

**ABBYY®**

# FlexiLayout 定義

---

応用編



# 1. First Found (最初の発見)

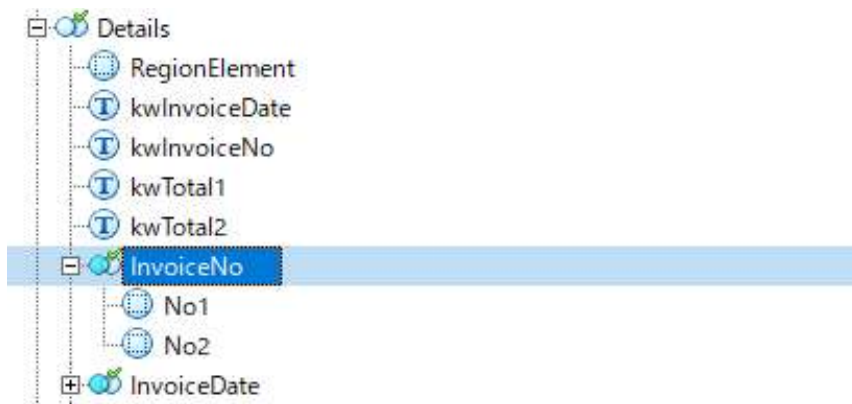
First Found (最初の発見)は、あるブロックを探索する際、複数の要素探索を(上から)順に実施、最初に発見された結果を採用する、というElement (要素)です。

## 【例】

InvoiceNoというFirst Foundエレメントに対し、No1, No2というRegion Elementを作成。

※No1は「kwInvoiceNo = ご請求#/INVOICE#」を基準に探索、No2は「kwInvoiceDate = 請求日/DATE」を基準に探索します。

探索の結果、No1が最初に発見されたため、InvoiceNo = No1となっています (No2は無視されます)。



【FlexiLayoutツリー】



【画像】

# 1. First Found (最初の発見)

ここで重要となる設定は、「基準となる要素が発見されなかった(false)場合、要素探索をしない(下記参照)」という設定です。これにより、kwInvoiceNoが発見されなかった場合、No1はスキップされ、No2の探索が開始される事になります。

(First Foundを使用せず)No1だけの設定では、OCR結果が悪くkwInvoiceNoが発見できなかった場合「請求書番号」にたどり着く事が出来ませんが、First Foundを利用する事によりバックアッププランを策定する事が可能となります(精度が向上します)。

有.	参照要素	発見	追加(+)...
<input checked="" type="checkbox"/>	Shraton.Details.kwInvoiceNo	false	削除(-)

【No1の設定】

有.	参照要素	発見	追加(+)...
<input checked="" type="checkbox"/>	Shraton.Details.kwInvoiceDate	false	削除(-)

【No2の設定】

## 2. Table (表)

Table(表)は、文字通りテーブル、表形式の要素を取得する際に使用します。ルールに則った設定([こちらの記事参照](#))をすることで、自動的にTableが検出され、出力可能となります。以下、Table使用時のメリットとなります。

### 【Tableの利点】

ページをまたぐ表を自動的に続けて読み取ることができる。

ページをまたぐ表のヘッダ行をある程度、自動的に読み飛ばすことができる。

表の末尾が空行ばかりの場合、空行は自動的に無視して読み取りを中止してくれる。

一方、Tableの検出方法は自動(ブラックボックス)のため、以下の通り、意図した通り検出できないケースや形状の制約があります。

### 【Table使用時の注意点】

- ✓ 四角い格子状、碁盤の目のような形式の表にのみ使用可能。
- ✓ 途中に大き目の空白行があると、どのように設定しても読み取れない場合がある。
- ✓ 事前認識結果の品質が悪い場合等、列と表が意図通りに取得できない場合がある。

