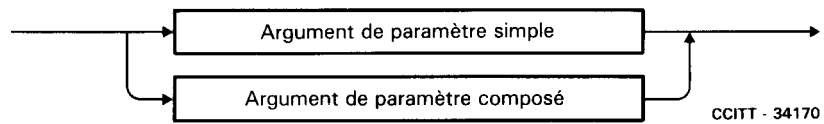




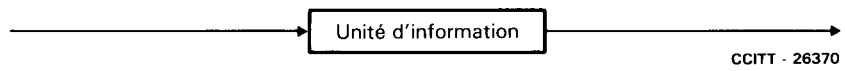
4.6 Valeur de paramètre



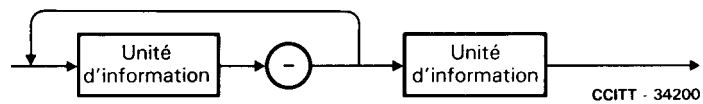
4.7 Argument de paramètre



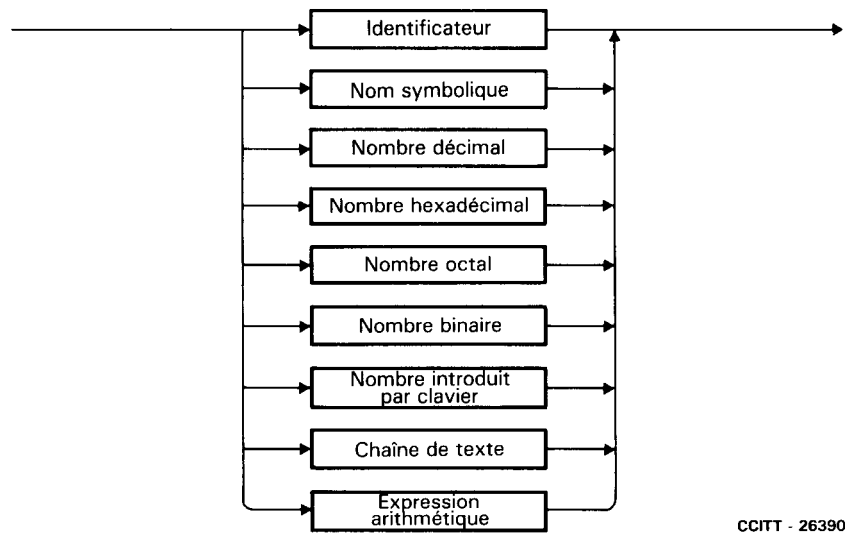
4.7.1 Argument de paramètre simple



4.7.2 Argument de paramètre composé



4.8 Unité d'information

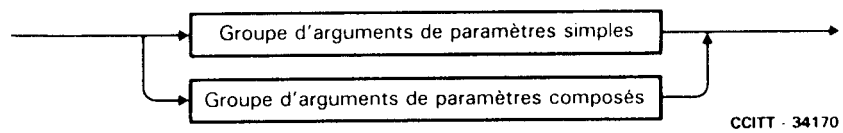


4.9 Groupement de l'information

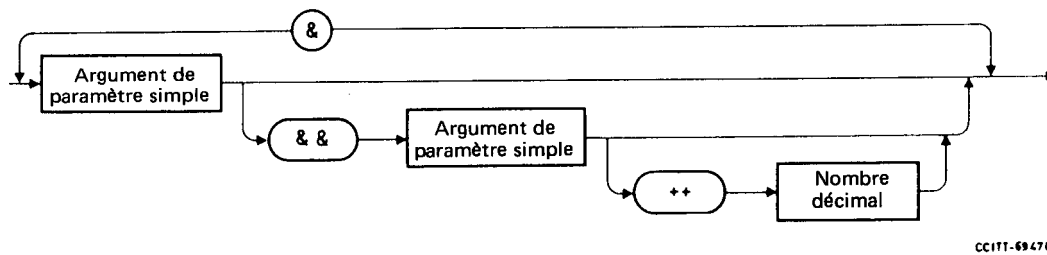
4.9.1 Groupe de blocs de paramètres

Voir le diagramme syntaxique du § 4.1.

