

Ejercicios de Laboratorio (5)

Programación Lógica con SWI-Prolog

Universidad de Santiago de Chile
Departamento de Ingeniería Informática
Paradigmas de Programación (2019-2)
Prof. Daniel Gacitúa Vásquez

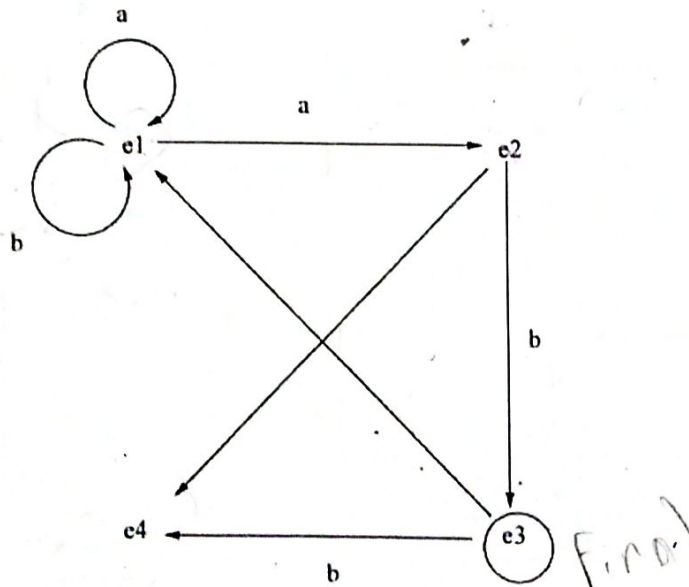
RUT: _____

Instrucciones:

- Tiene 70 minutos para responder esta guía de ejercicios
- La respuesta final debe estar en papel, pero puede usar su PC como apoyo metodológico
- Se utilizará evaluación de pares, considere los siguientes aspectos para la evaluación:
 - Punto base (intentó hacer el ejercicio) [0 o 1]
 - Descripción del diseño de la solución [0, 0.5 o 1]
 - Correcta documentación de la solución (comentarios) [0, 0.5 o 1]
 - Correcta formulación de los hechos [0, 0.5, 1, 1.5 o 2]
 - Correcta formulación de las reglas [0, 0.5, 1, 1.5 o 2]

Ejercicio 1:

El siguiente esquema representa un autómata no-determinista compuesto por los estados e1, e2, e3 y e4 además de las transiciones entre estos. En este autómata el estado e3 es estado final. Este autómata permite procesar cadenas de texto compuestas por los caracteres a y b. Por ejemplo, partiendo en el estado e1 las cadenas "abab", "aaaab", "abaaab" son reconocidas dado que la secuencia de transiciones termina en el estado e3. Por otro lado, si se parte en el estado e1, el string "aaabb" no es reconocido ya que termina en el estado e4. Cabe señalar que este autómata se considera no determinista porque para un estado existen más de una transición para un mismo símbolo (ej: estado e1 con dos transiciones para el símbolo a). En este autómata también existen transiciones nulas, esto es, que no hay símbolo de por medio.



1) Representar el autómata utilizando las siguientes relaciones:

- final(X) que se verifica si X es el estado final.
- trans(E1,X,E2) que se verifica si se puede pasar del estado E1 al estado E2 usando la letra X.
- nulo(E1,E2) que se verifica si se puede pasar del estado E1 al estado E2 mediante un movimiento nulo

2) Definir la relación acepta(E,L) que se verifique si el autómata, a partir del estado E, acepta la lista L de símbolos. Por ejemplo:

- acepta(e1,[a,a,a,b]) => true
- acepta(e2,[a,a,a,b]) => false

3) Mostrar a través de una traza cómo se evaluaría la consulta acepta(e1,[a,a,a,b])

Ejercicio 2:

La Coordinación de Paradigmas desea ahora tener un método en SWI-Prolog para calcular la condición de un alumno de Paradigmas de Programación en base a sus notas definitivas (tras aplicar correcciones y comodines). La Figura 1 muestra un ejemplo de los hechos base requeridos para calificar al alumno "Juan".

Se solicita crear los predicados necesarios para que, dado el nombre de un alumno, se pueda calcular su promedio de notas de cátedra y lab (según Reglamento 2019-2) y determinar si aprobó cátedra, lab o ambos.

Genere los hechos para un nuevo alumno y describa como sus reglas permiten obtener la condición de dicho alumno.

```

pep1(juan, 6.5).
pep2(juan, 3.3).
pep3(juan, 4.1).
pa(juan, 4.6).
pcat(juan, 5.5).
lab1(juan, 4.0).
lab2(juan, 5.5).
lab3(juan, 4.2).
lab4(juan, 4.8).
plab(juan, 4.0).

