

Evaluación

Duración de la prueba: 3 horas

Ejercicio 1 [evaluación individual del laboratorio]

Los siguientes puntos deben contestarse en base a la implementación del obligatorio realizado por su grupo:

- Describa brevemente (no más de cinco líneas) el procedimiento mediante el cual se realiza la resolución de los puzzles.
- Escriba la especificación en alto nivel del problema SAT para obtener cuadrados latinos, indicando el espacio de variables y las restricciones.

Ejercicio 2 [20 puntos]

- Sea P un programa lógico y G un objetivo. Indique si las siguientes afirmaciones son verdaderas o falsas. Justifique.
 - Todo programa lógico tiene al menos un modelo.
 - Un objetivo definido (cláusula con todos los literales negativos) es siempre insatisficible.
 - θ es una respuesta correcta de $P \cup \{G\}$ sii θ es una respuesta computada de $P \cup \{G\}$
- En las siguientes preguntas α y β refieren a fórmulas de la lógica proposicional.
 - Defina brevemente el problema SAT
 - ¿Es posible utilizar un algoritmo SAT para resolver el problema α implica lógicamente β ? Justifique brevemente.
 - ¿Es posible utilizar un algoritmo SAT para determinar si α es lógicamente válida? Justifique brevemente.

Ejercicio 3 [20 puntos]

- Implemente el siguiente metaintérprete para *Prolog* puro:
resMaximo(+G,+N) $\leftarrow$ resuelve G siempre que el árbol de la prueba no supere la profundidad N .
- Construya una gramática DCG que reconozca el lenguaje $L = \{y, w \in (a|b)^* / x = yww^Ry\}$.

Ejercicio 4 [20 puntos]

Sea el siguiente programa *Prolog*:

- $p([X|Xs], [X|Ys], Zs) \quad :- \quad q(Xs, Ys, Zs).$
- $p([X|Xs], Ys, [X|Zs]) \quad :- \quad q(Xs, Ys, Zs).$
- $q([X], [X], []).$
- $q([X, Y|Xs], [Y|Ys], [X|Zs]) \quad :- \quad q(Xs, Ys, Zs).$
- $q([X, Y|Xs], [X|Ys], [Y|Zs]) \quad :- \quad q(Xs, Ys, Zs).$

- Dé los valores de A y B que son solución de la consulta: $?- p([1,2,3,4], A, B).$
- Para las siguientes variantes con *cut* del programa anterior, indique los valores de A y B que son solución de la consulta de la parte (a). Justifique sus respuestas.
 - Se sustituye la línea 3 por:
 $q([X], [X], []) \quad :- \quad !.$
 - Se sustituye la línea 4 por:
 $q([X, Y|Xs], [Y|Ys], [X|Zs]) \quad :- \quad !, q(Xs, Ys, Zs).$

- c) ¿Cuál es la diferencia entre un *cut* rojo y un *cut* verde? Indique cuál es el color de los *cuts* de la parte (b).

Ejercicio 5 [25 puntos]

Mastermind es un juego en el cual se debe adivinar un código secreto.

El jugador A elige una secuencia de 4 dígitos decimales distintos como código secreto. El jugador B hace conjeturas y le consulta a A por el número de *buenas* (número de dígitos que aparecen en posiciones idénticas en la consulta y en el código) y *regulares* (número de dígitos que aparecen tanto en la consulta y en el código, pero en posiciones distintas).

Existe un algoritmo muy simple para jugar este juego: Imponer cierto orden en el conjunto de conjeturas posibles; luego iterar, haciendo que la próxima conjetura sea *consistente* con toda la información que se ha obtenido hasta el momento, hasta que se encuentre el código secreto.

El ciclo principal del programa está en el predicado `mastermind(+RespPrevias, -Codigo)`, cuya lógica se detalla en el siguiente pseudocódigo:

```
1: genero una nueva conjetura
2: chequeo la consistencia
3: consulto al oponente por las buenas y regulares de esa conjetura
4: chequeo si esa conjetura es solucion o no. En caso negativo, la guardo en el historial y vuelvo a 1.
```

En cada paso se genera una nueva conjetura y se la testea contra todas las respuestas obtenidas anteriormente para ver si la misma es *consistente*. Una vez que se obtiene una conjetura consistente, se le pregunta al jugador B cuáles son las buenas y regulares utilizando el siguiente predicado:

`consultarOponente(+Consulta, -Buenas, -Regulares) ← Consulta es una lista de 4 dígitos decimales distintos, Buenas y Regulares (ya definidos) resultan de comparar Consulta con el código secreto`

Puede asumir que este predicado ya está implementado.

A medida que avanza el juego, el jugador A irá construyendo una lista de *respuestas previas*, conteniendo todas las conjeturas fallidas junto con sus valores de buenas y regulares correspondientes. Esta información es vital a la hora de elegir la siguiente conjetura.

Decimos que una conjetura es *consistente* con un conjunto de respuestas previas, si al testear la conjetura contra cada respuesta, se obtienen el valor de buenas y regulares correspondientes.

Por ejemplo, considere que hasta el momento el jugador A generó las siguientes conjeturas y obtuvo las siguientes respuestas:

```
[2,3,1,6] -- 1B, 1R
[1,0,6,7] -- 1B, 0R
```

- Supongamos que elige como siguiente conjetura `[3,6,5,7]`. Si este fuera el código secreto, al testearlo contra `[2,3,1,6]` debió haber devuelto 0B, 2R. Debido a que la respuesta fue diferente, decimos que esta conjetura es *inconsistente*.

- Ahora supongamos que elige como siguiente conjetura `[1,3,5,8]`. Esta conjetura es *consistente* debido a que cumple con los valores de buenas y regulares para las dos respuestas conocidas.

Se pide implementar los siguientes predicados, de acuerdo a las especificaciones ya vistas:

```
generarConjetura(-Conjetura)
chequearConsistencia(+RespPrevias, +Conjetura)
mastermind(+RespPrevias, -Codigo)
```

Nota: Puede utilizar, sin implementarlos, todos los predicados extralógicos, metalógicos, aritméticos y de 2do. orden vistos en el curso. Cualquier otro predicado auxiliar utilizado en la resolución del problema debe ser implementado.