

Cahiers **GUT**enberg

☞ UNE ÈRE NOUVELLE : T_EX 3.0

☞ Michael J. FERGUSON

Cahiers GUTenberg, n° 5 (1990), p. 3-7.

<http://cahiers.gutenberg.eu.org/fitem?id=CG_1990__5_3_0>

© Association GUTenberg, 1990, tous droits réservés.

L'accès aux articles des *Cahiers GUTenberg*

(<http://cahiers.gutenberg.eu.org/>),

implique l'accord avec les conditions générales

d'utilisation (<http://cahiers.gutenberg.eu.org/legal.html>).

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.

Une ère nouvelle : T_EX 3.0

Michael J. Ferguson

*INRS-Télécommunications, Montréal,
3, place du Commerce, CDN-H3E 1H6, Verdun PQ, Canada*

Résumé Cet article présente quelques observations préliminaires sur les nouvelles fonctionnalités de T_EX 3.0 et leurs relations avec les fonctionnalités actuelles de T_EX Multilingue. Il propose aussi une modification simple à T_EX 3.0 qui donne approximativement les mêmes possibilités que l'actuel T_EX.

Abstract *This paper will present some preliminary observations on the new capabilities of T_EX 3.0 and their relationship to the current capabilities of Multilingual T_EX. This paper will also discuss a simple modification of the T_EX 3.0 that unlocks the features that give it approximately the capability of the current T_EX.*

1. Introduction

La version 3.0 de T_EX (appelée ici désormais T_EX 3.0¹) représente une grande amélioration par rapport à la version actuelle de T_EX. Cet article présente quelques observations préliminaires sur ses nouvelles fonctions et sur leurs relations avec les fonctions actuelles de T_EX Multilingue. Il propose aussi une modification simple à T_EX 3.0 qui donne approximativement les mêmes possibilités que l'actuel T_EX. Parmi les nouvelles possibilités, citons un raffinement supplémentaire dans les ligatures qui deviennent particulièrement puissantes et qui demanderont beaucoup d'imagination créatrice pour les utiliser lors de la définition, avec METAFONT, de nouvelles polices de caractères.

Une autre nouveauté de T_EX 3.0 qui devrait apporter une aide considérable à la production de documents professionnels

est une méthode puissante pour éviter, presque automatiquement, les boîtes trop pleines tout en produisant une très bonne coupure des lignes. Ce nouvel élément, `\emergencystretch`, évite les coupures horribles qui se produisent parfois lors de la mise en page sur deux colonnes avec L^AT_EX ; il permet à un éditeur d'obtenir une coupure adéquate sans avoir à toucher au texte original.

Combiné avec un ensemble de polices, de pilotes d'impression et de pilotes d'affichage sur écran cathodique, T_EX 3.0 pourrait servir à la fois de système de préparation de document et de base d'échange de documents. L'avenir est prometteur mais pour y arriver, nous ne devons pas emprunter un chemin trop rocailleux.

2. Modifications similaires à T_EX Multilingue

T_EX 3.0 a potentiellement toutes les fonctions de l'actuel T_EX Multilingue. Il est, d'abord et avant tout, multilingue. De plus, son nouveau codage interne sur 8-bits permettra d'inclure presque tous les caractères accentués et de mettre un trait d'union dans les mots contenant ces caractères.

Comme T_EX, T_EX 3.0 utilise une valeur entière, `\language`, pour distinguer les différentes langues. La valeur de `\language` détermine les motifs (`patterns`) à utiliser quand un mot prend un trait

¹Voir Donald KNUTH, « T_EX 3.0 ou le T_EX nouveau va arriver », *Cahiers Gutenberg*, n° 4, décembre 1989, 39-45.

d'union. Cependant, T_EX 3.0 permet de changer, à l'intérieur d'un paragraphe, les règles de césure. Dans ce cas, il est similaire à la version « mot à mot » de T_EX, mais ne nécessite pas l'augmentation de mémoire requise par T_EX. Ceci signifie que cette possibilité de « mot à mot » sera disponible sur les petits systèmes.

Le format d'entrée de `\patterns` est moins rigide que celui de T_EX puisque l'on peut citer la langue et appeler `\patterns` dans n'importe quel ordre. Cependant, comme la primitive `\patterns` ne peut être réouverte après avoir été fermée, on ne peut ajouter une nouvelle langue (à `lplain` par exemple) en faisant simplement tourner INIT_EX avec l'ancien format et un ensemble complet de nouveaux `patterns`. Ceci n'est pas très important. Comme dans T_EX, de nouvelles exceptions de césures peuvent être introduites à tout moment et affecter la langue alors en cours.

Les paramètres primitifs `\lefthyphenmin` et `\righthyphenmin` sont des paramètres entiers qui contrôlent la taille acceptable des fragments, respectivement au début et à la fin d'un mot. Contrairement à leurs équivalents `\starthyph` et `\stophyph` de T_EX, ces paramètres ne sont pas initialisés, respectivement à 2 et 3. Ils seront initialisés dans le nouveau format `plain` ou dans un autre format.