```

Figura 1

RUT Evaluador:	Ejercicio 1	Ejercicio 2
Punto base	1	
Descripción del diseño	1	
Documentación y comentarios		
Hechos	2	
Reglas		
TOTAL		

~~acepta(e3,[]): 3~~

Ejercicios de Laboratorio (6)
Programación Lógica con SWI-Prolog

Universidad de Santiago de Chile
Departamento de Ingeniería Informática
Paradigmas de Programación (2019-2)
Prof. Daniel Gacitúa Vásquez

RUT: _____

Instrucciones:

- Tiene 70 minutos para responder esta guía de ejercicios
- La respuesta final debe estar en papel, pero puede usar su PC como apoyo metodológico
- Se utilizará evaluación de pares, considere los siguientes aspectos para la evaluación:
 - Punto base (intentó hacer el ejercicio) [0 o 1]
 - Descripción del diseño de la solución [0, 0.5 o 1]
 - Correcta documentación de la solución (comentarios) [0, 0.5 o 1]
 - Correcta formulación de los hechos [0, 0.5, 1, 1.5 o 2]
 - Correcta formulación de las reglas [0, 0.5, 1, 1.5 o 2]

Ejercicio 1:

La Universidad de Santiago de Chile en su departamento de registro curricular tiene que realizar una asignación de cursos a salas de clases. Para realizar este proceso se conocen los cursos y la cantidad de estudiantes que estos tienen. Por otro lado la cantidad de salas es conocida. A continuación se presentan algunos de los cursos representados por su código y su sección, junto a la cantidad de estudiantes del semestre 2018-1. Al lado derecho se muestran las salas disponibles junto a su capacidad.

Curso	Sección	Cantidad de estudiantes	N° sala	Capacidad
10110	A1	45	521	45
10110	B2	44	522	33
10110	C3	45	523	33
13201	A1	31	524	20
13201	B2	14	525	20
13204	A1	26	526	56

Se le ha encargado realizar un programa en pseudo-Prolog que permita realizar la asignación de los cursos a las salas de tal forma que siempre los cursos estén en salas con la capacidad de estudiantes suficiente.

- Exprese la entrada de su programa con los cursos y salas como hechos de Prolog.
- Realice un predicado que permita determinar si a un curso es posible asignarle cierta sala.
- Realice un predicado que reciba una lista de pares de la forma `[CURSO, SECCION]` y proporcione una lista con las salas asignadas (mismo orden que la lista de entrada) para que el predicado sea verdadero.

Por ejemplo al consultar:

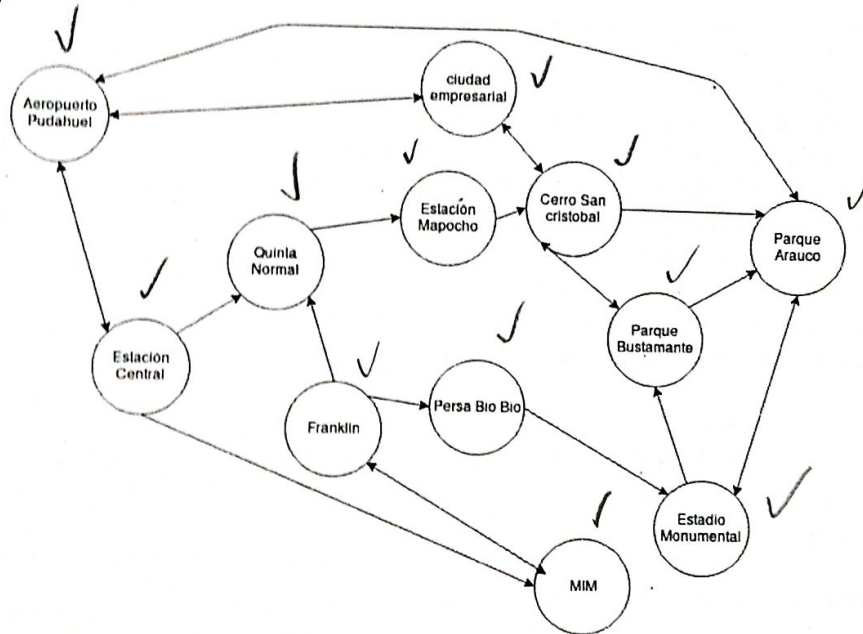
`asignarSalas([["10110", "A1"], ["10110", "B2"], ["13204", "A1"]], SALAS).`

`SALAS = [521, 526, 522].`

Se obtiene que al curso "10110" sección "A1" se le asigna la sala 521, al curso "10110" sección "B2" se le asigna la sala 526 y al curso "13278" sección "A1" se le asigna la sala 522

Ejercicio 2:

FeepEx, una empresa de distribución de correos y encomiendas se encuentra desarrollando un sistema que le permita guiar a los conductores de sus camiones y furgones de distribución. En este sistema se ha decidido incluir un módulo en Prolog que permite indicar la siguiente parada a la que debe concurrir un chofer. Por ejemplo si se tiene el siguiente mapa de paradas con sus calles (las cuales pueden ir en un solo sentido o algunas doble vía):



- Expresar el mapa de paradas como hechos en pseudo-Prolog
- Realizar un predicado en Prolog que permita saber el listado de paradas por las que se debe pasar para llegar desde A a B
- Realizar la traza para el llamado de su predicado de la pregunta anterior en el caso de ir desde la Estación Mapocho hasta el Parque Arauco

RUT Evaluador:	Ejercicio 1	Ejercicio 2
Punto base		
Descripción del diseño		
Documentación y comentarios		
Hechos		
Reglas		
TOTAL		