Procesamiento del Lenguaje Natural, núm. 35 (2005), pp. 51-57 recibido 29-04-2005; aceptado 01-06-2005

Construcción y minimización eficiente de transductores de letras a partir de diccionarios con paradigmas*

Sergio Ortiz Rojas, Mikel L. Forcada, Gema Ramírez Sánchez
Universitat d'Alacant
Grup Transducens del Departament de Llenguatges i Sistemes Informàtics
{sortiz,mlf}@dlsi.ua.es, gema@internostrum.com

Resumen: En este artículo se presenta un modelo de gestión de diccionarios basado en paradigmas para construir procesadores léxicos. Para ello, primero se muestran algunos ejemplos que permiten poner de manifiesto la potencia expresiva del modelo presentado y el amplio abanico de lenguas al que se puede aplicar este sistema. A continuación se explica un método para construir eficientemente transductores de letras a partir de los diccionarios aprovechando el uso de paradigmas. Finalmente se presentan los resultados que se han obtenido con el sistema implementado.
Palabras clave: procesamiento léxico, transductores de estados finitos, diccionarios, paradigmas

Abstract: This paper introduces a model of dictionary management to build lexical processors based on paradigms. First, examples are given to show the expressivity of this model and that it can be applied to a wide variety of languages. Next, a method is explained that allows for an efficient construction of letter transducers extracted from dictionaries by taking advantage of the use of paradigms. Finally, the result that has been obtained with the implemented system is presented.
Keywords: lexical processing, finite-state transducers, dictionaries, paradigms

1. Introducción

Una de las tareas que hay que realizar en el diseño y la implementación de sistemas de procesamiento léxico es la construcción de procesadores léxicos eficientes a partir de de los datos lingüísticos.

En particular, los procesadores léxicos que se describen aquí han sido usados para transformaciones léxicas tales como el análisis morfológico, la generación morfológica y la traducción palabra por palabra de formas léxicas. El *análisis morfológico* de una palabra es la obtención a partir de su *forma superficial* de todas las *formas léxicas* (constituidas por un lema y atributos morfológicos) que le correspondan dado un diccionario. La *generación morfológica* es el proceso inverso: dada una forma léxica, genera su forma superficial. La traducción palabra por palabra de formas léxicas consiste en hacer corresponder a una forma léxica de una palabra en una lengua otra forma léxica de otra palabra en otra lengua. Esta última operación es crucial para construir traductores automáticos.

Las palabras pueden tener una —es el caso de las *palabras invariantes*— o más formas. Las variaciones en las palabras reciben varios nombres atendiendo a su naturaleza. Pueden ser *derivaciones*, cuando una palabra se combina con otras o con morfemas que modifican su significado (p.e. *ameno* y *amenizar*, *presidente* y *vicepresidente*, etc.); *flexiones*, si se trata de las modificaciones gramaticales que ocurren en nombres, adjetivos y verbos en lenguas como las indoeuropeas (p.e. *lleva*, *llevó*, etc.); *aglutinaciones*, si se trata de afijos acumulados en ciertas palabras que afectan a todo un sintagma desde el punto de vista gramatical, como pasa en ciertas lenguas como el turco o el vasco (p.e. *urdin*, *urdina*, *urdinarena*, que en español se puede corresponder con *azul*, *el azul*, *el del azul*, respectivamente); o cualquier otro tipo de variaciones *ortográficas* que pueden ocurrir en cualquier lengua.

Las *regularidades* observadas en el procesamiento de estas modificaciones se pueden agrupar, por conveniencia, al construir diccionarios morfológicos (tanto para el análisis como para la generación), para evitar tener que escribir todas las formas de cada palabra. Desde el punto de vista de la gestión de

* Trabajo financiado por FIT-340101-2004-3.

diccionarios es interesante tener almacenada la flexión de las palabras en *paradigmas de flexión* identificados por un lado y los lemas que se flexionan por otro. Esto permite que introducir una palabra que se flexiona se reduzca a indicar el lema e identificar la flexión entre los paradigmas previamente definidos, o definir una flexión nueva que podrá servir para introducir otras palabras que presenten esa misma flexión. Por otra parte, si se identificase un error en uno de los paradigmas de flexión definidos sólo sería necesario corregirlo una vez.