このような場合、より柔軟に表形式の要素を取得可能なRepeating Group(繰り返しグループ)要素の使用を検討します。

## 2. Repeating Group (繰り返しグループ)

繰り返しグループ: Repeating Group の基本的な使い方を以下の画像例を用いて解説していきます。

### 【課題】

右図より、ご利用日、ご利用内容、ご請求金額をRepeating Groupとして取得します。

今後とも、引き続きお引き立て賜りますようお願い申し上げます。  
本請求書と行き違いにご送金いただきました際はご容赦ください。

		ご請求金額 TOTAL AMOUNT	¥159,100
ご利用日 DATE	ご利用内容 DESCRIPTION	ご請求金額 AMOUNT	備考 REMARKS
2015/03/26	宿泊御利用代金	41,000	
2015/03/26	宿泊御利用代金	41,000	
2015/03/27	宿泊御利用代金	77,100	
合計金額 TOTAL AMOUNT		159,100	

本書到着月の翌月末までに、記載の取引銀行にお振込下さいますようお願い致します。  
お振込手数料はお客様にてご負担くださいますようお願い致します。

## 2. Repeating Group (繰り返しグループ)

### 【ステップ1】 Repeating Groupの領域を指定

#### 1.1. 領域指定用のElementを以下(右図)の通り作成

Stasic Text: kwDate, kwDesc, kwAmount, kwRemarks, kwTotal  
 Separator: vSp\_Left, vSp\_Right1 ~ vSp\_Right4

#### 1.2. Repeating Group要素を作成し、以下の通り範囲を指定(右図赤線)



全請求書と行き違いにご注意いただきました際はご容赦ください。

ご請求日 DATE	ご請求内容 DESCRIPTION	ご請求金額 AMOUNT	ご請求 REMARKS
2015/03/26	宿泊御利用代金	41,000	
2015/03/26	宿泊御利用代金	41,000	
2015/03/27	宿泊御利用代金	77,100	
kwTotal 合計金額		TOTAL AMOUNT	159,100

本業に差支の無い限り、お振替の旨に銀行にお振替下さいませとお願い致します。

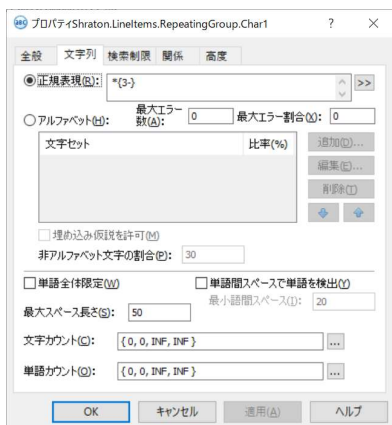
## 2. Repeating Group (繰り返しグループ)

【ステップ2】 Repeating Group内でご利用内容 (Description) 領域にある文字列を1行ずつ取得

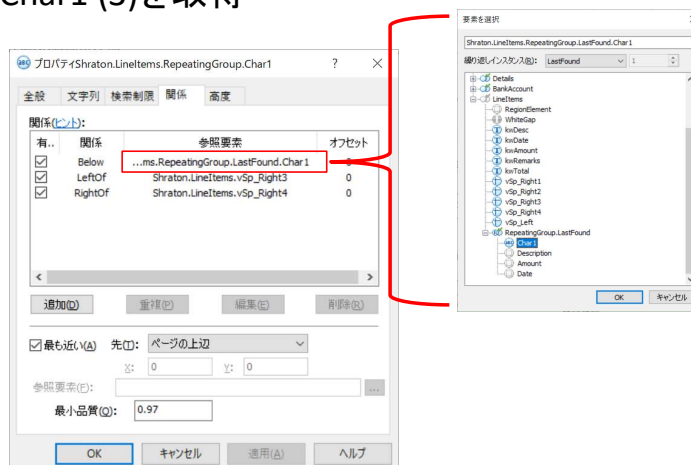
### 2.1. RepeatingGroupにChractor String (Char1)を作成



