

Puzzles con Diálogos

Blas C. Ruiz Francisco Gutiérrez José E. Gallardo
Dpto. de Lenguajes y Ciencias de la Computación, Universidad de Málaga
`{blas,pacog,pepeg}@lcc.uma.es`

Resumen

Muchos autores han planteado problemas con objeto de introducir la lógica y la aritmética elementales, y ejercitar estrategias de la matemática desde aspectos educativos y de investigación básica, llamados *Puzzles con Diálogos*; en éstos interviene un *árbitro* que elige ciertos datos, y reparte en forma secreta información parcial sobre éstos; el problema consiste en adivinar la información del árbitro a partir de un diálogo entre los participantes.

Estudiaremos en este trabajo variantes de un problema atribuido originalmente a John Conway. Esbozamos un modelo general que describe, utilizando el cálculo de predicados, una representación de la solución; ligeros ajustes permiten pasar a un programa de ordenador (en nuestro caso, escrito en *Haskell*) que además de resolver el problema particular, permite obtener conjeturas sobre las soluciones de problemas similares.

Palabras clave: juegos matemáticos, programación funcional.

EL PROBLEMA Y SUS VARIANTES

Variante 0 *John* elige dos números p y b entre 1 y 9, y escribe cada uno de ellos en la frente de *Paco* y de *Blas* respectivamente; obviamente, *Blas* ve el número de *Paco* y recíprocamente, pero no hay espejos en la habitación. Posteriormente, *John* elige un número y arbitrario entre 2 y 18, distinto de la suma $x = p + b$, y lo escribe en una pizarra, junto a la suma. *Manolo*, que está fuera de la habitación, escucha el siguiente diálogo:

Blas .- No sé el número de mi frente.

Paco .- Yo tampoco.

Blas .- Pues no me sirve de ayuda tu última afirmación.

Paco .- Ni la tuya.

Blas .- Casi lo sé, pero necesito más información.

Paco .- Y yo.

Blas .- Pues, sigo sin saberlo, y me estoy poniendo nerviosillo.

Paco .- Yo sigo sin saberlo y también estoy de los nervios.

Blas .- Ajá, ya sé el mío.

¿Tiene sentido el diálogo? ¿puede *Manolo* adivinar los números tomados por *John*, sabiendo que *Paco* y *Blas* son *inteligentes* pero no embusteros?

Realmente, quitando los adornos del diálogo, vemos que se repite 4 veces la secuencia de afirmaciones: *Blas* .- No lo sé, *Paco* .- No lo sé. Veremos que la última frase del diálogo no es trascendente para *Manolo*. Podemos modificar algo el problema:

Variante 1 Supongamos que *John* toma los números p y b del intervalo $\mathcal{I} = [1..t]$, el número y del intervalo $[2, 3, \dots, 2t]$, siendo distinto de la suma, y el diálogo se reduce a:

Blas .- No conozco el número de mi frente.

Paco .- Yo tampoco.

Blas .- Pues, sigo sin saberlo.

Paco .- Entonces, yo tampoco.

Blas .- Ya lo sé

¿Puede *Manolo* adivinar el valor de t ? ¿y los números?

Variante 2 Si eliminamos la última frase del diálogo y además *Manolo* conoce el valor de t ¿puede *Manolo* saber los números?

Variante 3 Para cada t , ¿cuál es el diálogo más largo del mismo tipo, y con sentido?

UNA PLANTILLA *HASKELL* PARA DESCRIBIR PUZZLES CON DIÁLOGOS

En esta sección presentaremos un modelo genérico para describir la solución a Puzzles con Diálogos del tipo anterior utilizando el lenguaje de programación *Haskell* (Peterson & Hammond, 1997; Ruiz, Gallardo, Gutiérrez, & Guerrero, 1995). En dichos problemas suelen intervenir dos o más *oradores* y un *oyente* que escucha el diálogo de éstos; para simplificar el planteamiento consideremos únicamente dos oradores. Supondremos también que el oyente *está aislado*: escucha la conversación y debe resolver el problema a partir de cierta información global que, por otro lado, es conocida por todos; por ejemplo, en la **Variante 0** del problema anterior ésta información sería el valor de t , donde los números se toman del intervalo $[1..t]$; esta información global será denotada con *glob*. Por otro lado, los *oradores* disponen de otra información pública sólo para ellos y para el árbitro y desconocida por cualquier otra persona que quiera resolver el problema; p.e., para el problema anterior, esta información sería la escrita en la pizarra; además, los números escritos en las frentes de los oradores conforman información privada para cada orador; la información pública (y la privada) será denotada con la variable *pub* (y *priv*). Estas variables tomarán valores sobre ciertas estructuras de datos, mientras que la información global corresponderá a una "función global":

```
data InfPrivada = ...
```

```
data InfPublica = ...
```

```
data InfGlobal = ...
```

```
glob :: InfGlobal
```

```
glob = 9 - - p.e., el valor de  $t$ 
```

