Implementación de una herramienta didáctica para la búsqueda de la estrategia ganadora en un juego progresivo finito

Montse Massana, Joan Gimbert, Josep Maria Ribó
Universitat de Lleida
Escola Universitària Politècnica
mmassana@paeria.es
{josepma, joangim}@eup.udl.es

31 de Mayo de 1998

Resumen

En este trabajo se presenta la primera herramienta desarrollada en el marco de un trabajo de final de carrera a partir de una iniciativa conjunta de profesores de los departamentos de Matemáticas e Informática e Ingeniería Industrial de la Universidad de Lleida (UdL) para facilitar la asimilación de ciertos aspectos de la teoría de grafos y de la manipulación de estructuras de datos. En este caso se trata de una herramienta para comprender el proceso de obtención de una estrategia ganadora en un juego progresivo finito (JPF) a partir del cálculo del núcleo de un cierto digrafo asociado al juego. Esta herramienta, como el resto de las que pretendemos diseñar, se ha desarrollado como una applet escrita en Java y está disponible a través de Internet.

1 Introducción

Sin duda, la búsqueda de estrategias de innovación docente que resulten motivadoras para los estudiantes y que faciliten su aprendizaje (por supuesto, sin llegar a trivializarlo) supone un reto importante para aquellos profesores universitarios que han desarrollado una sensibilidad hacia la docencia. Algunas de las fuentes más claras en donde buscar caminos de innovación pedagógica son las nuevas tecnologías de la información que en este final de siglo nos acompañan infatigables dondequiera que vayamos.

En este sentido, un grupo de profesores de los departamentos de *Informática e Ingeniería Industrial* y también de *Matemática* de la Universidad de Lleida (UdL) estamos desarrollando una iniciativa conjunta para facilitar a los estudiantes de primeros cursos de las titulaciones de informática de esta universidad el aprendizaje de ciertos algoritmos de la teoría de grafos y de manipulación de estructuras de datos. La iniciativa consiste en proponer trabajos de final de carrera con un doble objetivo: Por un lado, crear herramientas didácticas en forma de aplicaciones informáticas para facilitar la comprensión de los algoritmos que forman parte de estas disciplinas. Por otro, hacer que éstas sean herramientas *universales* en el sentido que puedan ser utilizadas por cualquier persona interesada desde cualquier plataforma en la que trabaje, presumiblemente a través de Internet.

Si bien para cumplir con el primer requisito podríamos escoger cualquier lenguaje de programación para el desarrollo de tales herramientas, el segundo nos invita a pensar en un lenguaje con un alto grado de portabilidad, y que, si es posible, una aplicación escrita en este lenguaje pueda ser fácilmente insertada en una página *web* y utilizada por cualquier usuario que abra esa página. Java [4] es un lenguaje de programación que cumple esta condición.

Java permite escribir pequeñas aplicaciones (llamadas *applets*) que se integran dentro de una página *web*. Las *applets* se compilan para generar un código intermedio *único* para todos los sistemas. Este código intermedio puede ser interpretado por una *máquina virtual* que debe estar residente en cualquier navegador que un usuario quiera utilizar para visualizar la página en la que se ha integrado la *applet*. Además, Java dispone de una nutridísima biblioteca jerarquizada de clases que facilitan la programación y, en particular el diseño de la interficie con el usuario (elemento importante para nuestros propósitos).

Así pues, el objetivo final de nuestro trabajo es el de crear una biblioteca de herramientas didácticas desarrolladas en Java e integradas como *applets* en páginas *web* para facilitar la comprensión de los aspectos algorítmicos más significativos de la teoría de grafos (Dijkstra, Floyd, Kruskal, Prim, emparejamientos, juegos progresivos finitos, recorridos eulerianos y hamiltonianos...) y de la manipulación de estructuras de datos (manipulación de listas, tablas de dispersión, árboles binarios de búsqueda, árboles B...).

La primera de estas herramientas, la que tratamos en este trabajo, se ocupa de facilitar la comprensión de la teoría que permite hallar la estrategia ganadora en un *juego progresivo finito (JPF)* a partir del cálculo del núcleo de un cierto digrafo asociado al juego.

2 Juegos progresivos finitos

Posiblemente el lector haya sido invitado, en alguna ocasión, a participar en alguna clase de juego (sin apuestas) en el cual, después del resultado desfavorable de unas cuantas partidas, se empieza a tener la ligera sospecha o bien que el adversario es muy diestro o bien que hay alguna "trampa" en el juego conocida por el contrincante. Por trampa aquí entenderemos que hay una estrategia que seguida al pie de la letra permite, a quien la sabe, ganar siempre la partida por mucho que se esfuerze el oponente. Entre tales juegos se encuentran los juegos progresivos finitos, cuyo estudio, resolución y simulación constituye el objectivo de este trabajo. Tal estudio toma como referencias básicas los trabajos de [1], [2], [3] y [6].

