

ODA に基づいた文書割付け処理の実現方式(3)

— 繰返し構造の処理方式 —

6P-3

山口 琢*, 松平 秀樹**, 上原 徹三***, 鍵政 秀子***

* (株) 日立製作所 ソフトウェア工場, ** 日立ソフトウェアエンジニアリング(株),

*** (株) 日立製作所 中央研究所

1. はじめに

ODA¹⁾などの共通割付け構造における繰返し構造の処理(以下「REP処理」)に単純Backtrack法を用いるときの主な問題は次の2つである⁴⁾。

- ★ 無限ループに陥る可能性がある、
- ★ 繰返し回数の修正に伴う再割付け処理のオーバーヘッドが大きい、

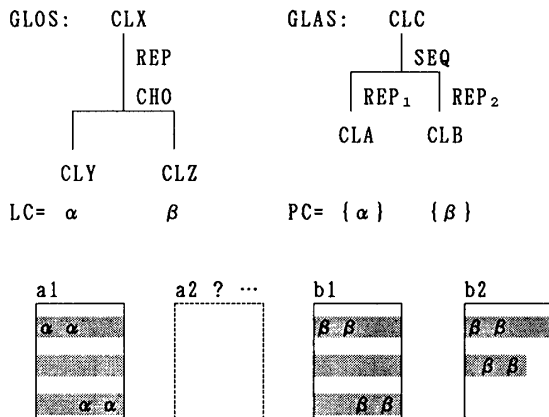
今回、この問題点の解決法を考案し、それをプロトタイプで実装したので報告する。

2. REP処理の問題点

上記問題点について、例をあげて説明する。

2.1 無限ループ

図1の例1では、論理構造上 category α, β の2種類の内容が混在して現れる。一方割付け構造においては α 、あるいは β の内容だけを許容する2種類のページがそれぞれ繰返し構造に指定されている。



CLX, CLY, CLZ, CLA, CLB, CLC : Object Class
an : CLA の instance, bn : CLB の instance
LC : Layout Category, PC : Permitted Category
GLOS : 共通論理構造, GLAS : 共通割付け構造

図1 例1: REP処理不能の例

まず α の内容が割付けられその後に β の内容が割付けられたとする。するとまず前のREP₁の代替情報が stack され次に後から現れたREP₂の代替情報が stack される。処理が進み、 α を許容するページが不足したとき、この窮状を解決するにはREP₁の繰返し回数を増やすしかない。ところが、REP₂にはこの窮状に無関係な代替手段が無限にあり、いつまでたってもREP₁の代替情報にbacktrack できない。割付け処理は無限ループに陥る。

この例からどの代替情報に backtrack するのか選択する機能が必要であることが分かる。

2.2 再割付けのオーバーヘッド

図2の例2でも、例1のような単純Backtrack法を用いている。 α を許容するページが不足してからe2を生成したのでは、f1に既に割付けられた内容について明らかに再割付けが必要である²⁾。CLEのinstanceの数は一般的に多くなると考えられ、繰返し回数を増加させるときのこの再割付けオーバーヘッドは致命的である。

2.3 REP処理の課題

以上2例より、REP処理においては次の2つの機能

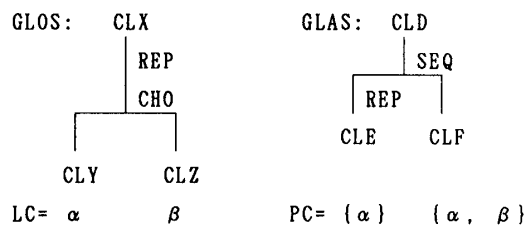


図2 例2: REP処理のオーバーヘッド

が要求されることが分かる、

☆選択的な Backtrack、

☆繰返し回数増加時の再割付け回避。

3. REP処理方式

3.1 処理方式の要点

以下の3点を基本とするREP処理方式は2.3の機能を持ち、例1、例2の問題を解決することができる。

- (1) 各繰返し構造単位に、REP+、REP- のいずれかの処理モードを設定する。
- (2) REP+ の処理モードでは、その repetitive construction factor の instance で内容が割付けられていないもの(以下「空箱の組」)の数が0になったときに、これを契機として新しい空箱の組を生成する。このとき再割付け不要である。
- (3) REP- の処理モードでは、その他の失敗を契機として、繰返し回数を減少する。