3. Modification spécifiques à T_EX 3.0

Les caractères sur 8-bits et la primitive `\emergencystretch` sont des fonctionnalités propres à T_EX 3.0. Les nouvelles ligatures intelligentes et les paramètres primitifs `\errorcontextlines`, `\badness`, `\inputlineno`, et `\holdinginserts` n'ont aucun équivalent dans

T_EX.

T_EX 3.0 utilise des caractères internes de 8-bits tandis que T_EX utilise des pseudo-caractères de 8-bits. Ceci signifie que tous les caractères d'entrée sont maintenant des « citoyens de première classe » et peuvent donc être utilisés, par exemple, comme noms de macros. Ainsi le nom de macro `\noël` est valide dans T_EX 3.0 mais non en T_EX. N'importe quel caractère peut devenir actif simplement en le spécifiant.

T_EX 3.0 fera maintenant une troisième passe sur paragraphe pour essayer de couper les lignes sans produire de boîtes `overfull`. Ceci ajoute des liens supplémentaires, appelés `\emergencystretch` entre les mots de chaque ligne et lui donne plus de liberté pour couper la ligne et éviter certains espacements particulièrement laids qui se manifestaient en utilisant, par exemple, le mode à deux colonnes dans un article comme celui-ci. Cet article, par exemple, a été produit en choisissant `\emergencystretch=10pt`. Il n'a pas été nécessaire (cependant ceci aurait été possible) d'en augmenter la valeur. Ce mécanisme semble permettre une meilleure coupure des lignes que l'option `multipass` introduite récemment dans T_EX.

Les primitives `\errorcontextlines`, `\badness`, `\inputlineno`, seront utiles pour produire une interface légèrement plus amicale pour l'utilisateur de T_EX. La nouvelle primitive `\holdinginserts` devrait être utile pour définir les routines `\output` qui réalisent des mises en page complexes. Le placement d'une insertion, (par exemple d'une `\footnote` ou d'une figure) dans l'`\output`, se fait en dernier. La primitive permet de mieux contrôler ce positionnement.

3.1. Ligatures élégantes et ligatures de bordure

La modification la plus originale de T_EX 3.0 est le nouveau et puissant mécanisme de ligature. Les possibilités potentielles de cette innovation sont encore largement inconnues. Ce mécanisme semble être essentiel à certaines langues telles que l'arabe mais devrait aussi permettre des compositions originales dans toutes les langues. On aurait aimé présenter un exemple dans cet article mais les nouvelles polices ne sont pas encore disponibles.

Un usage potentiel du nouveau mécanisme de ligature pourrait être l'insertion d'une forme de coupure `\discretionary` spécifique à chaque langue. Le symbole de trait d'union dans les polices, habituellement le « - », en conjonction avec les autres caractères pourrait déterminer cette insertion. Ceci ne pourrait servir que si les conventions de la langue sont suffisamment régulières pour le permettre. Il faudrait plus de travail pour savoir s'il en est ainsi. Comme il n'y a actuellement aucun mécanisme pour rendre le programme des ligatures adapté à une langue, il faudra disposer de fichiers de métriques de fontes propres à chaque langue qui utilise cette possibilité. Il n'y aura alors plus besoin d'avoir un fichier de « pixel » par langue.

4. Fonctions spécifiques à T_EX Multilingue

T_EX permet la coupure de mots avec des caractères explicitement accentués tandis que T_EX 3.0 ne le permet pas. Dans le futur, quand les nouvelles polices de caractères codés sur 8 bits auront été produites pour des jeux « complets » de

caractères (accentués compris), on aura moins besoin de cette possibilité, voire plus du tout. Il semble cependant que dans un court terme, et peut-être dans un long terme pour l'usage de fontes moins classiques, cette possibilité sera encore nécessaire.

T_EX permet aussi la coupure de mots ayant des césures explicitement décrites par la commande `\discretionary`. Ceci permet la coupure de mots tels que INRS-Télécommunications, et l'insertion de coupures `\discretionary` complexes comme requises pour l'allemand.

4.1. Modification à T_EX 3.0

La modification `charsub` est une méthode simple et élégante pour étendre les fonctionnalités de T_EX 3.0 afin de permettre la coupure de mots avec des caractères explicitement accentués. Mais elle ne permet quand même pas la coupure de mots comportant la commande `\discretionary`. La modification exploite le codage interne sur 8-bits de T_EX 3.0 en représentant le caractère explicitement accentué par un code sur 8-bits. Juste avant que ce caractère ne soit envoyé au fichier `.dvi`, il est reconstruit en utilisant le même algorithme que celui utilisé par la primitive `\accent`. Une nouvelle primitive, `\charsublist`, définit la séquence de substitution. Elle est semblable à `\chardef` mais requiert trois nombres (ou équivalents) comme arguments : le code interne sur 8-bits, le code pour l'accent et le code pour le caractère de base. La définition de `é` est donc :

```
\charsublist '\é '023 '\e
```

Ceci suppose que le code 8-bits pour la saisie de `é` soit approprié.

