

Lógica para la Computación

Convocatoria de junio. Segundo parcial, 13/01/25

Nombre:

DNI:

PCEO ☐

Grupo A ☐

Grupo B ☐

NOTA: Es necesario acumular un mínimo de 3 ptos (el 50% de la puntuación total) en el conjunto de las pruebas escritas para sumar las notas prácticas. La duración del examen es de 1'5 horas, y se puntúa sobre 3'5 ptos.

1. (1 pto) Sea el predicado PROLOG $p/3$ definido por la cláusula siguiente:

$$p(X-RX, Y-RY, X-RY) :- RX=Y.$$

NOTA: $X=Y$ es cierto sii X e Y son unificables.

- (a) (0.25 ptos) Describir su semántica declarativa.

El predicado $p/3$ se verifica sii el primer y segundo punteros de su tercer argumento ($X-RY$), coinciden respectivamente con el primer y segundo puntero de sus dos argumentos iniciales ($X-RX$ e $Y-RY$) ... y, además, el segundo puntero de la estructura del primer argumento $X-RX$ es unificable con el primer puntero de la estructura del segundo argumento $Y-RY$.

Esto es, $p/3$ implementa la concatenación de las estructuras incompletas asociadas a sus dos primeros argumentos.

- (b) (0.25 ptos) Identificar el resultado de la pregunta y justificar la respuesta.

$$:- p([a,b|Cddr1]-Cddr1, [c,d|Cddr2]-Cddr2, L-[]).$$

La respuesta es la siguiente:

$$Cddr1 = [c, d] \quad Cddr2 = [] \quad L = [a, b, c, d]$$

Basta para ello con unificar la pregunta y la cabeza de la cláusula que define el predicado.

- (c) (0.25 ptos) Proponer una implementación alternativa del predicado en forma de axioma.

$$p(X-RX, RX-RY, X-RY).$$

- (d) (0.25 ptos) Explicar la razón por la que no es necesario definir el operador $-$ utilizado en el predicado.

El operador ya está predefinido como operador aritmético y, por tanto, es utilizable notacionalmente.

2. (1 pto) Implementar un predicado **assertb/1** que haga sensible **asserta/1** al retroceso. Esto es, si en el transcurso de una resolución se produce un retroceso sobre el objetivo **asserta(Clausula)** que había determinado la inclusión de **Clausula** en nuestra base de datos, el intérprete deberá ahora eliminarla. Justificar la respuesta, describiendo la semántica declarativa del nuevo predicado **assertb/1**.

Ver el Ejemplo 12.3.1 en el fichero

Documentos y Enlaces/Material de Estudio/Prolog/prolog.pdf

de la entrada Moovi de la asignatura, y que ha servido de guía a las clases.

3. (0.5 pts) Completar el programa de cláusulas PROLOG siguiente:

```

nat 0.
nat s N :- nat N.

```

que pretende caracterizar el conjunto $\mathbb{N}$, de forma que podamos plantear preguntas del tipo `:- nat X`. Justificar la respuesta.

Basta con definir de forma adecuada los dos operadores usados, por ejemplo, en la forma:

```

:- op(100,fy,s).
:- op(200,fx,nat).

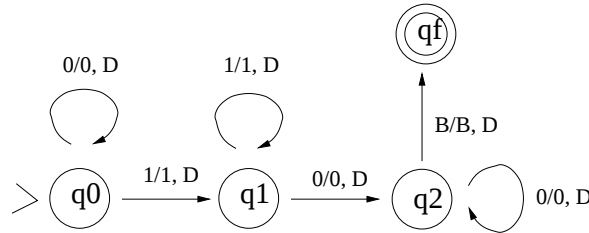
```

ambos como prefijos, siendo el operador `s` más prioritario que `nat`. Dado que se pretende denotar un natural `n` mediante la notación `s .. s 0`, necesitamos que `s` sea asociativo. No así el operador `nat`.

4. (1 pto) Se pretende reconocer el lenguaje $\mathcal{L} = \{0^n 1^{m+1} 0^{k+1}, \text{ tal que } n, m, k \geq 0\}$.

(a) (0.5 pts) Diseñar el grafo de estados de una Máquina de Turing que resuelva el problema.

Una posible solución es la dada por el grafo siguiente:



(b) (0.25 pts) Trazar los movimientos de la máquina para la entrada $w = 010$.

Para la máquina propuesta, la secuencia de movimientos sería la siguiente:

$$q_0 010 \vdash 0q_0 10 \vdash 01q_1 0 \vdash 010q_2 \vdash 010Bq_f$$

(c) (0.25 pts) Justificar el diseño de la máquina.

El estado q_0 se corresponde con el reconocimiento del primer grupo de 0's (0^n). El paso a q_1 garantiza que al menos debe haber un 1, mientras el ciclo sobre dicho estado reconoce al resto de la secuencia (1^m). De modo análogo, el paso a q_2 garantiza que al menos debe haber un 0 siguiendo a la secuencia de 1's, mientras el ciclo sobre dicho estado reconoce al resto de la secuencia (0^k).