

---

## ECGI Estocástico

ECGI, tal como se ha presentado en el capítulo anterior, es una herramienta muy eficaz que ya en los experimentos piloto demostró su gran poder de modelización. Proporcionó, desde un primer momento, resultados del orden del 97% aciertos, en experimentos de reconocimiento llevados a cabo sobre dígitos hablados (monolocutor), los cuales se hallaban representados mediante cadenas de símbolos muy poco elaborados (ver capítulo 8: experimentos con corpus piloto).

Sin embargo, aunque mejorable (utilizando símbolos más elaborados, e incluso una representación continua), esta tasa de reconocimiento es aún insuficiente en aplicaciones prácticas; más cuando se aspira a resolver tareas mucho más difíciles, en las que las tasas de aciertos de los mejores métodos descienden drásticamente (multilocutor, grandes diccionarios, diccionarios difíciles, ...). Afortunadamente, existe una vía inmediata para mejorar las capacidades de modelización de las estructuras inferidas por ECGI, la cual se deriva directamente de la constatación de que los autómatas inferidos no han retenido ninguna información relacionada con la importancia relativa de las distintas (sub)estructuras inferidas, es decir, ninguna información relativa a las *frecuencias de utilización* de las distintas reglas, no terminales y terminales<sup>1</sup>. Esta, es precisamente la información incorporada por los métodos estocásticos, la cual estaría disponible si ECGI infiriera una *gramática estocástica*.

En los siguientes apartados se expone cómo se han incorporado probabilidades a las reglas de las gramáticas generadas por ECGI, y cómo se han aprovechado estas probabilidades en reconocimiento.

---

<sup>1</sup>Esto no es del todo cierto como se comprobará cuando se defina el *tráfico* y el *número de caminos*. en el capítulo 10.

## 7.1 Ambigüedad y probabilidad

En el caso de que la gramática inferida por ECGI sea no ambigua se pueden obtener, mediante el procedimiento expuesto en el capítulo 2 y con todas las garantías de estocasticidad, las probabilidades de utilización de las reglas, con sólo llevar la cuenta del número de veces que se utilizan en la generación de las muestras de  $R^+$ . Sin embargo, en la realidad las gramáticas generadas por ECGI son ambiguas (ver capítulo 9), con lo que en principio se hará necesario disponer de un número mucho mayor de muestras (o una reconsideración repetitiva de  $R^+$ ) y utilizar algún método de reestimación de probabilidades (p.e.: reestimación por Viterbi [Levinson,83] [Rabiner,89]) o también Baum-Welch) que asegure la convergencia y fiabilidad de dicha reestimación (ver también capítulo 2). Una aproximación de este tipo se ha planteado y llevado a la práctica en [Castaño,90] proporcionando resultados satisfactorios, aunque no necesariamente mejores que los aquí obtenidos.

No obstante, estos métodos, relativamente costosos, no son imprescindibles en la práctica y, como se mostrará a continuación, una aproximación no estricta (y por lo tanto subóptima) al problema permite obtener tasas de reconocimiento equivalentes (e incluso, en primera aproximación, mejores) a las de un método estrictamente correcto. En consecuencia, en todo el resto del trabajo actuaremos como si las gramáticas generadas por ECGI no fueran ambiguas, asumiendo que todos los resultados que se obtengan serán probablemente subóptimos.

## 7.2 Construcción estocástica

La utilización de la información estadística para mejorar la estructura inferida por ECGI, es decir, su empleo para realizar la búsqueda de la derivación óptima sobre la que luego se basará la construcción, tropieza con dos dificultades:

- La información frecuencial se renueva a cada muestra.
- No sólo es necesaria la información probabilística referente a las reglas de la gramática que se ha inferido, sino que también se debe disponer de la misma información sobre las demás reglas de la gramática *expandida* (sobre las reglas de error), puesto que todas ellas se utilizan para el análisis sintáctico de las muestras.

La modificación de las frecuencias de uso de las reglas a cada muestra implica que, tras cada de ellas, es necesario renormalizar toda la gramática (es decir, convertir las nuevas frecuencias en probabilidades). Aunque ello, en contra de lo que pueda parecer, no resulta excesivamente gravoso comparado con la complejidad del análisis sintáctico corrector de errores, se

ha renunciado (en primera aproximación) a esta posibilidad. Por lo tanto, en todo el resto de este trabajo, el modelo estocástico se estudiará únicamente con miras a mejorar las capacidades de reconocimiento de ECGI, **no utilizándose en ningún caso las probabilidades en el proceso de aprendizaje de la estructura del modelo**. Conviene notar que ésta es la sistemática más frecuente en reconocimiento de formas: estimación de probabilidades de un modelo estructural fijo, obtenido sin hacer uso alguno de estas probabilidades.