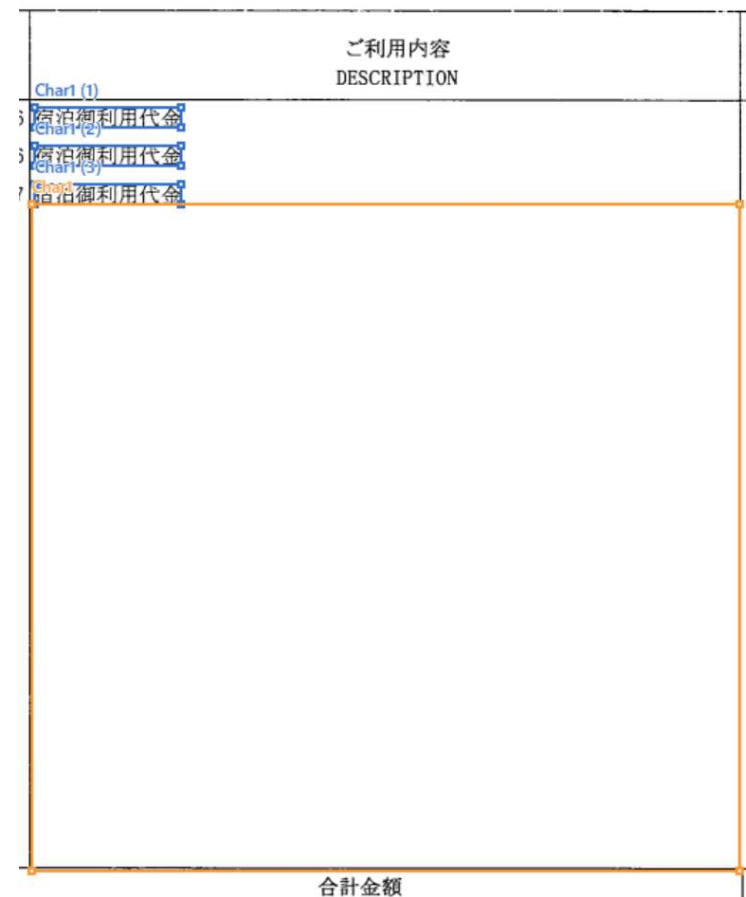
2.2. Char1のプロパティを以下の通り設定する  
→ 右図の通りChar1 (1) ~ Char1 (3)を取得



3文字以上の文字列を探索



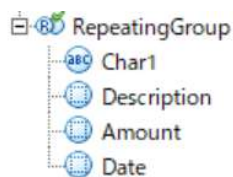
左右: vSp\_Right3とvSp\_Right4の間  
上下: 自分自身のLastFoundより下 (Below)  
最も近い: ページ上辺



## 2. Repeating Group (繰り返しグループ)

【ステップ3】 Char1を基準にDate, Description, Amountを特定(右図)

3.1. 以下のようにRegion Elementを作成



3.2. 以下のように設定(Descriptionの例)

名前(N): Description  
フルネーム(F): Shraton.LineItems.RepeatingGroup.Description  
コメント(C):  
検索制御  
○ オプション要素(O)  
○ 必須要素(B)  
○ 禁止要素(P)  
次の場合に要素を検索しない(N):  
有.. 参照要素 発見 追加...  
[x] ...peatingGroup.CurrentInstance.Char1 false 削除...  
生存戻数(L): 1

有..	関係	参照要素	オフセット
<input checked="" type="checkbox"/>	Above	...roup.CurrentInstance.Char1.Bottom	-10
<input checked="" type="checkbox"/>	Below	...ngGroup.CurrentInstance.Char1.Top	-10
<input checked="" type="checkbox"/>	LeftOf	Shraton.LineItems.kwDesc.Right	0
<input checked="" type="checkbox"/>	LeftOf	Shraton.LineItems.vSp_Right3	50
<input checked="" type="checkbox"/>	RightOf	Shraton.LineItems.vSp_Right4	0

追加(A) 重複(D) 編集(E) 削除(R)  
□ 最も近い(N) 先(C): 要素  
※: 0 Y: 0  
参照要素(E):

ご利用日 Date (1)	ご利用内容 DESCRIPTION	ご請求金額 AMOUNT Amount (1)	備考 REMARKS
2015/03/26	宿泊利用代金	41,000	
2015/03/26	宿泊利用代金	41,000	
2015/03/27	宿泊利用代金	77,100	
合計金額 TOTAL AMOUNT		159,100	