Un *juego progresivo finito* es aquel que cumple las condiciones siguientes:

1. Intervienen dos únicos jugadores con turnos alternativos.
2. Desde cada posición o estado del juego, el número de posibles movimientos lícitos es finito;
3. El juego siempre termina, después de un número finito de movimentos, habiendo un único ganador.

El nombre de juego progresivo finito, responde al hecho de que en cada jugada del mismo se produce un avance hacia una posición final de la partida, progresión que siempre tiene lugar en un número finito de movimientos. Justamente esta característica permite justificar, tal y como veremos más adelante, que todo JPF tiene una única *estrategia ganadora*, es decir, una única forma de proceder que permite al primer jugador o al segundo, según corresponda, ganar la partida.

3 Algunos ejemplos de JPF

Una vez vistas las características de un JPF, podemos descartar de nuestro estudio aquellos juegos que no se ajustan a la definición dada, como son, por ejemplo, las damas, el ajedrez o el tres en raya, ya que el juego puede no terminar (tablas). Como ejemplos de JPF podemos citar los siguientes:

J 1 *(Juego de las dos pilas)* Se disponen de dos pilas con p y q fichas, respectivamente. Los movimentos lícitos son:

- Tomar cualquier número de fichas (al menos una) de una de las dos pilas.
- Quitar el mismo número de fichas de las dos pilas.

Gana aquel jugador que retira las últimas fichas.

J 2 *(Juego take-away simple)* Se inicia el juego con una fila de n fichas. En cada jugada se debe retirar un número de fichas comprendido entre 1 y una cantidad prefijada d. Gana el jugador que saca las últimas fichas.

J 3 *(Juego tipo Nim)* Empieza el juego con k pilas, cada una de ellas con un número n_i de fichas, $1 \leq i \leq k$. En cada jugada se puede quitar cualquier número de fichas, al menos una, pero de una misma pila. Gana aquel jugador que quita las últimas fichas.

4 Digrafo asociado a un JPF

La teoría de grafos proporciona el lenguaje y las herramientas adecuadas para representar un

JPF y hallar, al menos teóricamente, su estrategia ganadora. Así, dado un JPF J construiremos su digrafo asociado, que denotaremos por $D_J = (V, A)$, de la siguiente manera:

- Cada vértice del digrafo D_J representa una posición o estado del juego J;

- Cada arco de D_J corresponde a un movimiento del juego J, es decir, si x e y representan dos posiciones del juego, entonces tendremos el arco (x, y) si, y sólo si, existe alguna jugada válida que permita pasar de la posición x a la posición y.

Ejemplo 4.1 *En la figura 1 podemos ver el digrafo correspondiente a la modelización del juego take-away, en el caso de disponer de 9 fichas y permitir sacar, en cada turno del juego, un máximo de 3.*

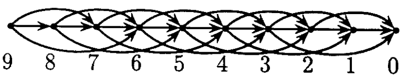

Figura 1: Modelización del *juego take-away*.

Observemos, en primer lugar, que el vértice 0 corresponde a la posición final (ganadora) mientras que el vértice 9 corresponde a la posición inicial. Fijémonos, además, que el vértice 4 correspondería a lo que podríamos llamar *posición pre-ganadora* ya que una vez alcanzada, independientemente de cual sea el próximo movimiento del oponente, ganaremos la partida. Y para llegar a tal posición nos interesaría situarnos previamente en la posición 8, es decir, empezar el juego quitando una sola ficha. Veremos, a continuación, que el conjunto de vértices $\{0, 4, 8\}$, correspondientes a las posiciones estratégicas del juego, constituye precisamente el núcleo del digrafo D_J.

5 Concepto de núcleo

Dado un digrafo $D = (V, A)$, un *núcleo* K de D es un subconjunto de vértices de D que cumple las dos condiciones seguientes:

1. K es *estable*, es decir,
$$\forall x \in K, \quad \Gamma(x) \cap K = \emptyset,$$
2. K es *absorvente*, es decir,
$$\forall x \notin K, \quad \Gamma(x) \cap K \neq \emptyset.$$

donde $\Gamma(x)$ denota el conjunto de vértices sucesores de x, es decir, adyacentes desde x.

Observemos que hallar el núcleo del digrafo asociado a un JPF equivale a encontrar su estrategia ganadora ya que ésta consistiría en moverse siempre a vértices del núcleo hasta que se alcanzase una posición final, la cual siempre pertenecerá al mismo. La traducción del siguiente resultado, en términos de juegos, permite concluir que todo JPF tiene una única estrategia ganadora.