Una consecuencia de esta decisión, entre otras, es la de imposibilitar la integración aprendizaje-reconocimiento en el caso del aprendizaje continuo, al emplearse algoritmos de optimización distintos en cada etapa.

Por otra parte, hay que notar que el procedimiento de renormalizar a cada muestra no es el único factible si se desea utilizar la información estocástica durante la inferencia de la estructura del modelo generado por ECGI. Existe otra posibilidad intermedia, estudiada en [Castaño,90], que consiste en particionar (o reutilizar)  $R^+$  para reestimar las probabilidades. En vez de proceder a una renormalización a cada muestra, se reconoce y construye con las probabilidades estimadas en un paso anterior hasta reunir cierto número de muestras. En ese momento es cuando se procede a la renormalización. Los resultados obtenidos por M.A.Castaño muestran una ligera pérdida (no explicada) en la eficiencia del reconocimiento, pero muestran la factibilidad de la idea.

En cuanto al segundo de los puntos arriba mencionados, que destaca la necesidad de disponer del modelo estocástico también para la gramática expandida, es fácil comprobar que ello es una dificultad inherente al proceso *incremental* de aprendizaje de ECGI. Renunciando a una aproximación incremental, sería posible prescindir de las probabilidades asociadas a las reglas de error. En efecto, al estar incluido el conjunto de muestras  $R_+$  en el lenguaje de la gramática inferida, se pueden estimar las probabilidades de las reglas a partir de  $R_+$ , puesto que no resulta necesario expandir la gramática para realizar el análisis sintáctico.

De todas maneras, y como se verá en los apartados siguientes, es posible estimar de manera fiable un modelo de error estocástico y muy eficaz.

### 7.3 Reconocimiento estocástico

La estimación de las probabilidades de las reglas de error de la gramática expandida de una gramática  $G=(N,V,P,S)$ , supone una serie de dificultades que se agudizan más, si cabe en el caso particular de ECGI. En efecto, la misma idea en que se basa el método hace imposible estimar realmente la probabilidad de una regla de error concreta: siempre que alguna es utilizada,

el proceso de construcción de EGCI se encarga inmediatamente de anularla generando las reglas mínimas imprescindibles para que dicha regla de error no sea necesaria para reconocer (generar) la nueva muestra (y otras similares).

Por otra parte, aún en el caso de que fuera posible, el enorme número de reglas de error<sup>2</sup>:  $(2 \cdot |V| + 1) \cdot P$ , exigiría disponer de un número prohibitivo de muestras de aprendizaje para poder efectuar una estimación aceptable de las probabilidades correspondientes.

Para soslayar ambos problemas, no queda más remedio que renunciar efectivamente a la estimación individual de *todas* las probabilidades de las reglas de error, y recurrir a una aproximación más "global", agrupando dichas reglas por *clases* y asignar a cada regla la probabilidad de su clase. Se hace entonces la asunción de que **la probabilidad de una regla de error no depende de los no terminales asociados a dicha regla** (de la "posición" de la regla). En estas condiciones, en la probabilidad de una regla de error sólo interviene el tipo de ésta (inserción, borrado o sustitución) y el símbolo (no terminal) que tiene asociado, por lo que se pueden agrupar las reglas de error en  $2 \cdot |V| + |V|^2$  clases, siendo éste el número de probabilidades que efectivamente hay que estimar.

Al no estar estas probabilidades asociadas a una regla en concreto, se puede considerar que una estimación de ellas se puede obtener a partir del conteo de la ocurrencia *de las de su clase* durante la construcción y/o inferencia de la gramática a partir de  $R^+$  (incluso aunque muchas de las reglas que hayan intervenido en el conteo queden anuladas por el proceso de construcción). En consecuencia, EGCI rellena durante la construcción una tabla, que denominaremos *Tabla de Sustitución* (al considerar inserciones y borrados como sustituciones en las que interviene el símbolo nil  $\epsilon$ ), que contiene la frecuencia de sustitución de cada uno de los terminales por otro:  $f_s(b,a)$  (sustitución de 'a' por 'b'),  $f_b(a)$  (borrado de 'a') y  $f_i(b)$  (inserción de 'b').  $f_s(a,a)$  nos da la frecuencia de las reglas NO erróneas que tienen asociado el símbolo 'a'.  $f(\epsilon,\epsilon)$  se define 0.