「Char1が無いと、探索しない」

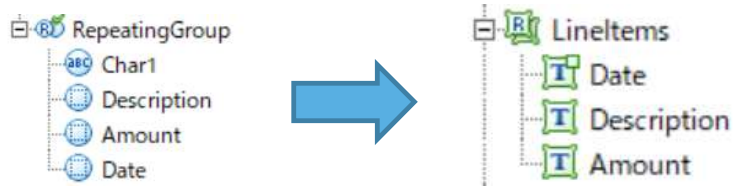
左右: vSp\_Right3 (kwDescriptionの右端)とvSp\_Right4の間  
上下: Char1の上下を基準に設定



## 2. Repeating Group (繰り返しグループ)

### 【ステップ4】 Blocks設定

4.1. 以下のようにBlocksを作成し、SearchElementsの該当要素をセット



ご利用日 Date (1)	ご利用内容 DESCRIPTION Description (1)	ご請求金額 AMOUNT (1)	備考 REMARKS
2015/03/27	宿泊御利用代金	Amount(1)000	
2015/03/27	宿泊御利用代金	Amount(1)000	
2015/03/27	宿泊御利用代金	Amount(1)000	
2015/03/27	宿泊御利用代金	77,100	
合計金額 TOTAL AMOUNT		159,100	

### 3. FlexiLayout Language (スクリプト)

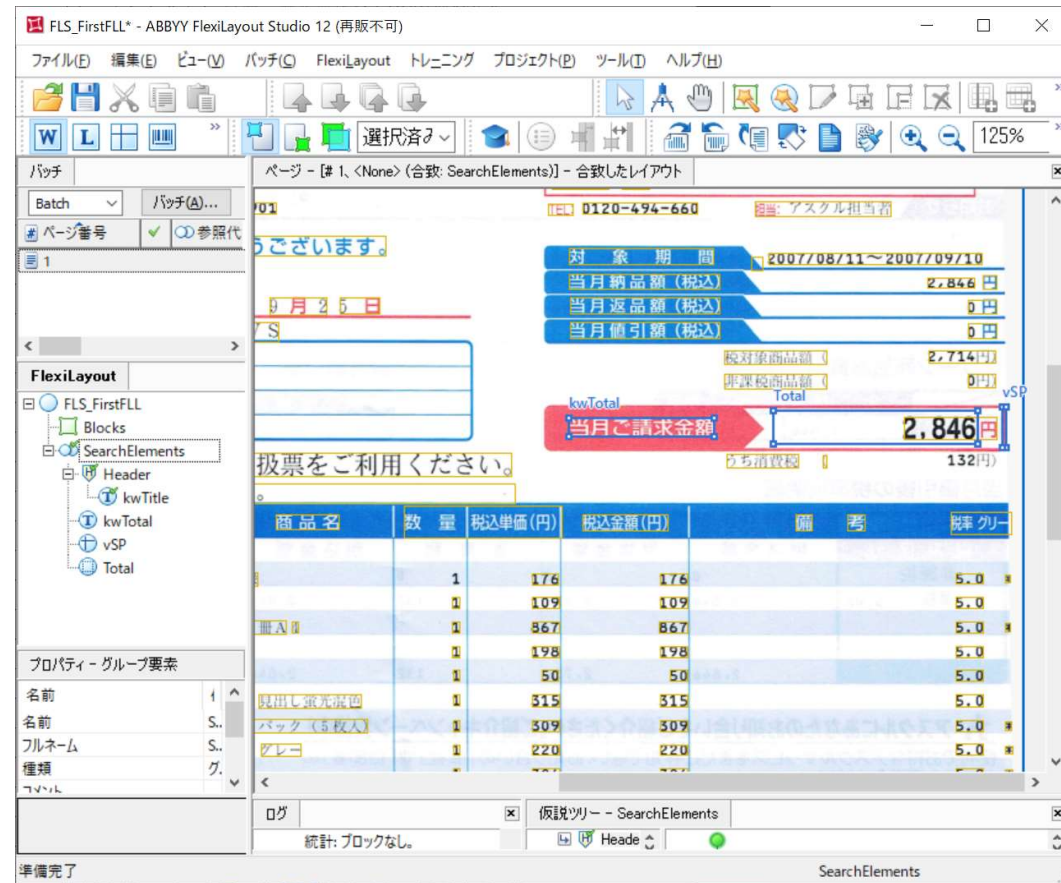


FlexiLayout Language (スクリプト) は FlexiLayout Studio 独自の言語です。FlexiLayout Studio は GUI の設定のみでも文書定義が可能ですが、GUI のみでは 100% チューニング パフォーマンスが出ないケースがあります。以下、例を挙げて説明いたします。

