

TERCER PARCIAL EXAMEN DE LÓGICA

Trimestre 12-P. Junio 26 de 2012.

Grupo: CCB02 **Profesor:** Dr. Carlos Barrón Romero

SOLUCION

Instrucciones. Conteste todas las preguntas.

El marco de sus respuestas y comprensión de los temas de Lógica son los objetivos de la UEA de Lógica (clave 111222) que transcribo a continuación:

1. Comprender los principios básicos de la lógica matemática.
2. Demostrar la validez de argumentos mediante reglas formales.
3. Aplicar principios de lógica matemática en la elaboración de programas de cómputo.

PREGUNTAS y RESPUESTAS

1) **(20)** Explicar los detalles (que regla o equivalencia u operación algebraica se usa) y justificar si es correcta o incorrecta toda la argumentación de la inferencia de la demostración siguiente.

Demostrar $p(5)$, dado $(\forall x, x \in \mathbb{N}, q(x) \Rightarrow (p(x) \vee r(x))), q(5), \neg\neg\neg r(5) \wedge 1$.

Demostración.

Deducción, de una proposición general se obtiene un caso particular,

$(\forall x, x \in \mathbb{N}, q(x) \Rightarrow (p(x) \vee r(x))) \Rightarrow 5 \in \mathbb{N}, q(5) \Rightarrow (p(5) \vee r(5))$.

A la proposición obtenida, se le aplica Modus Ponendo Ponens,

$q(5) \Rightarrow (p(5) \vee r(5))$

$q(5)$

 $p(5) \vee r(5)$

Algebra para simplificar $\wedge 1$,

$\neg\neg\neg r(5) \wedge 1 \equiv \neg\neg\neg r(5)$.

Algebra para cancelar o simplificar una doble negación,

$\neg\neg\neg r(5) \equiv \neg r(5)$.

Modus Tollendo Ponens y se obtiene el resultado,

$p(5) \vee r(5)$
 $\neg r(5)$

 $p(5)$. Todo la inferencia es correcta.

2) **(20)** Traduzca a predicados o enunciados y justifique su definición de equivalencia de la compilación de programas del siguiente predicado:

Todos los programas de cómputo de un lenguaje de alto nivel que se compilan, generan un programa compilado con un código de instrucciones y declaraciones de un lenguaje de máquina equivalente.

Sugerencia, ajuste el predicado: $\forall D, p$ y m son equivalentes $\Rightarrow p(D)=m(D)$.

Donde p es un programa en lenguaje de alto nivel, m es un programa en lenguaje de máquina, D es un conjunto de datos de entrada, $p(D)=\{\text{resultados que genera } p \text{ con } D\}$ y $m(D)=\{\text{resultados que genera } m \text{ con } D\}$.

RESPUESTA.

$\forall p, C(p)=m, \forall D, \text{son equivalentes} \Rightarrow p(D)=m(D)$.

Donde p es un programa en lenguaje de alto nivel, $C(p)$ es la creación por compilación de un código de instrucciones a partir del programa p , D es un conjunto de datos de entrada, $p(D)=\{\text{resultados que genera } p \text{ con } D\}$ y $m(D)=\{\text{resultados que genera } m \text{ con } D\}$.

3) **(20)** Justifique que predicados o enunciados (a, b o c) se obtienen lógicamente de:

Sea p , el programa de computo, que con el conjunto de datos D , funciona y en un tiempo finito crea el conjunto de resultados R .

a) $\forall E$ (conjunto de datos de entrada adecuados), $E \neq D \Rightarrow p$ funciona y en tiempo finito crea el conjunto de resultados correspondiente a E .

b) $p(D) \Rightarrow R$.

c) $p(D) = R$.

RESPUESTA.

Nunca el inciso a) porque afirma una generalización que no se puede obtener lógicamente de un solo enunciado sobre el programa, p , y el juego de datos, D .

Los incisos b) y c) son dos posibles traducciones de la proposición dada. b) Es una forma funcional de enunciados y c) es una proposición en forma de una ecuación de igualdad de conjuntos.

4) Dado el siguiente programa en lenguaje C y PF con el esquema de correctez, explique y justifique sus respuestas.

```
#include <iostream>
int main() {
    int i=0;
    float c=1.0;
    float a;
    printf("\nDar un valor positivo para a => ");
    scanf("%f",&a);
    while (1) {
        i=i+1;
        c=c+a;
        if (c>3.0)
            break;
    }
    printf("\nEl valor de c=\%f, iteraciones=%d\n",c,i);
    system("PAUSE");
    return EXIT_SUCCESS;
}
```

a) (10) De un ejemplo de un valor de a , donde termina en tres iteraciones y escriba los resultados de las variables c e i .