La tabla de sustitución será por lo tanto una matriz de  $(|V|+1)^2$  elementos tales que (En tabla 7.1 se muestra un ejemplo):

$$T_s(b,a) = f_s(b,a) \quad \forall a,b \neq \epsilon; \quad T_s(b,\epsilon) = f_i(b); \quad T_s(\epsilon,a) = f_b(a); \quad T_s(\epsilon,\epsilon) = 0$$

El conteo de estas frecuencias se realiza simultáneamente con el análisis de la derivación óptima durante la construcción, con sólo añadir al contador de

---

<sup>2</sup> Como se verá más adelante, y debido a las particularidades del algoritmo ViterbiCorrector, el número de reglas de error es en realidad menor al presentado: todas las reglas de inserción de las reglas con el mismo no terminal a la izquierda son comunes.

los errores de cada clase aquellos errores que aparecen en dicha derivación. En la implementación práctica, debido a que se genera un autómata de estados etiquetados (LAS, ver capítulo 2) y a que ECGI añade los símbolos inicial y final (ver capítulo 6), la tabla incorpora también a estos últimos, con frecuencias  $f_s(\sim, \sim)$  y  $f_s(\bullet, \bullet)$  iguales al número de cadenas de  $R^+$ , puesto que nunca se insertan, sustituyen o borran.

**Tabla 7.1** Tabla de errores de sustitución obtenida por ECGI durante la inferencia de un modelo para el dígito hablado /cuatro/ a partir de 150 muestras. Los terminales son  $V=\{E,I,K,N,O,R,S,T,U,Z,e,n,o,r,v,\#\sim\bullet/\}$ . Obsérvese que además están incluidos el símbolo inicial y final y que el símbolo  $\epsilon$  se representa como /. El símbolo # es una provisión por si apareciera un símbolo desconocido, y su frecuencia debe de ser siempre 0.

	E	I	K	N	O	R	S	T	U	Z	e	n	o	r	v	#	~	.	/
E	55	0	1	0	5	4	0	1	0	2	0	0	0	3	8	0	0	0	13
I	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
K	0	0	32	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	2
N	0	0	0	5	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
O	4	0	2	0870	0	0	0	1	0	0	0	0	0	0	3	0	0	0	8
R	0	0	0	0	0	67	0	3	0	0	0	0	0	0	0	0	0	0	3
S	1	0	0	0	0	0	22	1	0	1	0	0	0	0	0	0	0	0	0
T	0	0	0	0	0	0	0729	2	3	0	3	1	0	0	0	0	0	0	8
U	0	0	1	2	7	0	0	10946	0	0	0	8	0	1	0	0	0	0	3
Z	2	0	0	0	1	1	2	2	0275	1	0	0	3	1	0	0	0	0	7
e	2	0	0	1	0	0	0	0	0	0	11	0	0	0	0	0	0	0	2
n	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
o	0	0	1	0	4	0	0	2	2	0	0	0348	0	2	0	0	0	0	5
r	3	0	0	0	0	2	0	0	0	1	0	1	0	63	2	0	0	0	3
v	2	0	1	0	4	4	0	2	1	2	2	0	1	1484	0	0	0	0	12
#	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
~	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0	0
.	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	150	0
/	11	0	2	0	8	3	1	6	6	3	2	0	4	0	2	0	0	0	0

Aún con la drástica reducción aplicada, el número de muestras es relativamente pequeño respecto al de parámetros a estimar, por lo que gran parte de las frecuencias en la tabla de sustitución de cada gramática (autómata) son de valor muy reducido o nulo. Ello conduce a una estimación extremadamente pobre de las probabilidades. Para mejorar dicha estimación, se procede a *sumar* las tablas de sustitución de todas las clases a reconocer, reduciéndolas a una *única tabla de sustitución*, común a todas ellas. Dicho de otro modo, **la extensión estocástica de ECGI supone que las probabilidades de los distintos tipos de reglas de error son las mismas para todas las clases a reconocer**, e iguales a la probabilidad de cada tipo en el conjunto de las clases. Esto es conveniente a menos de disponer de un muy elevado número de muestras: en [Prieto,88], experimentos de reconocimiento de letras habladas proporcionaron en general una tasa de reconocimiento de 0.5% mejor utilizando las tablas de substitución promediadas (p.e. 65% en lugar de 64.5%) (ver apéndice A).