Además, debe existir un algoritmo que a partir de la información disponible por cada orador genere el conjunto de valores posibles correspondiente a la información privada del otro. Si describimos los conjuntos con listas *Haskell*, el tipo asociado será:

```
compatible :: InfPrivada → InfPublica → [InfPrivada]
```

```
compatible priv pub = ...
```

No aparece la variable global *glob* como argumento ya que es conocida por todos; precisamente ésta es la interpretación del hecho de ser una "función global". El algoritmo *compatible* debe reflejar el comportamiento inteligente de los *oradores* y del *oyente*. El algoritmo puede ser utilizado por todos, y por supuesto por el *oyente*, que debe resolver el problema. A partir de estas consideraciones, podemos escribir una función booleana *dialogo* que captura la conversación; es decir, a cada frase del diálogo le corresponde una ecuación para la función y ésta recoge la información que dicha frase aporta:

```
dialogo :: NdeOrden → InfPrivada → InfPublica → Bool
```

Por ejemplo, a la primera frase de un diálogo del estilo a la **Variante 0** le corresponde la ecuación

$$dialogo\ 1\ priv\ pub = varios\ (compatible\ priv\ pub)$$

(la función *varios* comprueba si una lista tiene más de un elemento), mientras que a la segunda frase le corresponde la ecuación

$$dialogo\ 2\ priv\ pub = varios[\ priv' \mid \\ priv' \leftarrow compatible\ priv\ pub, \\ dialogo\ 1\ priv'\ pub]$$

($\leftarrow$ se lee "pertenece"). Para la **Variante 0** del problema esta ecuación significa que el conjunto de posibles números que *Paco* puede tener en su frente tales que viéndolo *Blas* no puede adivinar el suyo, tiene cardinal > 1 . Podemos considerar la siguiente estructura de datos, siendo t la función que proporciona la información global

$$\begin{aligned} data\ InfPrivada &= FrentePaco\ Int \mid FrenteBlas\ Int \\ data\ InfPublica &= Pizarra\ Int\ Int \\ t &= 9 \end{aligned}$$

El algoritmo *compatible* puede ser descrito en la forma

$$\begin{aligned} compatible\ (FrentePaco\ p)\ (Pizarra\ x\ y) &= map\ FrenteBlas\ (comp\ p\ x\ y) \\ compatible\ (FrenteBlas\ b)\ (Pizarra\ x\ y) &= map\ FrentePaco\ (comp\ b\ x\ y) \\ comp\ z\ x\ y &= [z' \mid let\ z' = x - z, z' > 0, z' \leq t] \\ &++ [z' \mid let\ z' = y - z, z' > 0, z' \leq t] \end{aligned}$$

(el operador $++$ permite concatenar dos listas, y corresponde a la unión de dos conjuntos; la función *map* toma como primer argumento una función y la aplica a cada elemento de la lista segundo argumento). Observamos que la lista *comp z x y* puede contener como máximo dos valores y éstos son los números que puede tener en la frente una persona, obtenidos a partir de los números que ve (x e y son los números de la pizarra y z el de la frente del compañero). A la vista de la descripción anterior, podemos simplificar las estructuras, ya que lo relevante es el índice de la afirmación del diálogo, la persona que realiza la afirmación y los números que ve; así, consideramos funciones para cada orador en la forma siguiente:

$$\begin{aligned} blas\ 1\ p\ x\ y &= varios\ (comp\ p\ x\ y) \\ blas\ (n+1)\ p\ x\ y &= varios\ [b \mid b \leftarrow comp\ p\ x\ y, paco\ n\ b\ x\ y] \\ paco\ n\ b\ x\ y &= varios\ [p \mid p \leftarrow comp\ b\ x\ y, blas\ n\ p\ x\ y] \\ espacio &= [(p, b, x, y) \mid p \leftarrow [1..t], b \leftarrow [1..t], \\ &\quad let\ x = p + b, y \leftarrow [2..2 * t], y/ = x] \end{aligned}$$

donde la función *espacio* computa el "espacio de búsqueda": el conjunto de posibles valores iniciales de los números. A partir de dos funciones auxiliares,

$$\begin{aligned} hastaBlas\ n &= [(p, b, x, y) \mid (p, b, x, y) \leftarrow espacio, blas\ n\ p\ x\ y] \\ hastaPaco\ n &= [(p, b, x, y) \mid (p, b, x, y) \leftarrow espacio, paco\ n\ b\ x\ y] \end{aligned}$$

podemos obtener las soluciones del problema. Obsérvese que una llamada a cualquiera de las funciones anteriores realiza una *búsqueda hacia atrás*.