#### 【例】

右図の定義方針は以下の通りです。

- 取得したい項目は、Total (請求金額合計)。
- Total にたどり着く準備として、kwTotal (静的テキスト)、vSP (区切り) をそれぞれ取得。
- kwTotal, vSP からの相対位置を指定することにより、Total の位置と特定。



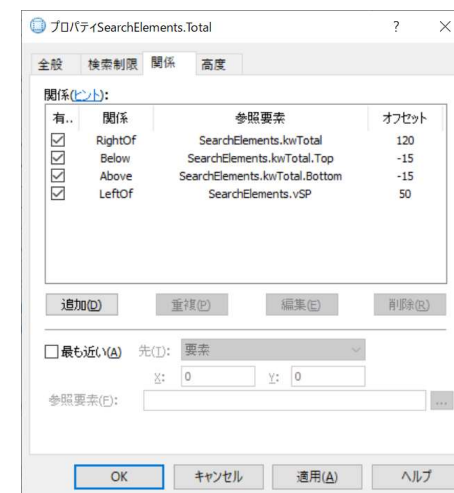
### 3. FlexiLayout Language (スクリプト)

この時、GUIのみでのアプローチは右図の通りとなります。  
GUIによる設定では、基準となる要素からの相対位置、プラス(絶対値)オフセットによって設定を行いますが、この際以下のような問題が発生します。

#### 【GUIのみの課題】

1. 事前認識結果が思わしくなく、vSP(区切り)が正しく取得できなかった場合、Totalの位置情報が正しく特定できない。
2. スキャンの設定がテンプレート画像と異なる(解像度や用紙サイズが異なる、等)場合、オフセットが意図していた幅よりずれてしまう。

これらの問題点を解決する手段として、FlexiLayout Language (FLL) の活用を検討いたします。



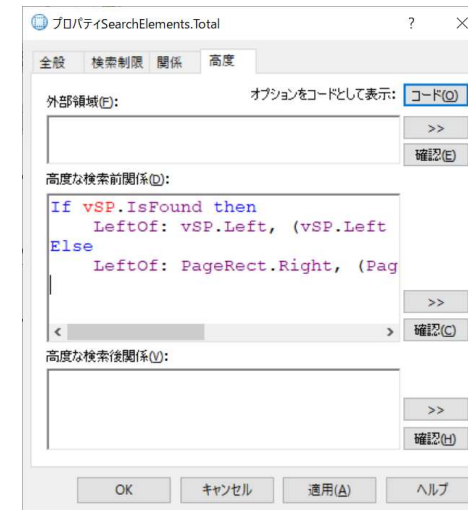
### 3. FlexiLayout Language (スクリプト)

まず1.の問題については、以下のFLL(高度な検索前関係)によって回避可能です。

```

If vSP.IsFound then
  LeftOf: vSP.Left, (vSP.Left - kwTotal.Right)*9/100*dot;
Else
  LeftOf: PageRect.Right, (PageRect.Right - kwTotal.Right)*14/100*dot;