## 7.4 La estocasticidad de la gramática expandida

### 7.4.1 Las probabilidades de deformación

Una vez inferida por ECGI una gramática, es necesario obtener, a partir de la correspondiente tabla de sustitución y de las frecuencias de uso en  $R^+$  de las reglas, las probabilidades de éstas y las de las reglas de la gramática expandida; todo ello asegurando la estocasticidad, tanto de la gramática sin expandir como de la expandida.

Utilizando las definiciones dadas en el capítulo 2, si el no terminal  $A$  toma parte de las  $m$  reglas  $A \rightarrow x_k B_k, k=1..m$ ; de la gramática característica y  $n_k$  es el número de veces que se utiliza una de ellas  $r_k$  en  $R^+$  basta calcular:

$$p(r_k) = \frac{n_k}{\sum_{j=1..m} n_j}$$

para obtener su probabilidad, asegurando simultáneamente la estocasticidad de la gramática:

$$\sum_{k=1..m} p(r_k) = 1$$

En el caso de la gramática expandida, para cada una de estas reglas hay que añadir las reglas de error de sustitución (con probabilidad de deformación  $p_s(b | x_k) \forall b \in V$ ), de borrado ( $p_b(\epsilon | x_k)$ ) y de inserción ( $p_i(b | x_k) \forall b \in V$ ). A partir de la tabla de sustitución, que nos proporciona las frecuencias  $f_s(b, a)$ ,  $f_b(a)$  y  $f_i(b)$ , es directo obtener una estimación de cada una de estas probabilidades:

$$p_s(b | x_k) = \frac{f_s(b, x_k)}{N_{x_k}}; \quad p_b(\epsilon | x_k) = \frac{f_b(x_k)}{N_{x_k}}; \quad p_i(b | x_k) = \frac{f_i(b)}{N_{x_k}};$$

$$N_{x_k} = \sum_{\forall b \in V} f_s(b, x_k) + f_b(x_k) + \sum_{\forall b \in V} f_i(b)$$

que cumple la condición para que las probabilidades de deformación sean consistentes.

Sin embargo, esta estimación no es válida para el caso del ECGI. Un primer signo de ello se tiene si se observa que, debido a que en la tabla de

sustitución las frecuencias de inserción  $f_i(b)$  no dependen del símbolo anterior (el de la regla), la expresión de  $P_i(b | x_k)$  depende de  $x_k$  sólo a través de la normalización. Ello obviamente es consecuencia directa del modelo de error utilizado, y se refleja en el algoritmo ViterbiCorrector, el cual, al construir las reglas de error en el trellis, únicamente añade una regla de inserción por cada símbolo y **por cada no terminal a la derecha** (y no por cada regla, como obligaría un modelo de error más complejo y que permitiría una dependencia real de  $x_k$ ). Es decir, en el trellis se tendrán las siguientes reglas de error para las  $m$  reglas  $A \rightarrow x_k B_k$ ,  $k=1 \dots m$ :

$$\begin{array}{lll} A \rightarrow b B_k & \forall b \in V; b \neq x_k; k=1 \dots m & \text{(sustitución)} \\ A \rightarrow B_k & k=1 \dots m & \text{(borrado)} \\ A \rightarrow b A & \forall b \in V & \text{(inserción)} \end{array}$$

Esta peculiaridad del modelo de error utilizado por ECGI<sup>3</sup>, impide que se pueda utilizar el hecho de que las probabilidades de deformación sean consistentes como garantía de la estocasticidad de la gramática expandida (ver capítulo 2), lo que hace necesario adaptar la condición a este caso particular:

Recuérdese que si  $G$  es *estocástica* y el no terminal  $A$  toma parte en las  $m$  reglas  $r_k = A \rightarrow x_k B_k$  de probabilidad  $p(r_k)$ ,  $k=1, \dots, m$ ; se cumple:

$$\sum_{k=1 \dots m} p(r_k) = 1$$

Al construir  $G^e$ , para cada una de estas reglas asociadas a  $A$ , hay que añadir las reglas de error de sustitución (con probabilidad de deformación  $p_s(b | x_k) \forall b \in V$ ) y la de borrado ( $p_b(\epsilon | x_k)$ ) correspondientes; y para todas ellas en conjunto hay que añadir las reglas de inserción ( $p_i(b) \forall b \in V$ ). La probabilidad de una regla de sustitución o borrado se puede entonces definir como la probabilidad combinada de que se produzca el error en cuestión y de que se utilice la regla original  $r_k$ ; es decir, la probabilidad de una regla de sustitución sería  $p_s(b | x_k) \cdot p(r_k)$  y la de una regla de borrado  $p_b(\epsilon | x_k) \cdot p(r_k)$ , mientras que la probabilidad de inserción es independiente de la regla  $r_k$  aplicada. La suma de todas estas reglas deberá ser la unidad para poder asegurar la estocasticidad de  $G^e$ , con lo que la condición se puede escribir (recuérdese que  $p_s(x_k | x_k) \cdot p(r_k)$  es la probabilidad de la regla NO errónea):

---

<sup>3</sup> Modelo, por otro lado, muy utilizado en análisis sintáctico corrector de errores en gramáticas regulares.

$$\sum_{k=1..m} p(r_k) \cdot \left( \sum_{\forall b \in V} p_s(b | x_k) + p_b(\varepsilon | x_k) \right) + \sum_{\forall b \in V} p_i(b) = 1$$

En los siguientes apartados se expone el método que emplea ECIG para estimar las probabilidades de deformación.

### 7.4.2 Estocasticidad según ECIG

ECIG separa totalmente las probabilidades de las reglas de inserción de las de las demás reglas de error. Para ello, se define la probabilidad de inserción de un símbolo 'b', como:

$$p_i(b) = \frac{f_i(b)}{\sum_{\forall a, b \in V} f_s(a, b) + \sum_{\forall a \in V} f_b(a) + \sum_{\forall b \in V} f_i(b)}$$

es decir, como la relación entre el número de veces que se ha insertado 'b' en  $R_+$  y el número total de errores y no errores que se han producido en el mismo  $R_+$ .

Las otras probabilidades de error se definen siguiendo la filosofía usual, pero aplicando un factor corrector  $(1-P_i)$  para asegurar la estocasticidad:

$$p_s(b | x_k) = (1-P_i) \cdot \frac{f_s(b, x_k)}{N_{x_k}}; \quad p_b(\varepsilon | x_k) = (1-P_i) \cdot \frac{f_b(x_k)}{N_{x_k}};$$

$$N_{x_k} = \sum_{\forall b \in V} f_s(x_k, b) + f_b(x_k)$$

$$P_i = \sum_{\forall b \in V} p_i(b)$$

$P_i$  representa la *probabilidad de inserción de cualquier símbolo* en cualquier posición de la gramática, es decir la probabilidad de que se utilice una regla de inserción.  $(1-P_i)$  obviamente es la probabilidad de que se emplee cualquier otro tipo de regla. Nótese que  $P_i$  también se puede obtener directamente como la relación entre el número total de inserciones  $N_i$  y el número total de errores  $N_e$  en  $R_+$ :

$$P_i = \frac{N_i}{N_e}$$

$$N_i = \sum_{\forall b \in V} f_i(b); \quad N_e = \sum_{\forall a, b \in V} f_s(a, b) + \sum_{\forall a \in V} f_b(a) + \sum_{\forall b \in V} f_i(b)$$



Es fácil comprobar que con estas definiciones la estocasticidad se cumple:

$$\sum_{k=1..m} p(r_k) \cdot \left( \sum_{\forall b \in V} p_s(b | x_k) + p_b(\epsilon | x_k) \right) + \sum_{\forall b \in V} p_i(b) =$$

$$\sum_{k=1..m} p(r_k) \cdot (1 - P_i) + P_i = 1 \cdot (1 - P_i) + P_i = 1$$

En resumen, y utilizando las definiciones anteriores, las reglas de error utilizadas por ECGI para el análisis sintáctico corrector de errores estocástico, con sus probabilidades, se derivan de la gramática característica a partir de cada grupo de m reglas  $A \rightarrow x_k B_k$   $k=1...m$  con el mismo no terminal A a la izquierda, de la siguiente manera:

$$p(A \rightarrow b B_k) = p(r_k) \cdot p_s(b | x_k); \quad \forall b \in V; k=1...m \quad (\text{substitución y no error})$$

$$p(A \rightarrow B_k) = p(r_k) \cdot p_b(x_k) \quad k=1...m \quad (\text{borrado})$$

$$p(A \rightarrow b A) = p_i(b) \quad \forall b \in V \quad (\text{inserción})$$

### 7.4.3 En la práctica

Realizando una pequeña conversión, es posible escribir de otra manera la probabilidad  $p_i(b)$  de inserción del símbolo 'b':

$$p_i(b) = \frac{f_i(b)}{N_e} = \frac{N_i}{N_e} \cdot \frac{f_i(b)}{N_i} = \frac{N_i}{N_e} \cdot \frac{f_i(b)}{N_i} = P_i \cdot \frac{f_i(b)}{N_i} = P_i \cdot p_{ri}(b)$$

