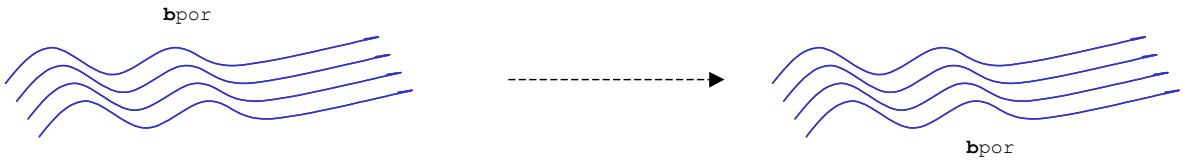
ISIS 211 Diseño de Algoritmos
Semestre 2003-1 - Parcial 3
Abril 29, 2003
Prof. Rodrigo Cardoso

Un barquero (b) está en la orilla de un río con un perro (p), una oveja (o) y un bulto de repollo (r). En el bote del barquero solo caben él y uno de los tres cargamentos. Por otro lado, en ningún momento debe dejar solos al perro con la oveja ni a la oveja con el repollo. La siguiente figura representa las situaciones inicial y final (el bote está siempre con el barquero):



Se quiere desarrollar un algoritmo de agenda para resolver el problema.

Represente un estado del proceso con una pareja de cadenas en L={b,p,o,r} cuyos elementos no tienen letras en común. Por ejemplo:

(bp,or), (b,por), (po,br), (bo,pr), ...

La idea es interpretar con la primera componente el conjunto de lo que hay en la primera orilla y, con la segunda componente, lo que hay en la segunda orilla. En los ejemplos, las primeras tres parejas representan estados indeseados, porque el perro esté solo con la oveja o ésta sola con el repollo.

1 **[40/100]** Exprese los diferentes elementos de una solución del problema con un algoritmo de agenda (SOLPOS, sat, …) utilizando la notación anterior.

2 **[10/100]** Determine un $k \in \mathbf{nat}$, tal que $|SUC(x)| \leq k$, para todo estado x y, además, $|SUC(x_0)| = k$, para algún estado x_0.

3 **[10/100]** Argumente si su algoritmo amerita o no manejo de nodos marcados.

4 **[10/100]** Muestre que, para todo estado x, no inicial, existe al menos un $y \in SUC(x)$ que puede no entrarse en AGENDA.

5 **[15/100]** Llame un predicado *dominó** a uno que cumple la primera de las condiciones de los predicados dominó, pero la segunda se cambia por la afirmación más fuerte:

[d2*] $\neg dom\ x \Rightarrow \neg sat\ y$, para $y \in SUC(x) \cup MARCADOS$.

Pruebe que puede definirse un predicado dominó*, que mejore el desempeño del algoritmo.

6 **[15/100]** Determine una cota superior para la complejidad exacta del algoritmo, con operación básica el número de viajes de la barca (mejor, si es más aproximada)

7 [20/100] Puede señalar una heurística admisible?

ISIS 211 Diseño de Algoritmos
Semestre 2003-1 - Parcial 3
Abril 29, 2003
Prof. Rodrigo Cardoso

***1** [40/100] Expresse los diferentes elementos de una solución del problema con un algoritmo de agenda (SOLPOS, sat, ...) utilizando la notación anterior.*

Sea:
$L_4 = \{\gamma |\ \gamma \subseteq L,\ \#S \leq 4\}$

Un $\gamma \in L_4$ se puede denotar con una palabra que contenga sus elementos. Por ejemplo, el conjunto $\{p,o,r\}$ se puede representar con la palabra por (o bien opr, o bien pro, etc.).

Para $\alpha,\beta \in L_4$, se usará la notación:

$\alpha \cap \beta$ ≡ "α y β no tienen letras en común"
$\alpha \cup \beta$ ≡ "α y β tienen todas las letras de L"
$v(\alpha)$ ≡ "en α no están solos p,o ni o,r"

SOLPOS = $\{(\alpha,\beta) : L_4 x L_4 \mid \alpha \cap \beta,\ \alpha \cup b\}$
[10/40]

sat(α,β) ≡ $\alpha = \langle\rangle$
[5/40]

SOL = $\{(\alpha,\beta) : SOLPOS \mid \alpha = \langle\rangle\}$
[2/40]

El espacio de búsqueda es el mismo SOLPOS:
BUSQ = SOLPOS.
[5/40]

La relación de sucesión entre nodos del espacio de búsqueda se puede definir así:

$(\mathbf{b}x_1 \ldots x_n, \gamma) \rightarrow (x_1 \ldots x_{i-1} x_{i+1} \ldots x_n, \mathbf{b}x_i\gamma)$, i=0,1,...,n
$(\gamma, \mathbf{b}x_1 \ldots x_n) \rightarrow (\mathbf{b}x_i\gamma, x_1 \ldots x_{i-1} x_{i+1} \ldots x_n)$, i=0,1,...,n

i.e., el barquero puede pasar al otro lado cualquiera de los objetos que lo acompañan (el caso i=0 corresponde al hecho de que el barquero viaje solo)
[15/40]

La búsqueda empieza en:
s = (bpor,⟨⟩).
[3/40]

Variantes:
Variaciones notacionales, como
No definir $\alpha \cap \beta$, $\alpha \cup \beta$

***2** [10/100] Determine un $k \in$ **nat**, tal que $|SUC(x)| \leq k$, para todo estado x y, además, $|SUC(x_0)| = k$, para algún estado x_0.*

Para un estado x, el número de sucesores depende del número de objetos en el lado del barquero. En la descripción genérica de los sucesores de x hay arcos

$(\mathbf{b}x_1 \ldots x_n, \gamma) \rightarrow (x_1 \ldots x_{i-1} x_{i+1} \ldots x_n, \mathbf{b}x_i\gamma)$, $i=0,1,\ldots,n$
$(\gamma, \mathbf{b}x_1 \ldots x_n) \rightarrow (\mathbf{b}x_i\gamma, x_1 \ldots x_{i-1} x_{i+1} \ldots x_n)$, $i=0,1,\ldots,n$

En cada caso: $|SUC\ x| = n+1 \leq 4$ (porque $n \leq 3$).

[5/40]

Además: $|SUC\ (bpor,\langle\rangle)| = |\{(por,b),(or,bp),(pr,bo),(po,br)\}| = 4$.

[5/40]

3 ***[10/100]*** *Argumente si su algoritmo amerita o no manejo de nodos marcados.*

Son necesarios: siempre hay posibilidad de repetir estados.

[10/10]

4 ***[10/100]*** *Muestre que, para todo estado x, no inicial, existe al menos un $y \in SUC(x)$ que puede no entrarse en AGENDA.*

Para un nodo no inicial siempre es posible que el barquero regrese deshaciendo la última acción que realizó. Esto se evidencia al definir los sucesores de x. El nodo del que se partió debió marcarse, de maner que no debe entrar a AGENDA.

[10/10]

5 ***[15/100]*** *Llame un predicado dominó* a uno que cumple la primera de las condiciones de los predicados dominó, pero la segunda se cambia por la afirmación más fuerte:*
$[d2^*]$ $\neg dom\ x \Rightarrow \neg sat\ y$, para $y \in SUC(x) \cup MARCADOS$.
Pruebe que puede definirse un predicado dominó, que mejore el desempeño del algoritmo.*

Se puede definir el siguiente predicado, para $(\alpha,\beta) \in BUSQ$:

$d(\alpha,\beta) \equiv v(\alpha) \wedge v(\beta)$

Se muestra que d satisface la primera propiedad de los predicados dominó:

Lema 1: $sat(\alpha,\beta) \Rightarrow d(\alpha,\beta)$
Dem:
Hip: $sat(\alpha,\beta)$
$d(\alpha,\beta)$
= $\langle$Hip: $sat(\alpha,\beta) \equiv \alpha=\langle\rangle \wedge \beta=bpor$ $\rangle$
$d(\langle\rangle, bpor)$
=
$v\langle\rangle \wedge v(bpor)$
=
true

[5/15]

Lema 2: $\neg d(\alpha,\beta) \Rightarrow \neg sat(\alpha,\beta)$
Dem:
$\neg d(\alpha,\beta)$ significa que la condición de no dejar solos a p,o ni a o,r es violada. Eso significa que la solución no pasa por este nodo.

[10/15]

6 ***[15/100]*** *Determine una cota superior para la complejidad exacta del algoritmo, con operación básica el número de viajes de la barca (mejor, si es más aproximada)*

Una cota superior es el número de nodos de BUSQ:

$$2*[bin(3,0)+bin(3,1)+bin(3,2)+bin(3,3)] = 2*2^3 = 16$$

`bin(3,k)` posibilidades para el primer conjunto de la pareja que define un estado, $k \in 0..3$. Además, el barquero puede estar en dos posiciones (a la derecha o a la izquierda).

[15/15]

Variantes:

Mejoras a esta cota, considerando menos ramificación, predicado dominó*, etc.
De todas maneras, 16 estados son fáciles de analizar!

7 [20/100] Puede señalar una heurística admisible?

...