```



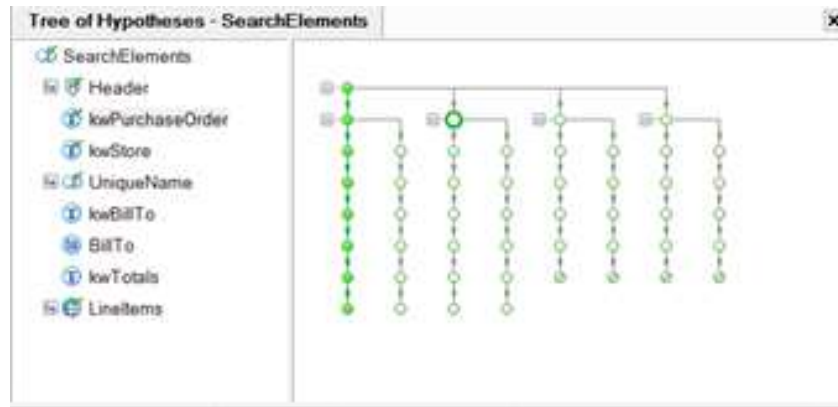
上記の方法により、vSP(区切り)が見つからなかった場合でも、Totalの位置を正しく取得する事が可能となります(PageRectは画像ページ全体の情報、FLLでしか使用できません)。

また同様に(今回は行っていませんが)kwTotalが取得できなかった場合のケアも可能です。

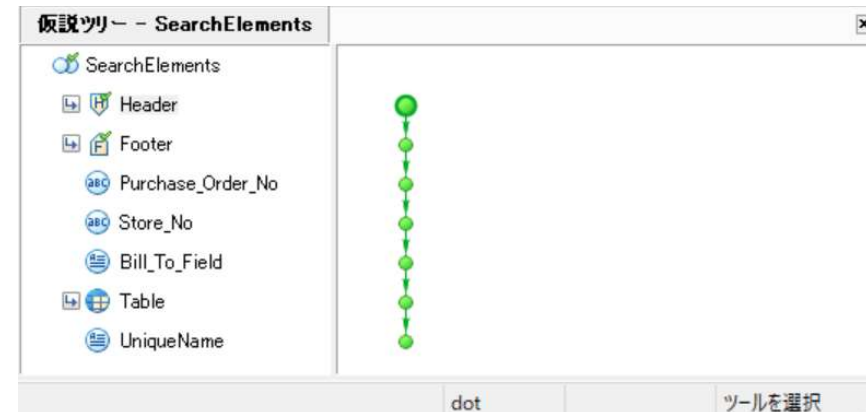
次に2.については、FLLではオフセットの指定を、要素間の相対位置(例: -kwTotal.Height\*30/100\*dot)で設定する事により、問題が解決いたします。

## 4. Hypotheses Tree (仮説ツリー)

FlexiLayout Studioで認識テストを実施すると、以下のように仮説ツリーが表示されます。



【悪い例】



【良い例】

これは、FlexiCaptureがどのように各要素を探索したのかを表しており、濃いグリーンが最終的に選択された要素、白抜きのもが選択されなかった候補を表しています。

分岐が多い方(上図左)が不安定で、分岐が少ない方(上図右)が安定しているという意味になります。

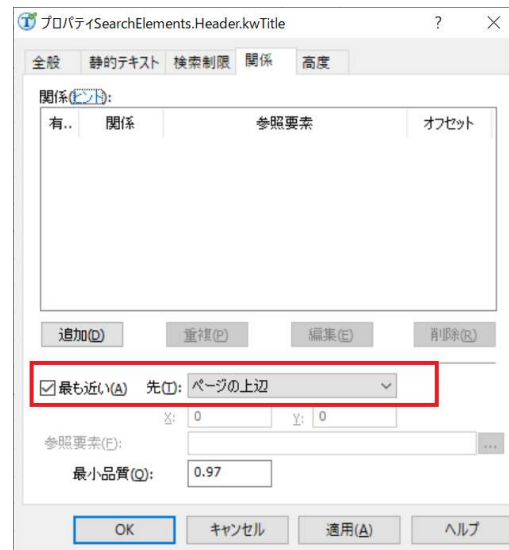
また、分岐が増えすぎると探索自体もストップし、仕分け・認識処理そのものが出来なくなってしまう(不明な文書となる)ため、注意が必要となります(各グループ要素中の分岐についても確認要)。

## 4. Hypotheses Tree (仮設ツリー)

仮設ツリーに関する基本的な考え方としては、ツリーを右図(良い例)のように1本に近づけるよう、定義する事が重要となります。

分岐をなくすためには探索結果が一意になる(複数の候補が無い)事が重要で、具体的な対策としては、関係タブのNearest(最も近い)の活用や、検索前品質設定(次ページ参照)を活用する等のアプローチがあります。