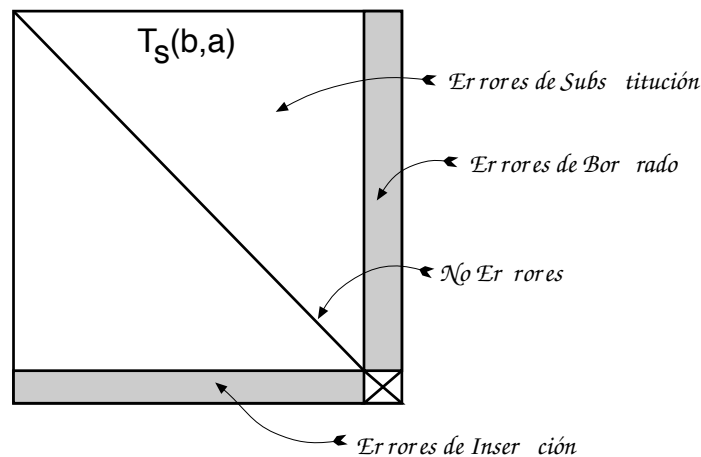
con lo que podemos ponerla en función de la probabilidad de inserción de cualquier símbolo y de la *probabilidad relativa de inserción* del símbolo 'b' (relación entre la frecuencia de inserción de 'b' y el número total de inserciones):

$$p_{ri}(b) = \frac{f_i(b)}{N_i}$$

Esto permite un tratamiento más unificado de los valores en la tabla de sustitución  $T_s$ . La obtención de las probabilidades a partir de  $T_s$  implica simplemente normalizar la tabla por líneas (figura 7.1): a cada elemento se le asigna el valor resultante de dividir el valor original por la suma de los valores en la misma línea:

$$T_{ps}(b,a) = \frac{T_s(b,a)}{\sum_{\forall x} T_s(x,a)} \begin{cases} p_s(b|x) = \frac{f_s(b,x)}{N_x} & a \neq \varepsilon \\ p_b(a) = \frac{f_b(a)}{N_x} & b = \varepsilon \\ p_{ri}(b) = \frac{f_i(b)}{N_i} & a = \varepsilon \end{cases}$$

$$N_x = \sum_{\forall b \in V} f_s(x,b) + f_b(x); \quad N_i = \sum_{\forall b \in V} f_i(b)$$



**Figura 7.1** Estructura de  $T_s(b,a)$  y  $T_{ps}(b,a)$ , tabla de frecuencias de error y tabla de probabilidades de error respectivamente.

Nótese que en la implementación real se ha permitido que los símbolos inicial y final del LAS intervengan en la normalización, no teniendo ello importancia: todos los valores que tienen asociados son 0 excepto los de la diagonal. Sólo queda afectada la estimación de  $P_i$  ya que se contabiliza el número de no errores asociado a ambos símbolos, lo que rebaja ligeramente la probabilidad relativa de una inserción.

Debido al número limitado de cifras significativas de las que dispone un calculador real, y a que en el mismo la rapidez de la adición es usualmente muy superior a la del producto, normalmente se realiza todo el cálculo utilizando los logaritmos de las probabilidades (esto, en su caso, también permitiría trabajar únicamente con enteros):

$$T_{Is}(b,a) = -\log(T_{ps}(b,a))$$

siendo esta pues la tabla que se utilizará en la realidad. También se ha procedido a un cambio de signo (por comodidad: son todos negativos), y se tomará la precaución de saturar al máximo número representable cuando el argumento del logaritmo es 0.