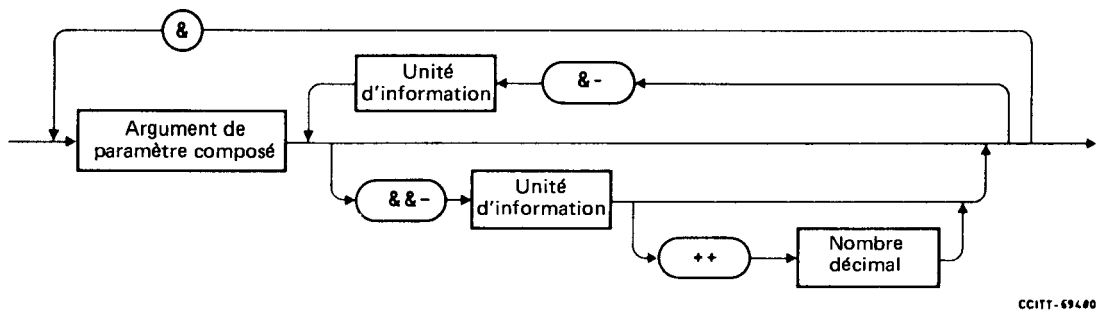
4.9.2 Groupe d'arguments de paramètre



4.9.2.1 Groupe d'arguments de paramètre simples



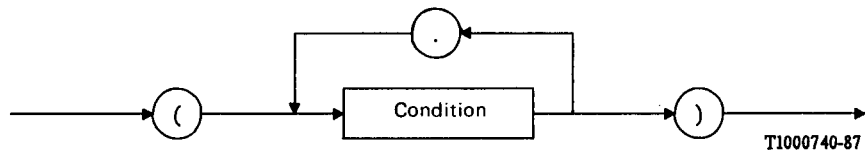
4.9.2.2 Groupe d'arguments de paramètre composés



CCITT-69400

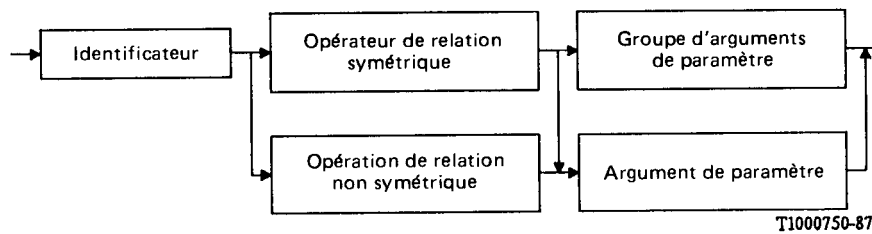
4.10 Interrogation d'une base de données

4.10.1 Argument de sélection



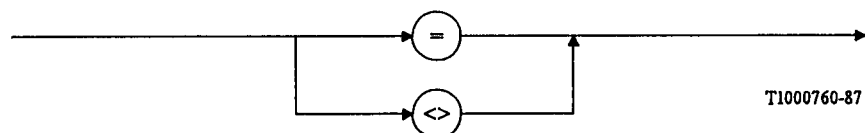
T1000740-87

4.10.1.1 Condition



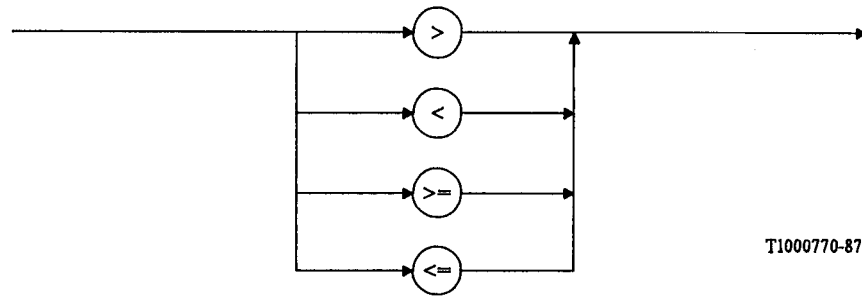
T1000750-87

4.10.1.2 Opérateur de relation symétrique



T1000760-87

4.10.1.3 Opérateur de relation non symétrique



4.10.2 Groupe d'arguments de sélection