Nearest(最も近い)は以下により設定可能で、下記例では、複数候補がある際、ページの上辺にあるものを探索結果とします。



## 4. Hypotheses Tree (仮設ツリー)

以下、Hypotheses Treeに関する補足説明(高度)となります。

### 【Hypothesesに関する用語】

<p>Hypotheses (Hypotheses Quality)</p>	<p>0~1で表す、確からしさ(要素見つけた)。複数見つけた場合、分岐が走る。 分岐の数は、生存仮説数(Number of surviving hypotheses)で設定(表示される項目の内容は、FLSIに依存)。後述する、検索領域(検索前品質)と、検索後品質(高度な検索後関係)の積で品質(0~1の値)が確定する。 Hypotheses Quality = 検索前品質 x 検索後品質</p>
<p>検索前品質 (Pre-search quality)</p>	<p>検索領域(関係や、高度な検索前関係で設定)で、見つけた確からしさ(要素見つけた)。要素を見つけた場合:1(区切り要素の際、FLSが任意割当)</p>
<p>検索後品質 (Post-search quality)</p>	<p>検索後(高度な検索後関係/Advanced Post Search Relations) の処理で見つけた(絞り込めた)確からしさ(要素見つけた)。設定しない場合:1。基本、FuzzyQuality関数で制御。</p>
<p>帰無仮説 (Null hypothesis Quality)</p>	<p>Element(要素)が見つからない場合の、設定値。Optional Elements (オプション要素)の際、利用。</p>
<p>チェーンの品質 (Chain Quality)</p>	<p>現在の要素の確からしさ品質と、前の確からしさ品質との積。 以下の場合、チェーン(次の要素検索)を取りやめます。  <ul style="list-style-type: none"> <li>・ 品質が0になった場合</li> <li>・ 最初の要素から、最後の要素までの、ベストパスが無いとFLSが判断した場合。</li> </ul> </p>

# 4. Hypotheses Tree (仮設ツリー)

Hypotheses Quality(探索の確からしさ)に関する主な設定箇所:

高度な検索前関係①: エリア内による判定。基本、エリア内に要素無 : 0.97  
 Nearest: PageTop; エリア内による判定。基本、エリア内に要素有 : 1  
 区切り(Separator)の場合 : FLS 側依存 <≒1

高度な検索後関係②: 基本 FuzzyQuality 関数による、↓  
 品質 (優先順位) 設定 (設定無場合 : 1)

