ISIS 211 Diseño de Algoritmos
Semestre 2010-1 - Parcial 3 Solución
Mayo 6, 2010
Prof. Rodrigo Cardoso

Considérese un predicado p sobre **int**×**nat**, tal que p(n,k) sea verdadero si y solo si en la fórmula

n = ?1?2? … ?k

se puedan colocar signos '+' o '−' en los lugares de cada '?', de manera que la ecuación se satisfaga. Como ejemplos, se puede ver que p(12,7) vale, ya que 12=−1+2+3+4+5+6−7, pero p(12,3) es falso.

Dado un par (n,k)∈**int**×**nat** *se quiere calcular* p(n,k)*, con un algoritmo de agenda para búsqueda en grafos.*

1 **[20/100]** Justifique cada una de las afirmaciones:

[P1] p(n,0) ≡ n=0

La sumatoria a la derecha resulta una suma vacía cuando k=0.

[5/20]

[P2] p(n,k) ≡ p(n-k,k-1) ∨ p(n+k,k-1) , k>0

p(n,k)

=

n = ± 1 ± 2 ± … ± k

=

(n = ± 1 ± 2 ± … ± k-1 + k) ∨ (n = ± 1 ± 2 ± … ± k-1 - k)

=

(n-k = ± 1 ± 2 ± … ± k-1) ∨ (n+k = ± 1 ± 2 ± … ± k-1)

=

p(n-k,k-1) ∨ p(n+k,k-1)

[10/20]

[P3] p(n,k) ≡ p(-n,k)

p(n,k)

=

n = ± 1 ± 2 ± … ± k

=

-n = -± 1 -± 2 -± … -± k

=

p(-n,k)

[5/20]

2 **[20/100]** Utilice las siguientes definiciones:

SOLPOS = **nat**×**nat**
sat(x,y) ≡ x=y=0
SOL = {(0,0)}
BUSQ = SOLPOS.
s = (n,k)

2a Defina los arcos del grafo de búsqueda de manera que, para un nodo no terminal:

(x,y)→(u,v) ∧ p(u,v) ⇒ p(x,y)

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

$$(x,y) \to (|x-y|, y-1) \quad , \text{ si } y\geq 1$$
$$(x,y) \to (x+y, y-1) \quad , \text{ si } y\geq 1$$

Los arcos son justificados por [P2] y [P3]. De esta manera se cumple que, para (x,y) no terminal:

$$(x,y)\to(u,v) \ \wedge \ p(u,v) \ \Rightarrow \ p(x,y)$$

[10/20]

Dem: (Bono)

```
Hip: (x,y)→(u,v), p(u,v)
Casos: (u,v) = (|x-y|,y-1), (u,v) = (x+y,y-1)
Caso: (u,v) = (|x-y|,y-1)
          p(u,v)
     =          ⟨ Caso: (u,v) = (|x-y|,y-1) ⟩
          p(|x-y|,y-1)

          Casos: x≥y, x<y
          Caso: x≥y
               p(|x-y|,y-1)
          =          ⟨ Caso: x≥y ⟩
               p(x-y,y-1)
          ⇒          ⟨ P2 ⟩
               p(x,y)

          Caso: x<y
               p(|x-y|,y-1)
          =          ⟨ Caso: x<y ⟩
               p(y-x,y-1)
          =          ⟨ P3 ⟩
               p(x-y,y-1)
          ⇒          ⟨ P2 ⟩
               p(x,y)

Caso: (u,v) = (u,v) = (x+y,y-1)
          p(u,v)
     =          ⟨ Caso: (u,v) = (x+y,y-1) ⟩
          p(x+y,y-1)
     ⇒          ⟨ P2 ⟩
          p(x,y)
```

[+10]

2b Explique por qué la búsqueda de $(0,0)$ desde (n,k) equivale a decidir la verdad de $p(n,k)$.

Los arcos del grafo se construyen de modo que, si $(x,y)\to(u,v)$ y $p(u,v)$, entonces $p(x,y)$. Entonces, es fácil ver que:

$$(u,v)\in SUC^{*}(x,y) \ \wedge \ p(u,v) \ \Rightarrow \ p(x,y)$$

Es claro que $p(0,0)$ vale. Encontrar $(0,0)$ desde (n,k) significa que $(0,0)\in SUC^{*}(n,k)$, de manera que $p(n,k)$ debe valer también.

Si $p(n,k)$ vale, debe haber un camino desde (n,k) a $(x,0)$, usando repetidamente [P2], hasta asignar los signos de los k sumandos. Por [P1], se debe tener que $x=0$, es decir, se encuentra $(0,0)$.

[10/20]

3 **[20/100]**

3a Argumente si su algoritmo amerita o no manejo de nodos marcados.

No: no hay posibilidad formar ciclos, porque la segunda componente siempre baja en una unidad.

[5/20]

3b Considere el predicado $d(x,y) \equiv x \le y(y+1)/2$. Muestre que, si $\neg d(x,y)$, puede podarse la búsqueda en cuanto a sucesores de (x,y).

El máximo valor representable con sumas y restas de elementos de $1..y$ es $y(y+1)/2$. Si x supera este valor, no se puede construir, i.e., $\neg p(x,y)$. Por esto se puede definir:

$$d(x,y) \equiv x \le y(y+1)/2.$$

Supóngase $\neg d(x,y)$. Si para $(u,v) \in SUC^*(x,y)$ se tuviera $p(u,v)$, debería tenerse también que $(0,0) \in SUC^*(u,v)$ y, por tanto, $(0,0) \in SUC^*(x,y)$. Esto, según **2b**, implicaría que $p(x,y)$ vale, lo que es imposible, si $\neg d(x,y)$. Es decir, es posible desechar los sucesores de (x,y), porque entre ellos no está $(0,0)$.

[15/20]

4 **[10/100]** Estime, en términos de n, el orden de complejidad de la verificación del predicado de satisfacción.

$O(1)$: Chequear la igualdad $(x,y)=(0,0)$.

[10/10]

5 **[10/100]** Estime, en términos de n, el orden de complejidad del paso

$$AGENDA:=AGENDA \cup SUC(x)$$

$O(1)$: Hay, a lo sumo, dos sucesores para incluir en la agenda.

[10/10]

6 **[10/100]** Estime, en términos de n y k, el orden de complejidad de su algoritmo.

Se construye un árbol binario completo de raíz (n,k) y nodos de la forma $(u,0)$, i.e. $k+1$ niveles. La complejidad es $O(2^k)$. La poda con el predicado d es bastante eficiente, pero su análisis amerita un estudio más profundo.

[10/10]

7 **[20/100]** Para cada una de las siguientes afirmaciones, juzgue la veracidad de la misma, y explique su respuesta:

7a El problema de decidir $p(n,k)$ es un problema **NP**.

Sí es NP. Basta analizar como certificado que una asignación de signos valida la ecuación.

[5/20]

7b No se ha podido demostrar que existan problemas en **NP** que estén en **P**.

[5/20]

Falso: todo problema en **P** está en **NP**.

7c Defina `k-CLIQUE` como el problema de decidir si un grafo no dirigido tiene una *clique* de k vértices. Pruebe que `3-CLIQUE` está en **P**.

Basta verificar, en el peor caso, todos los conjuntos de tres nodos, buscando que alguno de esos conjuntos formen un triángulo. Si el grafo tiene n vértices, hay $\binom{n}{3} = n(n-1)(n-2)/6$ conjuntos de 3 elementos, el procedimiento descrito es $O(n^3)$.

[10/20]