L'implémentation actuelle de T_EX suppose aussi qu'il n'y a seulement qu'un ac-

cent et un caractère de base appartenant à la même police. Ceci fonctionne parce que les informations sur la police sont présentes pendant que le caractère est traité par \TeX . Un caractère est reconstruit seulement s'il n'y a pas de description pour ce même code de 8-bits dans la police et s'il y a une liste de substitutions de caractère pour ce code. Durant le processus, \TeX tente de déterminer si le caractère existe et si les dimensions requises sont satisfaites en substituant le caractère de base du paramètre `\charsublist` au code de 8-bits. Si `\tracinglostchars > 100`, alors chaque substitution de caractère est affichée à l'écran et écrite dans le fichier `.log`. En pratique, cette information ne présente pas un grand intérêt.

En plus des macros qui définissent `\charsublist` pour les lettres accentuées, des macros ont aussi été définies pour assurer la compatibilité avec les macros d'accentuation en \TeX actuel : elles utilisent un code équivalent de 8-bits s'il existe une liste de substitution, mais, sinon, utilisent explicitement `\accent`.

Enfin, un groupe de définitions `\accenthyphcode`, en gros identiques aux macros de compatibilité mais sans tests d'existence de liste de substitution de caractères, permettent d'utiliser directement les césures de \TeX en \TeX 3.0.

5. Lacunes de \TeX 3.0 et de \TeX Multilingue

La faiblesse la plus évidente de \TeX 3.0 et de \TeX est leur incapacité à traiter automatiquement les coupures de type `\discretionary` pour une langue comme l'allemand. Il est possible que l'approche de la ligature intelligente expliquée plus haut marche, mais ceci n'est pas encore prouvé.

6. Discipline en échange de documents

\TeX 3.0, le nouveau METAFONT et les pilotes d'imprimante adaptés forment un ensemble suffisamment souple pour pouvoir servir de mécanisme d'échange de documents. Si l'on veut qu'il en soit ainsi, il nous faut un minimum de discipline et notamment prévoir les points suivants :

- Le codage des caractères dans une police doit correspondre au codage des caractères en saisie de la façon la plus « normale » possible. Nous devrions donc utiliser, autant que possible, les normes internationales. Cinq alphabets latins sont actuellement définis par l'ISO. Tous les cinq sont identiques au code ASCII actuel pour les 128 premiers caractères mais ils diffèrent pour les 128 caractères suivants. Les 32 premiers caractères et les 32 compris entre 128 et 160 sont des codes de commandes et sont non-imprimables. Mais, dans les polices actuelles, on a affecté arbitrairement certaines lettres aux caractères non-imprimables. La norme ISO Latin 1 couvre pratiquement toutes les langues de l'Europe de l'Ouest, y compris l'islandais. De plus, il comprend des signes, comme « œ » qui apparaissent dans la police romaine actuelle de \TeX . Les quatre autres normes couvrent une variété de langues incluant, entre autres, le serbo-croate, l'albanais et le turc. Il faut que nous rationalisons le contenu de nos polices pour qu'elles puissent servir au plus grand nombre possible de langues, tout en tenant compte du fait que \TeX a besoin de ses propres polices mathématiques.

- Concevoir un ensemble de macros génériques qui pourront traiter les conventions de chaque langue. Elles devraient être, autant que possible, indépendantes de toute mise en page particulière.
- Concevoir un ensemble cohérent de conventions pour nommer, en fonction de la langue, les différentes macros, paquets de macros et styles qui sont utilisés. Une suggestion est d'utiliser le code à deux lettres utilisé actuellement comme adresse de domaine pour le courrier électronique (uucp ou internet). Les fichiers de `patterns` en allemand, espagnol et italien seraient ainsi appelés respectivement `dghyphen.tex`, `eshyphen.tex`, et `ithyphen.tex`. Il faudrait quelques variantes à cette convention pour les cas où il n'y a pas de code national unique.
- Utiliser le mode d'encodage actuel de T_EX, par exemple `\'e` pour `é`. Ceci est facilement convertible et permet l'échange de documents avant qu'une entente universelle sur le codage n'existe.
- Utiliser un format standard, comme *Encapsulated PostScript*, pour l'échange de graphiques. L'échange de documents complets nécessite cette possibilité.

7. Conclusion

Nous sommes à l'aube d'une ère nouvelle dans la production et l'échange de documents. Pour ceci, nous devons nous entendre et développer l'infrastructure adéquate comprenant macros, polices et systèmes d'affichage et d'impression. Il faut