De igual forma, ciertos mecanismos de derivación se pueden tratar de manera parecida, siempre que sean sistemáticas en ciertos lemas: por ejemplo, la formación de superlativos a partir de adjetivos en lenguas como el catalán o el español, la composición de ciertos lemas con determinados prefijos (como *ex-*, *vice-* o sufijos, etc.), y otros casos que pueden ser tratados de la misma manera que la flexión para que estos fenómenos se puedan beneficiar de las mismas ventajas que en aquel caso.

En este artículo denominaremos a la agrupación de transformaciones regulares entre partes de palabras —para gestionar los fenómenos que se han expuesto— como *definición de paradigmas*, sin reducirnos a tratar la exclusivamente la flexión.

Como formato para los diccionarios se define uno específico que utiliza XML, además de por interoperabilidad, tanto por las ventajas que presenta para explicitar relaciones entre elementos como porque permite expresar la codificación de caracteres de los datos de manera explícita, como por la abundancia y la potencia de las herramientas que existen para procesar y transformar datos incluidos en documentos XML.

Por último veremos cómo es posible explotar la división de entradas del diccionario en lema y paradigma para construir eficazmente transductores de letras mínimos. Estos transductores de letras mínimos estarán diseñados para su uso por procesadores eficientes del lenguaje. En (Garrido et al., 1999) se presentó un compilador de estas características pero que no aprovechaba completamente la factorización que permiten los paradigmas para acelerar la construcción. En (Daciuk et al., 2000; Carrasco and Forcada, 2002; Garrido-Alenda et al., 2002) se presentan métodos de construcción incremental de transductores de letras mínimos como alternativa al modelo que se presenta en este artículo.

2. Formato XML para los diccionarios

Se ha diseñado un formato basado en XML para almacenar la información de los diccionarios. La DTD (*document type definition*, una de las formas de especificar un formato XML) de este formato incluye secciones para especificar los caracteres que se consideran alfabéticos —en el sentido de que pueden formar parte de una palabra—, para definir símbolos que tengan sentido morfológico, definición de paradigmas y de identificación de expresiones regulares tales como números o direcciones de Internet. En el momento de enviar este artículo no se incluye ninguna referencia a esta DTD porque se encuentra todavía en desarrollo.

La figura 1 muestra un ejemplo de definición de un paradigma y su uso en el diccionario. Los paradigmas tienen *entradas* (elemento `<e>`), y para este caso, cada entrada consiste en una pareja (`<p>`), con parte izquierda (`<l>`) y parte derecha (`<r>`). Dentro de estos elementos se puede incluir texto o símbolos morfológicos `<s>`. Las entradas del diccionario se definen de la misma forma, la etiqueta identidad `<i>` es una forma abreviada de especificar una pareja con la parte izquierda y la parte derecha idénticas. El paradigma de la palabra se expresa, para el caso de la figura, al final mediante una referencia a paradigma `<par>`.

Se pueden definir paradigmas cíclicos con sólo indicarlo mediante un atributo del paradigma. Cabe notar que no todos los paradigmas se pueden definir como cíclicos, sino sólo aquellos que no aceptan la cadena vacía, ya que se puede dar el caso de que la salida sea infinita para una entrada dada (bucle sin consumo de entrada). Detectar que un paradigma ha sido definido como cíclico incorrectamente es trabajo del compilador que construye el transductor de letras.

3. Obtención de paradigmas

Las formas léxicas que se corresponden con las formas superficiales de las entradas de estos diccionarios están compuestas de *lema* y una lista ordenada de *etiquetas morfológicas*. La primera de las etiquetas que se especifiquen se considera como la *etiqueta de*

```
<pardef n="-es n m">
  <e>
    <p>
      <l/>
      <r><s n="n"/><s n="m"/><s n="sg"/></r>
    </p>
  </e>
  <e>
    <p>
      <l>es</l>
      <r><s n="n"/><s n="m"/><s n="pl"/></r>
    </p>
  </e>
</pardef>
...
<e><i>pan</i><par n="-es n m"/></e>
```

Figura 1: Ejemplo de definición de paradigma y entrada en el diccionario para palabras que flexionan como *pan*.

categoría léxica, mientras el resto son llamadas *etiquetas de subcategoría léxica*.