if not IsNull then  
 {  
 //FuzzyQuality: Rect.Top - Pa.v  
 <

検索制御  
 オプション要素③  
 無無返却の品質④: 0.97  
 必須要素⑤  
 禁止要素⑥

次の場合に要素を検索しない⑦:  
 有. 参照要素 発見 追加...  
 削除

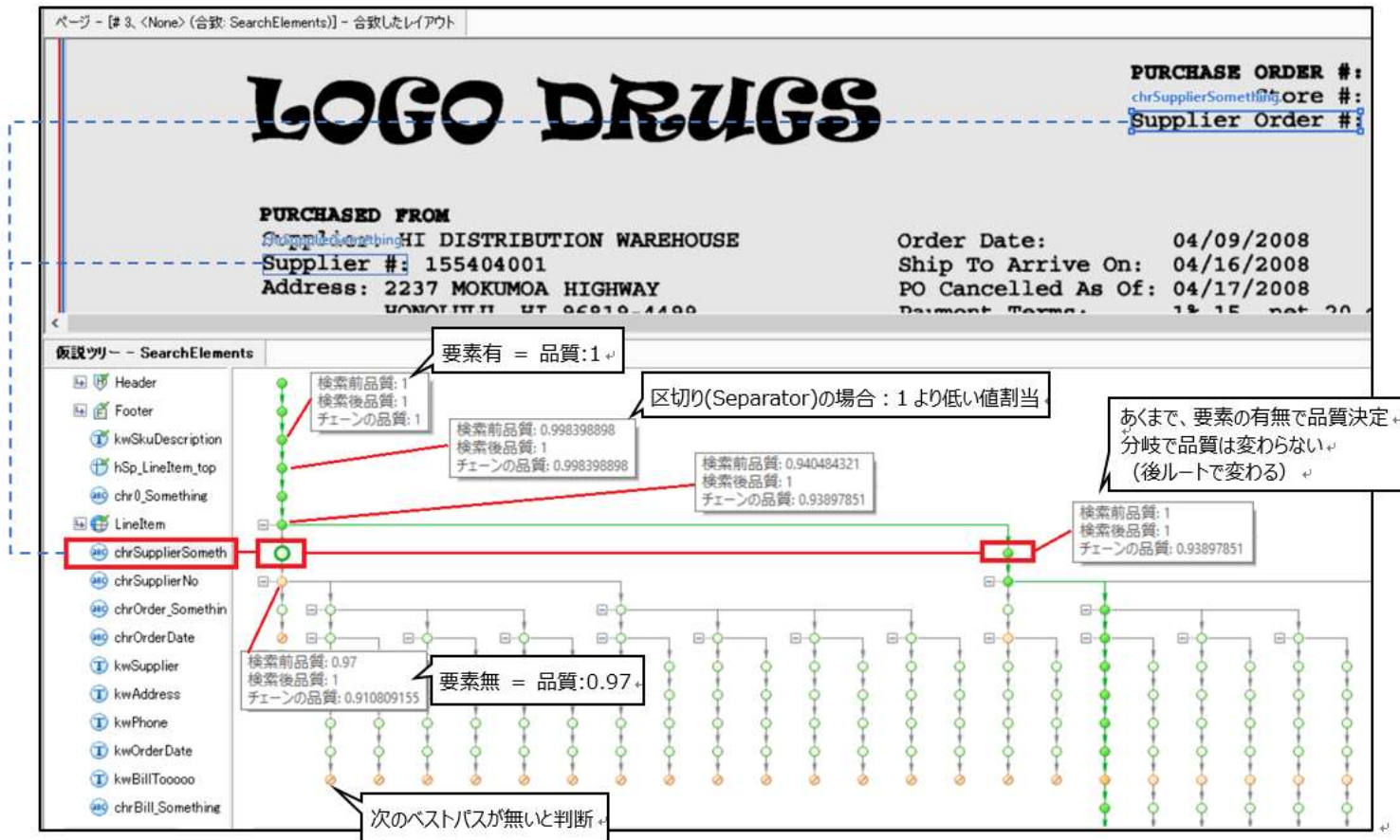
生存回数⑧: 5  
 複数要素が、見つかった場合の分岐の最大数

Hypotheses Quality(確からしさ) = 検索前品質 x 検索後品質



# 4. Hypotheses Tree (仮説ツリー)

Hypotheses Tree (仮説ツリーの分岐イメージ) と、品質内容イメージ:



# 関連記事一覧



カテゴリー	記事
最初の発見	<a href="#">FirstFound: 最初の発見の有効な利用方法</a>
Table	<a href="#">表: Table の使い方: 基本 – FlexiLayout</a>
	<a href="#">Table: 表の列を分割して読み取りたい</a>
Repeating Group	<a href="#">Repeating Group の基本的な使い方: 表をRepeating Group: 繰り返しグループを使って読み取る。</a>
	<a href="#">Table: 表の行の読み取りが途中で途切れてしまう。テーブルの末尾まで読み取れない。行を読み飛ばす。</a>
	<a href="#">RepeatingGroup: 繰り返しグループ - 格子状の形でない一定の繰り返しパターンがある表を読みみたい。</a>
	<a href="#">Repeating Group: 繰り返しグループで多種多様な形式の表を読み取る01 (行を区切る罫線なし、表の列のヘッダが黒塗りに白文字、表の角が丸い)</a>
	<a href="#">Repeating Group: 繰り返しグループで多種多様な形式の表を読み取る02 (罫線なし、レシートのような画像・サイズと品質の帳票)</a>
Hypotheses Tree	<a href="#">仮設ツリー</a>

Thank you!!