



LÓGICA PARA CIENCIAS DE LA COMPUTACIÓN

Proyecto N° 1

Integración Java-Prolog: Predicción de texto para SMS

Primer Cuatrimestre de 2007

Introducción

Los teléfonos celulares se han vuelto un medio de comunicación muy popular, y con ellos los mensajes de texto o SMS/s. Debido a que no resulta práctico incluir en el celular un teclado alfanumérico (como el de una computadora), se han inventado métodos para escribir texto empleando solo el teclado numérico.

Considere un celular típico, como el mostrado en la figura 1. Cada una de las teclas correspondientes a los números del 2 al 9 tiene asociadas 3 o 4 letras (consecutivas) del alfabeto. Por ejemplo, la tecla 2 tiene asociadas las letras A, B y C, la tecla 3, las letras D, E, F y la tecla 7, las letras P, Q, R y S. Además, las teclas * y 0 tienen asociados los signos de puntuación punto (‘.’) y coma (‘,’), respectivamente, y la tecla # representa el caracter de espacio (‘ ’).

Un método simple para escribir un mensaje de texto empleando el teclado numérico consiste en lo siguiente. Si se desea introducir una cierta letra L en el mensaje, identificamos la tecla numérica T ($T \in 2, 3, 4, 5, 6, 7, 8, 9$) donde aparece L . Si L figura en primer lugar en T , debemos presionar T una vez, si figura en segundo lugar, dos veces y así siguiendo. Luego, para escribir la palabra “hola”, por ejemplo, debemos presionar dos veces la tecla 4, para introducir la ‘h’, tres veces la tecla 6 y así siguiendo. Este método de escritura resulta algo tedioso ya que, en el peor de los casos, para introducir una palabra de n letras podríamos tener que presionar hasta $4 * n$ veces el teclado.

Con el propósito de agilizar la escritura de SMS/s aparecieron los denominados métodos predictivos de escritura. Dichos métodos emplean un diccionario de palabras para asistir al usuario en la escritura, permitiendo escribir una palabra presionando solo una tecla por cada letra de la misma, y a lo sumo uno o dos tecleos más para seleccionar la palabra deseada de unas pocas candidatas.

El propósito del proyecto es implementar un método de escritura predictivo, incluyendo una interfaz gráfica que simule la interfaz del celular, permitiendo testear el método de forma interactiva. Concretamente, la interfaz gráfica deberá mostrar un celular como el de la figura 1, y deberá permitir al usuario escribir un mensaje de texto como si se tratara de un celular real (solo que en lugar de presionar teclas, el usuario hará click sobre ellas).



Figura 1: Celular

Predicción de Palabras

Si se emplea un método predictivo, para escribir una palabra P el usuario presiona, por cada letra de P , la tecla en la que figura dicha letra. Por ejemplo, para escribir “auto” el usuario presiona las teclas **2**, **8**, **8** y **6**. Sin embargo, dada una secuencia de teclas, existen muchas secuencias de letras asociadas (ya que cada tecla de la secuencia puede interpretarse como 1 de 3 o 4 letras posibles). Por ejemplo, la secuencia **2**, **8**, **8**, **6** tiene $3^4 = 81$ secuencias de letras asociadas: “attm”, “attn”, “atto”, “atum”, “**atun**”, ..., “autm”, “autn”, “**auto**”, ..., “bttm”, ..., “cvvn” y “cvvo”.

En este punto es donde el método debe *predecir* cuál de todas las secuencias de letras es la que el usuario desea introducir. Para efectuar dicha predicción, el método se basa en la siguiente hipótesis: *el usuario está intentando escribir una palabra “válida”*. Aquí es donde entra en juego el diccionario, que se emplea para identificar cuál o cuáles de todas las secuencias de letras constituyen realmente palabras. Para el caso de la secuencia 2, 8, 8, 6 las únicas dos palabras asociadas son “auto” y “atun”. Luego, el método presenta una lista con dichas palabras para que el usuario seleccione la que él pretendía escribir.

Para agilizar aún más la escritura, algunas variantes del método también sugieren palabras completas cuyos prefijos concuerden con la secuencia de teclas ingresadas. Esto permite al usuario introducir una palabra ingresando solo las primeras letras de la misma (prefijo), y eligiendo luego de una lista relativamente pequeña. Por ejemplo, si el usuario ingresó la secuencia **2**, **8**, **8**, **6**, se listan, además de “atun” y “auto”, las palabras “atunes”, “autonomo”, “autorizar”, “autos”, etc. Si por ejemplo se ingresa la secuencia **8**, **4**, **6**, se listan las palabras “tio” y “vio” (las únicas dos palabras completas que se corresponden con la secuencia) y además se listan “timbre”, “tinta”, “vine”, “vino”, etc, cuyos prefijos concuerdan con la secuencia de teclas.