Los paradigmas que se usan para construir los diccionarios que sean como los que se presentan en este artículo se pueden obtener mediante dos procedimientos:

- *Manualmente*. Un lingüista decide cómo se forman los paradigmas para unificar todas las formas superficiales y sus correspondientes formas léxicas. Esto puede ser necesario por conveniencia del lingüista.
- *Automáticamente*. Un programa de ordenador puede calcular paradigmas-sufijo unificando todas las entradas que tengan el mismo lema y la misma etiqueta de categoría léxica en una sola definición de paradigma. De forma análoga se puede realizar esto con paradigmas-prefijo o siguiendo cualquier otro criterio.
- *Automática y manualmente*. En ocasiones puede resultar necesario combinar las dos técnicas anteriores para conseguir los resultados que se busquen.

4. Construcción de transductores de letras

4.1. Definiciones preliminares

En este artículo, denotaremos con ϵ la *cadena vacía*, y mediante θ el *símbolo vacío*.

Definimos dos alfabetos Σ, o *alfabeto de entrada*, y Γ o *alfabeto de salida*.

Llamamos *transducción de cadenas* al par $(s : t)$ tal que $s \in \Sigma^\star$ es la *cadena de entrada* y $t \in \Gamma^\star$ la *cadena de salida*. Por su relación con la cadena vacía, podemos distinguir la transducción $(\epsilon : \epsilon)$ o *transducción nula*, las transducciones de la forma $(\epsilon : s)$ o *inserciones* y las transducciones de la forma $(s : \epsilon)$ o *borrados*. La transducción nula es un caso particular de inserción o de borrado. Las transducciones se pueden concatenar, $(s : t) \cdot (x : y) = (sx : ty)$.

4.2. Paradigmas

Llamamos *paradigma* (y lo denotamos con P) a un conjunto de transducciones sin ninguna restricción de tipo sobre su contenido. Diremos que el paradigma es *ciclable* si no incluye ninguna inserción.

Los paradigmas se pueden concatenar con transducciones o entre sí de la siguiente forma:

$$(s : t) \cdot P = \{(sx : ty) : (x : y) \in P\} \quad (1)$$

$$P \cdot (s : t) = \{(xs : yt) : (x : y) \in P\} \quad (2)$$

$$P_i \cdot P_j = \{(sx : ty) : (s : t) \in P_i \wedge (x : y) \in P_j\} \quad (3)$$

Los paradigmas considerados como transductores de letras tienen un estado inicial i_P y un conjunto de estados finales F_P. Cuando se hable del *estado final de un paradigma* se referirá a un único estado f_P que se puede crear en cualquier momento tal que todos los

$q \in F_P$ estén unidos a él mediante transiciones nulas ($\theta : \theta$).

Definimos *entrada* (de un paradigma) como la concatenación de paradigmas y transducciones en cualquier proporción y orden que define un subconjunto de transducciones de un paradigma dado.

4.3. Diccionarios

Un diccionario de transducciones se define (por ejemplo, para incorporar conocimiento lingüístico) usando paradigmas que describan la flexión. Un diccionario de transducciones se presenta como un conjunto $D = (E, \mathcal{P}, \mathcal{P}^c)$ en el que:

- E es el conjunto de *entradas* del diccionario de transducciones. El conjunto de entradas de un diccionario puede ser visto como un gran paradigma, con la restricción de que no puede contener ninguna inserción. En el diccionario, los paradigmas sirven para representar el conocimiento lingüístico existente en las realidades observadas por el procesador léxico.
- $\mathcal{P}$ es un conjunto de las definiciones del contenido del los paradigmas que se usan en las entradas del diccionario o en otros paradigmas.
- $\mathcal{P}^c$ es un subconjunto de $\mathcal{P}$, el de *paradigmas cíclicos*. Se impone la restricción de que sólo se puedan definir paradigmas cíclicos a partir de paradigmas que sean que sean ciclables.

Un diccionario D se puede representar mediante un transductor de letras que tiene como alfabeto de entrada $\Sigma \cup \{\theta\}$ y alfabeto de salida $\Gamma \cup \{\theta\}$; se define $L = (\Sigma \cup \{\theta\}) \times (\Gamma \cup \{\theta\})$. El conjunto de estados del transductor es Q, el estado inicial $q_I \in Q$, el conjunto de estados finales $F \subset Q$ y la función de transferencia $\delta : Q \times L \to 2^Q$, y por lo tanto es indeterminista tanto con respecto de la entrada como con respecto del nuevo alfabeto L