実はページ番号、ページ番号参照、recto/verso ページ区別の各ケースについては(2)の場合でも再割付けが必要になる。しかし、これらは特別のケースとし、以下考慮外とする。

3.2 処理方式の評価

☆選択的な Backtrack :

繰返し回数の不足は割付け失敗の原因となりうるが、3.1(2)の処理方式では、これを他の失敗とは

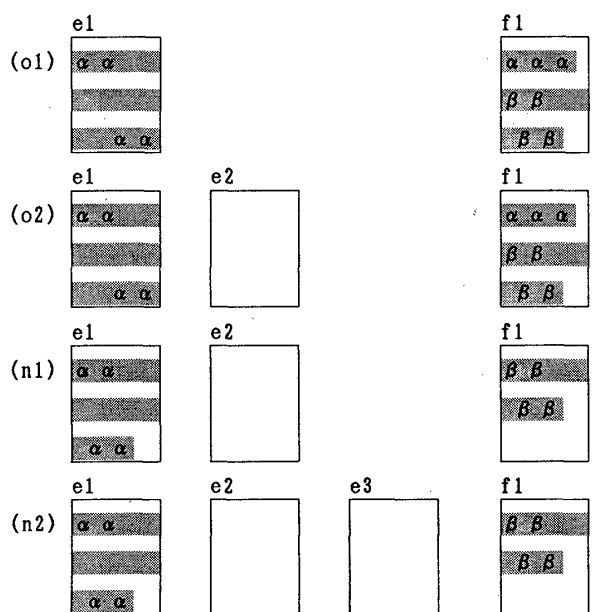


図3 繰返し回数増加の契機

別途検出することで、backtrackすべき代替情報を特定することを可能にしている。

☆繰返し回数増加時の再割付け回避 :

割付け構造修正後に再割付けが必要なのは、以前の構造で正しかった割付け判断が、新しい構造では正しくないときである。また、正しいか否か分からなければ再割付けせざるを得ない。

図3において、(o1)から(o2)へ割付け構造を変更すると、(o1)に到るまでは正しかった割付け処理上の判断が(o2)の割付け構造においては正しくない。例1、例2の方式では、このような契機で繰返し回数を増加させることがあり、再割付けが起きた。

一方、(n1)から(n2)へ割付け構造を変更しても、(n1)に到るまでに正しかった割付け処理上の判断は(n2)の割付け構造においても正しい。即ち、2つ目以降の空箱の組は、あってもなくても判断に影響しない。このような契機を捉えれば、繰返し回数を増加しても再割付けは不要である。これが3.1(2)の増加方式である。

4. まとめ

本処理方式の利点・欠点は次の通り。

- (1) 利点 ☆ REP+ を特別扱いにすることで無限ループを回避できる。

☆増加処理に伴う再割付けを回避できる。

- (2) 欠点 ★ 各REP毎の一つ以上の不要な空箱の組を作ることになる。

(2)の欠点があっても尚この方式は強力であると考える。

今後は、CHO処理に伴うオーバーヘッドの低減法³⁾と併せて、リアルタイムに最新の割付け結果が表示される対話型システムの実現を検討する。

5. 参考文献

- 1) IS08613 : Information Processing - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format
- 2) 林直樹 他 : 「ODA文書処理システムの試作(3)-割付け処理-」、第37回情報処理学会全国大会(1988)
- 3) 松平秀樹 他 : 「ODAに基づいた文書割付け処理の実現方式(2)-自動割り付けにおける効率の考察-」、第40回情報処理学会全国大会(1990)
- 4) 村上晴夫 他 : 「ODAに基づいた文書割付け処理の実現方式(1)-再試行の課題-」、第40回情報処理学会全国大会(1990)