Teorema 5.1 *Todo digrafo acíclico tiene un núcleo, y éste es único.*

Notemos que la pertenencia o no al núcleo del digrafo D_J de la posición inicial de un JPF J (que se corresponde con el vértice con grado de entrada 0), determina cual de los dos jugadores dispone de la estrategia ganadora.

6 Obtención de la estrategia ganadora de un JPF

La estrategia ganadora de un JPF J, o equivalentemente el cómputo del núcleo del digrafo acíclico asociado D_J, puede realizarse siguiendo diferentes métodos. Los algoritmos implementados en este trabajo han sido los siguientes:

1. Algoritmo recursivo basado en el cálculo del *nivel* de cada vértice x, entendido como la longitud mayor de los recorridos que van desde dicho vértice a un *vértice terminal*, es decir, con grado de salida cero.

2. Algoritmo de los vértices terminales consistente en ir reduciendo, iterativamente, el problema del cómputo del núcleo del digrafo inicial al cómputo de dicho núcleo en el subdigrafo resultante de suprimir los vértices terminales y sus antecesores (véase [5]).

3. Cálculo de la *función de Grundy*, definida como una función que asigna a cada vértice de un digrafo acíclico el menor entero no negativo no asignado a sus sucesores. Entonces, el núcleo del digrafo es precisamente el conjunto de vértices donde la función de Grundy toma el valor 0. Dicha función tiene ciertas propiedades que permiten reducir, en algunas situaciones, el cómputo de la misma en un digrafo complejo a su cómputo en digrafos más simples.

7 Herramienta didáctica

El objetivo último de este trabajo era el de disponer de una herramienta que resultase didáctica para el aprendizaje de la teoría que soporta la obtención de la estrategia ganadora en un JPF. Queríamos además, que esta herramienta pudiese ser utilizada fácilmente por cualquier persona interesada en este tema, no necesariamente de la UdL. Por ello hemos decidido implementarla en forma de *applets* escritas en lenguaje Java y disponibles a través de Internet. Dichas *applets* han sido programadas utilizando VJ++1.0 y JDK 1.1.6.

La programación de esta herramienta la hemos dividido en dos partes:

1. Una herramienta-editor que permite al usuario construir digrafos para modelizar JPF y, a continuación, aplicar los algoritmos presentados para la obtención del núcleo del digrafo generado. Estos algoritmos pueden aplicarse de dos maneras distintas:

   - *Paso a paso*, permitiendo al usuario avanzar a su ritmo y obtener información detallada en la *applet* acerca de la evolución de dicho algoritmo en cada instante.
   - *Ininterrumpida*, que permite un cálculo más rápido del nucleo del digrafo disponiendo solamente de una información global acerca del funcionamiento del algoritmo aplicado.

   Esta herramienta permite, además cargar los digrafos correspondientes a cuatro JPF, de los cuales se explica las reglas del juego y el sentido de su modelización mediante el digrafo que se presenta. La aplicación de los algoritmos a ese digrafo permite encontrar el núcleo del digrafo y, por tanto, la estrategia ganadora del JPF.

2. Tres juegos, programados en sendas *applets*, que permiten al usuario jugar contra el ordenador. Estas *applets* han sido diseñados teniendo en cuenta tanto la importancia de la interficie en el momento del juego como la presentación al usuario de información gráfica respecto la teoría que se esconde detrás de los JPF. Por tanto, el diseño de las *applets* consta de dos partes: una con la que el usuario interacciona para jugar contra el ordenador y otra en la que el usuario puede ver reflejados los cálculos de la estrategia ganadora en cada posición del juego. Además, los juegos programados son parametrizables.

Las *applets* a las que hemos hecho referencia se encuentran en:
http://www.udl.es/usuaris/e4089848/jpf/Nucli.html

Referencias

[1] J.M. Basart i P. Guitart, Grafs aplicats a la resolució de jocs. *Butlletí de la Societat Catalana de Matemàtiques*, n. 12 (1), 1997, pp. 17-25.

[2] C. Berge, *Graphs*. North-Holland, 1991.

[3] J. Gimbert, R. Moreno, J.M. Ribó y M. Valls, *Apropament a la teoria de grafs i als seus algorismes*, Ed. UdL, 1988.

[4] P. Naughton, H. Schildt, *Java.: Manual de referencia*, McGraw-Hill, 1997.

[5] S. Rudeanu, Notes sur l'existence et l'unicité du noyau d'un graphe. II Application des equations booleennes. *Revue française de recherche operationnelle*, n. 41, 1966, pp. 301-310.

[6] A. Tucker, *Applied combinatorics*, John Wiley, 2nd. edition, 1984.