4.4. Construcción de transductores de letras mínimos

A partir de una transducción de cadenas se puede construir una *secuencia de transducciones de letras* $S(s : t)$ de longitud $N = \text{máx}(|s|, |t|)$ que se define como sigue para cada elemento $1 \leq i \leq N$:

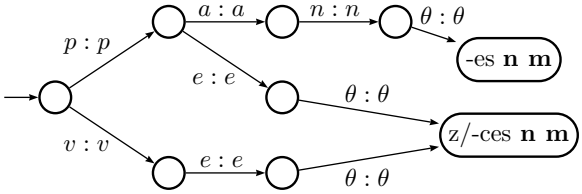

Figura 2: Construcción del diccionario como aceptor de prefijos y enlace con paradigmas mediante transiciones ($\theta : \theta$).

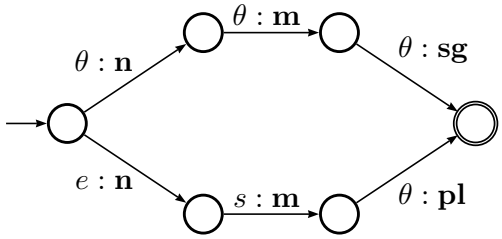

Figura 3: Paradigma «-es **n m**» minimizado que se usa en la figura 2.

$$S_i(s:t) = \begin{cases} (s_i : \theta) & \text{si } i \leq |s| \wedge i > |t| \\ (\theta : t_i) & \text{si } i \leq |t| \wedge i > |s| \\ (s_i : t_i) & \text{en otro caso} \end{cases} \quad (4)$$

Hay que destacar que, por construcción, se asegura que no puede existir ningún $(s : t)$ que sea igual a $(\epsilon : \epsilon)$, lo que es crucial para la consistencia del método de construcción que se verá posteriormente.

El método de construcción usa dos procedimientos, el procedimiento de *montaje* que se infiere de la ecuación 4 y el de minimización, que se realiza por un algoritmo convencional de minimización (van de Snepscheut, 1993) de autómatas finitos que consiste en invertir, determinizar, volver a invertir y volver a determinizar, tomando como alfabeto del autómata que hay que minimizar el producto cartesiano L y como transición vacía la $(\theta : \theta)$.

En la figura 2 se observa un ejemplo simplificado de este montaje. Se introduce transducción por transducción compuesta como en la ecuación 4 en un transductor en forma de *aceptor de prefijos* o *trie*, es decir, de manera en la que haya sólo un nodo para cada prefijo común del conjunto de transducciones que forman el diccionario. Con los sufijos de las transducciones (que no se comparten) se

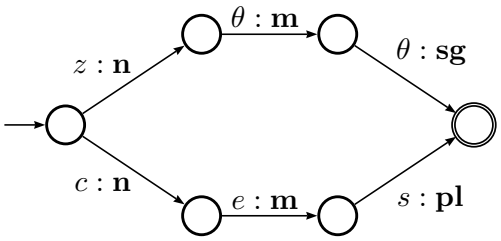

Figura 4: Paradigma «z/-ces **n m**» minimizado que se usa en la figura 2.

crean estados nuevos. En el punto en el que se haga referencia a un paradigma, se crea una réplica de ese paradigma y se enlaza a la entrada del diccionario que se está insertando en el transductor mediante una transducción nula ($\theta : \theta$).

Cada uno de los paradigmas, en tanto que pueden ser vistos como pequeños diccionarios, se han construido por este mismo procedimiento a su vez y se han minimizado para reducir el tamaño del problema en la construcción del diccionario grande. En las figuras 3 y 4 se muestra el estado de los dos paradigmas utilizados en la figura 2 después de la minimización.

Con los paradigmas cíclicos (ver figuras 5 y 6) se opera de una forma similar. En estos ejemplos se muestra como se construye un paradigma en el diccionario y con un transductor de letras. El paradigma construido puede ser usado en entradas del diccionario con ciertos nombres en lengua vasca que acaban en consonante.

4.5. Construcción avanzada de transductores de letras mediante paradigmas

Es un hecho que, en las lenguas de Europa, las modificaciones de las palabras tienen lugar al final o al principio de las mismas. Este hecho se puede explotar para mejorar la velocidad de construcción del transductor mínimo.

Como primera mejora, los paradigmas se pueden minimizar cuando se definen lo que permite manejar paradigmas más pequeños en el proceso de construcción. Como los paradigmas de los diccionarios de las lenguas con las que hemos trabajado suelen tener unos pocos cientos de estados, la minimización de estos paradigmas es un proceso muy rápido.