RESPUESTA.

```
{c=1.0; i=0; a=?; rutinas printf, scanf, etc.}
    printf("\nDar un valor positivo para a => ");
    scanf("%f",&a);
```

```
{c=1.0; i=0; a=0.67; rutinas printf, scanf, etc.}
    while (1) {
```

```
{c=1.0; i=0; a=0.67; rutinas printf, scanf, etc.}
        i=i+1;
```

```
{c=1.0; i=1; a=0.67; rutinas printf, scanf, etc.}
        c=c+a;
```

```
{c=1.67; i=1; a=0.67; rutinas printf, scanf, etc.}
        if (c>3.0)
```

```
{c=1.67; i=1; a=0.67; if es falso, repite el while; rutinas printf, scanf, etc.}
```

```

while (1) {
{c=1.67; i=1; a=0.67; rutinas printf, scanf, etc.}
    i=i+1;
{c=1.67; i=2; a=0.67; rutinas printf, scanf, etc.}
    c=c+a;
{c=2.34; i=2; a=0.67; rutinas printf, scanf, etc.}
    if (c>3.0)
{c=2.34; i=2; a=0.67; if es falso, repite el while; rutinas printf, scanf, etc.}
    while (1) {
{c=2.34; i=2; a=0.67; rutinas printf, scanf, etc.}
        i=i+1;
{c=2.34; i=3; a=0.67; rutinas printf, scanf, etc.}
        c=c+a;
{c=3.01; i=3; a=0.67; rutinas printf, scanf, etc.}
        if (c>3.0)
            break;
{c=3.01; i=3; a=0.67; if verdadero, sale del ciclo; rutinas printf, scanf, etc.}
        printf("\nEl valor de c=%f, iteraciones=%d\n",c,i);
{c=3.01; i=3; a=0.67; if verdadero, sale del ciclo; rutinas printf, scanf, etc.;
En la pantalla aparece
El valor de c=3.01, iteraciones=3
}
}

```

b) **(10)** De un par de valores distintos de a , donde no termina y explique porque no termina.
RESPUESTA.

Los valores menores a 1×10^{-8} no cambian el valor de $c=1.0$ por el número de dígitos de la mantisa de c en PF.

Dos valores pueden ser $a = 1 \times 10^{-9}$ o $a = 0$.

Como con estos valores $c=c+a$ es 1.0 (no cambia) el programa no genera la condición verdadera del $\text{if } c>3$ y nunca termina su cuenta, hay que abortarlo o interrumpirlo.

c) **(10)** De un ejemplo de un valor de a , donde termina y se obtenga un resultado exacto en la variable c .

Un valor puede ser $a = 3.0$

Como este valor

```
{c=1.0; a=3.0, i=1}
```

```
c=c+a
```

```
{c=4.0; a=3.0, i=1}
```

```
    if (c>3.0)
```

```
        break;
```

```
{c=4.0; i=1; a=3.0; if verdadero, sale del ciclo; rutinas printf, scanf, etc.}
```

```
    printf("\nEl valor de c=%f, iteraciones=%d\n",c,i);
```

```
{c=4.0; i=1; a=3.0; if verdadero, sale del ciclo; rutinas printf, scanf, etc.;
```

En la pantalla aparece

El valor de $c=4$, iteraciones=1

```
}
```

El resultado es exacto porque 4 esta dentro del rango de valores y del tamaño de la mantisa de 1.0 y 3.0.

d) (10) De un ejemplo de un valor de a , donde termina y se obtenga un resultado no exacto en la variable c .

Un valor puede ser $a = 1.0e11$

Como este valor

$\{c=1.0; a = 1.0 \times 10^{11}, i=1\}$

$c=c+a$

$\{c=1.0 \times 10^{11}; a = 1.0 \times 10^{11}, i=1\}$

if ($c>3.0$)

break;

$\{c=1.0 \times 10^{11}; a = 1.0 \times 10^{11}, i=1; \text{if verdadero, sale del ciclo; rutinas printf, scanf, etc.}\}$

printf("\nEl valor de $c=\%f$, iteraciones= $\%d$ \n", c,i);

$\{c=1.0 \times 10^{11}; a = 1.0 \times 10^{11}, i=1; \text{if verdadero, sale del ciclo; rutinas printf, scanf, etc.};$

En la pantalla aparece

El valor de $c=1.e11$, iteraciones=1

}

El resultado no es exacto porque 1.0 no está dentro del rango de valores y del tamaño de la mantisa de 1.0×10^{11} .