El mismo proceso se aplica a las probabilidades de las reglas  $P(r)$ . La contabilización de las frecuencias  $n_k$  de las reglas  $r_k=(A \rightarrow x_k B_k)$  se realiza también al recorrer la derivación óptima, simultáneamente con la contabilización de las frecuencias de utilización de los no terminales (estados)  $f(A)$ . Como todas las cadenas que hacen uso de un estado hacen uso de alguna de las transiciones que salen de ese estado, las probabilidades de las reglas se pueden calcular directamente como:

$$p(r_k) = \frac{n_k}{f(A)}$$

Resumiendo todo lo dicho hasta ahora, la relación de recurrencia que emplea ECGI durante el reconocimiento de una cadena muestra  $\alpha=a_1, \dots, a_n$  para calcular la derivación óptima (ver capítulo 5, algoritmo `ViterbiCorrector`), se puede escribir como<sup>4</sup>:

$$D(A_j, t) = \min \begin{cases} D(B_k, t-1) + L_{ni} + T_s(a_t, x_k) + L(B_k \rightarrow x_k A_j) & k=1 \dots m \\ D(B_k, t) + L_{ni} + T_s(\varepsilon, x_i) + L(B_k \rightarrow x_k A_j) & k=1 \dots m \\ D(A_j, t-1) + L_i + T_s(a_t, \varepsilon) & \end{cases}$$

$$L_{ni} = -\log(1-P_i); \quad L_i = -\log(P_i); \quad L(r) = -\log(p(r));$$

donde  $D(A, t)$  representa el coste acumulado en el vértice correspondiente al estado  $A$  y al terminal  $t$  de la cadena en el trellis del análisis sintáctico. Se ha minimizado al tratarse de los logaritmos *negados* de las probabilidades a maximizar.

Nótese que se ha utilizado la representación *hacia atrás* (se miran los  $m$  predecesores de  $A_j$  en lugar de los sucesores). Todo el razonamiento es absolutamente análogo para este caso, y como ya se ha mencionado, es la representación que utiliza la presente implementación de ECGI.

En el capítulo 9 se presenta una tabla resumen de experimentos, que permite comprobar como la aportación de la información estadística mejora sensiblemente las prestaciones de ECGI en reconocimiento de formas. En el mismo capítulo se efectúan también una serie de simplificaciones sobre el modelo aquí presentado, con el fin de comprobar empíricamente cuál es la importancia relativa de la información estadística que aportan las frecuencias de las reglas de la gramática sin expandir y las frecuencias de los distintos errores del modelo de error.

---

<sup>4</sup> En la práctica  $L_{ni} + L(B_k \rightarrow x_k A_j)$  se precalculan antes de la optimización, y lo mismo podría hacerse con  $L_i + T_s(b, \varepsilon)$  para utilizar directamente el  $-\log(p_i(b))$ .

## 7.5 Sólo sustitución

Al igual que en el caso no estocástico en el caso estocástico se ha estudiado la posibilidad de utilizar sólo errores de sustitución en reconocimiento. Si no se permiten los errores de inserción ni de borrado los problemas de estocasticidad se simplifican grandemente. Si consideramos que las probabilidades de inserción y borrado son nulas (y por lo tanto ignoramos sus respectivas frecuencias), se podrán obtener las probabilidades de deformación con el procedimiento usual. Para las  $m$  reglas  $A \rightarrow x_k B_k$ ;  $k=1..m$ :

$$p_s(b | x_k) = \frac{f_s(b, x_k)}{N_{x_k}}; \quad p_b(\varepsilon | x_k) = 0; \quad p_i(b | x_k) = 0;$$

$$N_{x_k} = \sum_{\forall b \in V} f_s(x_k, b)$$

estando asegurada la estocasticidad:

$$\sum_{i=1..m} p(r_i) \cdot \left( \sum_{\forall b \in V} p_s(b | x_i) \right) = 1$$

En la práctica, basta poner a cero las frecuencias de borrado en  $T_s$  y obtener  $T_{1s}$  de la misma manera que anteriormente. En la programación dinámica no se generarán las transiciones de borrado ni las de inserción.

Hay que notar que los heurísticos, adoptados para obtener un resultado aunque la cadena muestra sea más larga que la más larga de la gramática o más corta que la más corta (ver capítulo 6), no garantizan la estocasticidad si llegan a aplicarse (¡equivalen a aceptar una cadena que no está en la gramática expandida!). Obviamente ello es relativamente poco importante, considerando la ya gran "informalidad" del mismo heurístico.

Los resultados comparativos entre el modelo de error completo y el de sólo sustitución, en el caso no estocástico y estocástico se muestran en el capítulo 9.