Si suponemos que una entrada puede presentar una referencia a un paradigma en cualquier punto de la misma, podríamos pensar en copiar en ese punto el transductor que se calcula en la definición del paradigma. El procedimiento que se presenta aquí se basa en que no siempre es necesario copiar, sino que en determinadas ocasiones es posible reutilizar un paradigma que ya haya sido copiado. En particular, dos o más entradas que comparten un paradigma como sufijo pueden reutilizar la misma copia de ese paradigma y lo mismo sucede cuando ocurre como prefijo. Sin embargo, en general, no es posible reutilizar paradigmas si se encuentran en posiciones intermedias de entradas diferentes, ya que se pueden introducir nuevos sufijos (prefijos) a entradas existentes, lo que hace que la información que se introduce en el transductor no es consistente con el diccionario, y el transductor generado sería incorrecto (realizaría transducciones que no se encuentran en el lenguaje que definen los diccionarios).

Definimos como $P_i^{[n]}$ como la *copia enésima del paradigma* i durante la construcción de D. De igual forma, se definen la *copia enésima del estado inicial del paradigma*, $i_{P_i}^{[n]}$, y la *copia enésima del estado final del paradigma*, $f_{P_i}^{[n]}$.

Podemos decir que una entrada del diccionario que comience por P_i *puede reutilizar* $P_i^{[n]}$ *como prefijo* si su estado inicial $i_{P_i}^{[n]}$ tiene una única transición de entrada. Además, esta transducción de entrada debe enlazarlo necesariamente a q_I y debe ser nula ($\theta : \theta$). La reutilización de esta copia de este paradigma se lleva a cabo mediante el enlace de $f_{P_i}^{[n]}$ con el sufijo restante de la entrada mediante una transición nula.

Análogamente, una entrada del diccionario que termine por P_i *puede reutilizar* cierta copia $P_i^{[n]}$ *como sufijo* si existe una única transición de salida del estado final de esta copia del paradigma, $f_{P_i}^{[n]}$, que además sea nula y que enlace esta copia con un estado final del D, es decir, que $q \in F$. La reutilización de este paradigma consistirá mediante el enlace del prefijo restante de la entrada con $i_{P_i}^{[n]}$ mediante una transición nula.

5. Resultados

Para comprobar el rendimiento del sistema se han utilizado tres diccionarios disponibles en catalán, español y portugués, todos ellos con un número similar de entradas (de 30.000 a 35.000 lemas, que se corresponden, según el caso, con entre un millon y medio

```
<pardef n="(en|aren)+" cyclic="true">
  <e>
    <p>
      <l>en</l>
      <r><s n="det"/><s n="pl"/>+<s n="gen"/></r>
    </p>
  </e>
  <e>
    <p>
      <l>aren</l>
      <r><s n="det"/><s n="sg"/>+<s n="gen"/></r>
    </p>
  </e>
</pardef>

<pardef n="(en|aren)*a">
  <e>
    <p>
      <l>a</l>
      <r><s n="det"/><s n="sg"/>+<s n="abs"/></r>
    </p>
  </e>
  <e>
    <par n="(en|aren)+">
    <p>
      <l>a</l>
      <r><s n="det"/><s n="sg"/>+<s n="abs"/></r>
    </p>
  </e>
</pardef>
  <!-- ... -->
<e><i>mutil</i><par n="(en|aren)*a"/></e>
```

Figura 5: Ejemplo simplificado de definición de paradigma y entrada en el diccionario para la palabra vasca *mutil* (chico).

y tres millones y medio de formas superficiales). Se realizaron pruebas para ver cómo la minimización previa de los paradigmas y, adicionalmente a esta mejora, la reutilización de paradigmas, afectan a la velocidad de construcción de los transductores de letras.

En la tabla de la figura 7 podemos ver el número de estados que se genera en la construcción del transductor antes de minimizar y su comparación con el mínimo, y cómo influye el aplicar las dos mejoras consecutivamente para lograr un transductor no mínimo del orden de sólo unas cuatro veces más grande que el mínimo en lugar de cincuenta o cien veces. En la práctica, el rendimiento de este método de construcción reduce el tiempo de construcción del transductor mínimo de varias horas a unos pocos segundos. Además el tamaño de la memoria que necesita durante la construcción varía según el caso, pero es entre 10 y 20 veces menor que el que se necesita si no se reutilizan las copias de los paradigmas de esta manera.