El Diccionario

El diccionario empleado por los métodos de escritura predictivos se encuentra en general codificado empleando una estructura de datos conocida como *trie*. Un trie codificando un diccionario de palabras D es un árbol verificando las siguientes propiedades:

1. su raíz está etiquetada con el caracter ‘@’,
2. sus nodos internos (a excepción de la raíz) están etiquetados con letras del abecedario,
3. sus hojas están etiquetadas con el caracter ‘\$’,
4. dado un nodo interno N , no existen hijos de N con la misma etiqueta, y por lo tanto N tendrá a lo sumo un hijo por cada letra del abecedario (y posiblemente un hijo más etiquetado con ‘\$’).
5. cada camino de la raíz a una hoja del trie corresponde a una palabra del diccionario D . Más formalmente, P es una palabra en D si y solo si existe un camino $N_0, N_1, \dots, N_k$ en el trie, donde N_0 es la raíz y N_k es una hoja, E_i es la etiqueta de N_i , $1 \leq i \leq k$ y tal que $P = E_1 E_2 \dots E_k$.

La propiedad 4 implica que el trie factoriza los prefijos comunes de las palabras codificadas. Esto hace del trie una representación eficiente, tanto en espacio como en el tiempo de ejecución de operaciones básicas como verificar la pertenencia de una palabra al diccionario. En la figura

2 se muestra un trie para un diccionario conformado solo por las palabras “ata”, “atajo”, “atar”, “cae”, “caer”, “cal”. Notar como la representación (trie) factoriza el prefijo “ata” para las palabras “ata”, “atajo” y “atar”.

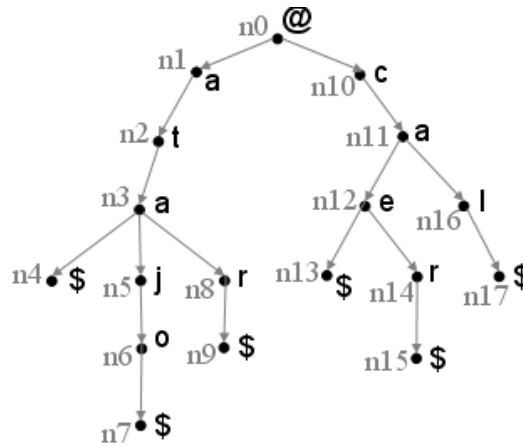


Figura 2: Diccionario Trie

Representación en Prolog

Una posible representación en PROLOG para un árbol trie consiste de:

- constantes $n0, n1, n2, \dots$ para representar nombres (o identificadores) de nodos
- hechos `trie(N, E, NP)` para representar que N es un nodo del trie etiquetado con E , y cuyo padre es el nodo NP , donde E es o bien una letra del alfabeto, la constante `@` (si N es la raíz) o la constante `$` (si N es hoja). Si N es la raíz del trie, NP es la constante `#`, representando que N no tiene padres.

Por ejemplo, el árbol de la figura 2 será representado mediante el siguiente código PROLOG:

```
trie(n0, @, #).  trie(n4, $, n3).  trie(n8, r, n3).  trie(n12, e, n11).  trie(n16, l, n11).
trie(n1, a, n0).  trie(n5, j, n3).  trie(n9, $, n8).  trie(n13, $, n12).  trie(n17, $, n16).
trie(n2, t, n1).  trie(n6, o, n5).  trie(n10, c, n0).  trie(n14, r, n12).
trie(n3, a, n2).  trie(n7, $, n6).  trie(n11, a, n10).  trie(n15, $, n14).
```

Interacción básica al escribir un SMS

La pantalla del celular se encuentra dividida en dos áreas, como muestra la figura 3. El área superior, o *área de mensaje*, muestra el mensaje de texto que se está escribiendo, mientras que el área inferior, o *área de sugerencias*, muestra la lista de palabras del diccionario sugeridas por el método para la secuencia de teclas que se está ingresando. El área de sugerencias tiene espacio para listar sólo 4 palabras simultáneamente. En caso de existir más de 4, el usuario podrá ir accediendo a las palabras restantes presionando la tecla con la flechita hacia abajo.