UNA CONJETURA SOBRE LA SOLUCIÓN DEL PROBLEMA

En primer lugar, calculemos el tamaño del problema; el número de combinaciones posibles o tamaño del *espacio de búsqueda* (el número de elementos de la lista devuelta por la función *espacio*) será $2t^2(t-1)$ (p y b pueden tomar t valores distintos, mientras que y puede tomar $2t-2$ valores distintos: entre 2 y $2t$, pero distinto del valor de la suma). Por tanto, el tamaño del problema es del orden de $2t^3$, por lo que para valores grandes de t el proceso de búsqueda es bastante ineficiente (puede mejorarse añadiendo predicados intermedios dentro de las llamadas); sin embargo ¿cuántos elementos quedarán después de filtrar con la primera afirmación? es decir, ¿cual es la longitud de la lista *hastaBlas 1* ? Es fácil demostrar que tal valor es la mitad del inicial: $t^2(t-1)$; en efecto, si $1 \leq p \leq t$, $1 \leq q \leq t$, y $x = p + b$, el predicado *blas n p x y* solamente será cierto cuando se tengan las desigualdades $1 \leq y - p \leq t$, junto a $x \neq y$; es decir, y puede tomar solamente $t-1$ valores. La ejecución de distintas llamadas a los predicados *hastaBlas* y *hastaPaco* para sucesivos valores de n y distintos valores de t permite obtener la siguiente tabla, donde *les* denota el tamaño del espacio de búsqueda (la longitud de la lista *espacio*), y hB_n y hP_n indican el número de posibles combinaciones después de cada frase del diálogo (o sea, las longitudes de las listas *hastaBlas n* y *hastaPaco n*) (los valores que no aparecen son nulos):

t	<i>les</i>	hB_1	hP_1	hB_2	hP_2	hB_3	hP_3	hB_4	hP_4	hB_5	hP_5
3	36	18	4								
4	96	48	12	4							
5	200	100	30	12	4						
6	360	180	56	24	12	4					
7	588	294	98	46	24	12	4				
8	896	448	152	76	40	24	12	4			
9	1296	648	228	114	66	40	24	12	4		
10	1800	900	320	168	100	60	40	24	12	4	
...											

La tabla conduce a conjeturar que hay una frase del diálogo de la cual se deduce que el número de combinaciones posibles hasta ese momento es cuatro; a través del programa anterior podemos obtener tales combinaciones, sin más que dar valores distintos a t :

$\underline{t}$	$\underline{\text{hastaPaco } \frac{t-1}{2}}$			
3	(1, 2, 3, 4)	(2, 2, 4, 3)	(2, 2, 4, 5)	(3, 2, 5, 4)
5	(2, 3, 5, 6)	(3, 3, 6, 5)	(3, 3, 6, 7)	(4, 3, 7, 6)
7	(3, 4, 7, 8)	(4, 4, 8, 7)	(4, 4, 8, 9)	(5, 4, 9, 8)
9	(4, 5, 9, 10)	(5, 5, 10, 9)	(5, 5, 10, 11)	(6, 5, 11, 10)
11	(5, 6, 11, 12)	(6, 6, 12, 11)	(6, 6, 12, 13)	(7, 6, 13, 12)

$\underline{t}$	$\underline{\text{hastaBlas } \frac{t}{2}}$			
4	(2, 2, 4, 5)	(3, 2, 5, 6)	(2, 3, 5, 4)	(3, 3, 6, 5)
6	(3, 3, 6, 7)	(4, 3, 7, 8)	(3, 4, 7, 6)	(4, 4, 8, 7)
8	(4, 4, 8, 9)	(5, 4, 9, 10)	(4, 5, 9, 8)	(5, 5, 10, 9)
10	(5, 5, 10, 11)	(6, 5, 11, 12)	(5, 6, 11, 10)	(6, 6, 12, 11)
12	(6, 6, 12, 13)	(7, 6, 13, 14)	(6, 7, 13, 12)	(7, 7, 14, 13)

A la vista de la tabla podemos establecer la siguiente **conjetura**:

Para t impar $= 2n + 1$, después de la n -ésima afirmación de Paco , el único número posible para Blas es $b = n + 1$, y la única afirmación con sentido por parte de Blas si prosigue el diálogo es: Ajá, ya sé mi número; por el contrario, para $t = 2n$, después

de la n -ésima afirmación de Blas , el número de Blas no está determinado; además, si prosigue el diálogo con Paco , el espacio queda reducido al vacío, por lo que el problema no tendrá solución con $2n$ frases en el diálogo.