Mediante el procedimiento expuesto se están desarrollando los datos y los procesadores del lenguaje del proyecto "Traducción automática de código abierto para las lenguas del Estado español" y se obtienen por el momento unas prestaciones de analizadores o generadores morfológicos que funcionan en un entorno de velocidad de 40.000 palabras por segundo en un PC de escritorio, y el tiempo de construcción de los transductores de letras es de unos quince segundos para diccionarios de hasta unas 35.000 entradas que con los paradigmas desarrollados se corresponden con

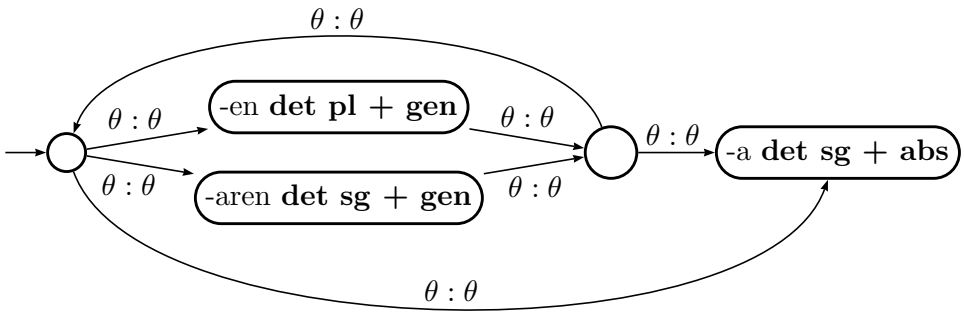

Figura 6: Ejemplo simplificado de construcción de un paradigma nominal cíclico para el vasco. Se ha desarrollado la representación que se infiere de la figura 5 para que pueda observarse el ciclo.

Lengua	Sin mejoras	Paradigmas preminimizados	**Reutilizando pref. y suf.**	Mínimo
Catalán	9.035.429	1.480.977	**204.504**	62.819
Español	3.641.407	1.531.536	**222.481**	59.815
Portugués	8.309.685	696.581	**154.951**	38.752

Figura 7: Número de estados de los autómatas construidos antes de la minimización, para los casos sin ninguna mejora, minimizando previamente los paradigmas, reutilizando paradigmas, y su comparación con el mínimo que es común para los tres procedimientos. El modelo que se presenta en este artículo se destaca en negrita.

unos cuatro millones de palabras diferentes. En cuanto al consumo de memoria, los diccionarios ocupan por el momento hasta cuatro megabytes en documentos XML. Si se expanden las transducciones de los diccionarios —la parte no cíclica—, estos ocupan hasta 400 megabytes en algún caso. Una vez compilado, el transductor resultante ocupa unos 600 kilobytes en disco, y en ejecución menos de 10 megabytes.

6. Conclusiones

Hemos visto cómo la gestión de diccionarios mediante paradigmas es una técnica que constituye una forma coherente de gestionar diccionarios de lenguas de naturalezas diferentes y que permite generar rápidamente procesadores léxicos muy eficientes. En desarrollos futuros se estudiará la forma de mejorar todavía más el procedimiento de construcción de los transductores de letras y de aumentar la eficiencia de los procesadores léxicos que se generan.

Bibliografía

Carrasco, R. C. and Forcada, M. L. (2002). Incremental construction and maintenance of minimal finite-state automata. artículo en Computational Linguistics.

Daciuk, J., Mihov, S., Watson, B. W., and Watson, R. E. (2000). Incremental construction of minimal acyclic finite-state automata. *Computational Linguistics*, 26(1):3–16.

Garrido, A., Iturraspe, A., Montserrat, S., Pastor, H., and Forcada, M. L. (1999). A compiler for morphological analysers and generators based on finite-state transducers. *Procesamiento del Lenguaje Natural*, (25):93–98.

Garrido-Alenda, A., Forcada, M. L., and Carrasco, R. C. (2002). Incremental construction and maintenance of morphological analysers based on augmented letter transducers. In *Proceedings of TMI 2002 (Theoretical and Methodological Issues in Machine Translation, Keihanna/Kyoto, Japan, March 2002)*, pages 53–62.

van de Snepscheut, J. L. A. (1993). *What computing is all about*. Springer-Verlag, New York.