Al iniciar la interacción, el área del mensaje se encuentra vacía. Para comenzar a ingresar una palabra basta con presionar la tecla numérica correspondiente a su primer letra. Inmediatamente, en el área de sugerencias aparece una lista con las (primeras 4) palabras sugeridas por el método. A medida que el usuario presiona más teclas, el área de sugerencias va mostrando la lista de palabras correspondiente a la secuencia de teclas ingresadas hasta el momento. En

cualquier momento de este proceso el usuario puede dejar de presionar teclas y puede comenzar desplazarse por la lista de palabras buscando la que él deseaba escribir. Una vez posicionado sobre la palabra deseada, podrá seleccionarla para su inclusión en el mensaje presionando la tecla # (espacio), en cuyo caso se introduce la palabra seguida de un espacio, o bien presionando la tecla * o la tecla 0, lo que provoca la inclusión de la palabra seguida de un punto (‘.’) o una coma (‘,’), respectivamente. Luego de eso, el usuario puede comenzar a ingresar una nueva palabra presionando cualquier tecla de la 2 a la 9.

Si en un dado instante el celular muestra una cierta lista de sugerencias y el usuario presiona alguna de las teclas *, 0 o # sin haber seleccionado previamente una ellas, entonces se introduce la primer palabra de la lista. En otras palabras, la primera opción de la lista es la opción seleccionada por defecto.

Finalmente, como muestra la figura 3, los primeros n caracteres de la primer opción de la lista de sugerencias se encuentran subrayados, donde n es la longitud de la secuencia de teclas (de la 2 a la 9) ingresada hasta el momento. De esta forma, el usuario percibe el progreso mientras introduce una palabra al ver como crece la parte subrayada de la primera opción. Más aún, el usuario tendrá siempre noción de cuántas letras de la palabra ingresó, lo que ayudará a evitar que cometa errores como saltarse alguna letra o introducirla dos veces.



Figura 3: Interacción básica al escribir un SMS

Requerimientos de Diseño e Implementación

Para el presente proyecto deberá implementar:

- **en Prolog**, la predicción de palabras empleando un diccionario. Concretamente, deberá implementarse un predicado `t9_pal/2` que dada una secuencia de teclas Ts , representada como una lista de números del 2 al 9, retorne una palabra P del diccionario que corresponda con la secuencia Ts . Además deberá implementar un predicado `t9_pref/2` que dada una secuencia de teclas Ts retorne una palabra P tal que Ts corresponda con un prefijo (propio) de P . Estos dos predicados serán luego utilizados para armar la lista de palabras sugeridas por el método (ver sección *Predicción de Palabras*) para una dada secuencia de teclas Ts .

El diccionario de palabras que se empleará para efectuar la predicción será brindado por la cátedra, y estará codificado como se describió en la sección *El Diccionario*.

- **en Java**, una interfaz gráfica que simule la interfaz del celular, y permita al usuario escribir un mensaje de texto siguiendo la interacción descrita en la sección anterior. Concretamente, la interfaz gráfica deberá mostrar un celular como el de la figura 1, y deberá permitir al usuario escribir un mensaje de texto como si se tratara de un celular real (solo que en lugar de presionar teclas, el usuario hará click sobre ellas). Naturalmente, dicha interfaz gráfica consultará los predicados `t9_pal/2` y `t9_pref/2` para construir la lista de sugerencias.

Condiciones de Entrega

1. Aparte de los programas implementados, se deberá realizar un informe que explique brevemente los programas realizados, así como las decisiones de diseño tomadas, y toda otra observación que se considere pertinente. Incluir una sección que explique detalladamente los pasos requeridos para ejecutar la interfaz suministrada.
2. Las comisiones pueden estar conformadas por hasta 2 integrantes (recordar que éstos deben estar previamente registrados con la cátedra).
3. La fecha límite de entrega del presente proyecto es el día miércoles 18 de Junio de 2008 en el horario de clase. Los proyectos entregados fuera de término recibirán una penalización en su calificación, la cual será proporcional al retraso incurrido.
4. Deberá presentarse un folio plástico CERRADO (no entregar carpetas) conteniendo los siguientes elementos:
 - El informe impreso, el cuál deberá estar encabezado por una carátula que identifique claramente a los integrantes de la comisión.
 - Un disquete o cd conteniendo:
 - a) Una carpeta **Prolog**, que contenga un archivo `predictivo.pl` con el código PROLOG de los predicados implementados.
 - b) Una carpeta **Java**, que contenga el código JAVA de la interfaz implementada.
 - c) Una carpeta **Test**, que contenga todos los archivos requeridos para correr el sistema implementado, entre ellos el ejecutable (`.jar`) generado por JAVA.