Demostración de la conjetura

Con objeto de probar la conjetura, *calcularemos* los predicados que capturan el diálogo; para ello, utilizaremos la notación habitual de la matemática en lugar de *Haskell*; así, denotaremos con $\mathcal{B}_n(p, x, y)$ y $\mathcal{P}_n(b, x, y)$ a los predicados correspondientes hasta la frase n -ésima de Blas o Paco ; siendo $\mathcal{I} = [1..t] \subseteq \mathbb{N}$, es fácil demostrar que se tiene (siendo $p, b \in \mathcal{I}$; $x, y \in [2..2t]$; $x \neq y$):

$$\begin{aligned} \mathcal{B}_1(p, x, y) &\equiv x - p, y - p \in \mathcal{I} \\ \mathcal{B}_{n+1}(p, x, y) &\equiv x - p, y - p \in \mathcal{I} \quad \wedge \quad \mathcal{P}_n(x - p, x, y) \quad \wedge \quad \mathcal{P}_n(y - p, x, y) \\ \mathcal{P}_n(b, x, y) &\equiv x - b, y - b \in \mathcal{I} \quad \wedge \quad \mathcal{B}_n(x - b, x, y) \quad \wedge \quad \mathcal{B}_n(y - b, x, y) \end{aligned}$$

y de aquí podemos obtener el siguiente teorema, que establece una demostración de la conjetura expuesta anteriormente (una demostración completa puede verse en (Ruiz, Gutiérrez, & Gallardo, 1999)).

Teorema Si $p, b \in \mathcal{I}$, $x, y \in [2..2t]$, $x \neq y$, y si además $x = p + q \vee y = p + q$, entonces:

- (1) Si $t = 2n + 1$, $\mathcal{P}_n(b, x, y) \Rightarrow b = n + 1 \wedge |x - y| = 1$
- (2) Si $t = 2n$, $\mathcal{B}_n(p, x, y) \Rightarrow |x - y| = 1$,
 $(p, b) \in \{(n, n + 1), (n + 1, n + 1), (n, n), (n + 1, n)\}$
y además , $\mathcal{P}_n(b, x, y) \equiv \text{Falso}$

Para el problema original ($t = 9$) a partir de la última afirmación de Paco , cualquier persona (inteligente) puede deducir que el número de Blas es el 5. Obsérvese que el corolario puede ser aplicado también para estudiar las distintas variantes del problema. Por ejemplo, para la **Variante 1**, tenemos que son ciertos los predicados $\mathcal{B}_2(p, x, y)$ y $\mathcal{P}_2(b, x, y)$, pero $\mathcal{B}_3(p, x, y)$ es falso, de donde deberá tenerse $t = 5$ (t no puede ser mayor que 5 ya que, por ejemplo, si $t = 6$, entonces los valores posibles son $(3, 3, 6, 7)$, $(4, 3, 7, 8)$, $(3, 4, 7, 6)$ y $(4, 4, 8, 7)$, por lo que, a la vista de la frente de Paco y de la pizarra es imposible que Blas sepa su número). Por tanto, para la **Variante 1**, Manolo sabe que $t = 5$, y que el número de Blas es el 3, pero el de Paco puede ser 2, 3 o 4; por el contrario, si Manolo viera la pizarra, tampoco podría saber el número de Paco .

Para la **Variante 2**, Manolo solamente puede conocer el número de Blas si $t = 5$. La **Variante 3** queda resuelta con el teorema anterior.

Referencias

- Peterson, J., & Hammond, K. (1997). *Report on the Language Haskell, A Non-Strict Purely Functional Language, Version 1.4* (Tech. Rep.). Yale University.
- Ruiz, B., Gallardo, J., Gutiérrez, F., & Guerrero, P. (1995). *Programación Funcional en Haskell*. Universidad de Málaga.
- Ruiz, B., Gutiérrez, F., & Gallardo, J. (1999). *Variaciones sobre un problema de John Conway* (Tech. Rep.). Universidad de Málaga, Dpto. de Lenguajes y Ciencias de la Computación.