

Universidad Tecnológica Nacional
Facultad Regional Buenos Aires
Cátedra de Paradigmas de Programación

Forall

Autor: Carlos Lombardi – carlombardi@gmail.com

Historial de Revisiones

Versión	Revisado por	Fecha	Detalles
V 1.0	Carlos	01/10/2007	Versión inicial

0.

Esperamos que estas notas, divididas en 8 ítems, ayuden a poder manejarnos mejor con el predicado `forall/2`.

1.

El `forall` es un predicado, o sea, no conviene pensar qué "hace", sino qué relaciona, y/o cuándo se verifica.

El predicado `forall` conviene pensarlo por lo segundo, o sea cuándo se verifica.

2.

El `forall` recibe dos parámetros, los dos pueden verse como consultas. Las consultas se hacen sobre predicados, o sea que `forall` es un predicado de orden superior, porque puede manejar predicados dentro de sus parámetros.

Hagámonos la pregunta ... ¿cuándo se verifica?

Cuando a todos los que les pasa lo primero, les pasa lo segundo.

Dicho un poco más "en técnico", cuando todas las respuestas de la primer consulta son respuestas de la segunda.

Entonces, en las situaciones donde decimos "a todos los que ...blah... les pasa ...bleh..." es probable que el `forall` nos venga bien.

3.

Un ejemplo: partimos de estos hechos

```
dulce(chocolate).
```

```
dulce(caramelo).
```

```
dulce(durazno).
```

```
amargo(radicheta).
```

```
amargo(cebada).
```

```
leGusta(roque,chocolate).
```

```
leGusta(roque,radicheta).
```

```
leGusta(pepe,cebada).
```

```
% ... y muchos hechos más que describen los gustos de un grupo
de personas
```

```
colorDePelo(roque,colorado).
```

```
colorDePelo(pepe,castanio).
```

```
% ... etc. con los colores de pelo
```

```
vive(roque,buenosAires).
```

```
vive(pepe,mendoza).
```

```
vive(lucas,salliquelo).
```

```
% ... y donde vive cada persona de la que queremos hablar
```

```
ciudadGrande(buenosAires).
```

```
ciudadGrande(mendoza).
```

```
% ... y así todas las ciudades grandes
```

y queremos definir `esTierno/1`, donde decimos que una persona es tierna si todas las cosas que le gustan son dulces.

Estamos en un caso candidato a `forall`: "a todas las cosas que le gustan les tiene que pasar ser dulces". En Prolog:

```
esTierno(Pers):- forall(leGusta(Pers,Alim), dulce(Alim)).
```

... `Pers` es tierno si ... todas las cosas que le gustan son dulces, exactamente lo que dijimos en castellano.

Ahora quiero definir el predicado `alimentoCurioso/1`, un alimento es curioso si solamente le gusta a gente de pelo colorado.

Para darme cuenta que el forall me puede servir, lo pienso en términos de "a todos los que ... les tiene que pasar ...". A veer

- a todas las personas que ... les gusta el alimento
- les tiene que pasar ... ser coloradas

Queda

```
alimentoCurioso(A):- forall(leGusta(P,A), esColorado(P)).
```

4.

¿Está bien si defino `esTierno` así?

```
esTierno(P):- forall(dulce(Alim), leGusta(P,Alim)).
```

Claramente no, porque estaría pidiendo que le gusten todos los alimentos dulces.

Si programar va a consistir en definir condiciones, es relevante entender la diferencia entre todos los alimentos que le gustan son dulces

y

le gustan todos los alimentos dulces.

5.

Supongamos que hacemos esta consulta:

```
?- esTierno(roque).
```

La Pers de `esTierno` se liga con roque ... entonces el forall se va a verificar (ver la defi técnica) cuando

- todas las respuestas a la consulta
`leGusta(roque,Alim)`
- verifiquen la consulta
`dulce(Alim)`

Para cada respuesta a la consulta `leGusta(roque,Alim)`, la variable `Alim` se va a ligar, en el ejemplo hay dos respuestas, una con chocolate y otra con cebada.

La consulta correspondiente ya viene con esa variable ligada, o sea que las consultas que se tienen que verificar para que se verifique el forall son

```
esDulce(chocolate).
esDulce(cebada).
```

Volvamos a la definición: el forall se verifica si todas las respuestas a la primer consulta son respuestas de la segunda. Mirando el ejemplo de recién debería cerrar el esquema.

6.

Veamos qué pasa con las variables y la inversibilidad.

¿Será inversible el predicado `esTierno/1`? Hagamos la consulta con una variable en el argumento

```
?- esTierno(X).
```

En este caso Pers llega sin ligar al forall.

Entonces la primer consulta es

```
leGusta(Pers,Alim).
```

Para cada una de las respuestas a esta consulta, se tiene que verificar

```
esDulce(Alim)
```

donde `Alim` es lo que ligó la primer consulta.

¿Cuáles son las respuestas a la primer consulta?

Todos los pares (persona,alimento) relacionados por `leGusta`.

Entonces, el forall sólo se va a verificar si cualquier cosa que le guste **a alguien**, no importa a quién, es dulce.

Claro, no es lo que queremos. Para lograr lo que queremos, tenemos que lograr que la variable P llegue ligada al forall:

```
esTierno(P):- persona(P), forall(leGusta(P,Alim),esDulce(Alim)).
```

Una que no falla:

fíjense que siempre decimos "a todos los blah que les pasa la consulta 1, les tiene que pasar la consulta 2".

Bueno, para ese "blah" va a haber una variable, que es Alim en el caso de esTierno (si todos los alimentos que le gustan ...) y P para alimentoCurioso (si todas las personas a quienes les gusta ...). Esa variable tiene que llegar al forall sin ligar.

7.

Qué pasa si se tienen que cumplir varias condiciones: digamos que un alimento es peculiar si todas las personas a las que le gusta son colorados y porteños ... nos queda

- a todas las personas que ... les gusta el alimento
- les tiene que pasar ... ser coloradas y ser porteñas

entonces la segunda consulta es un "y" entre dos condiciones.

Si pongo

```
esPeculiar(A):-
    forall(leGusta(P,A), colorDePelo(P,colorado), vive(P,buenosAires)).
```

está mal, porque el forall lleva dos parámetros, no tres.

Necesito agrupar colorDePelo(...) y vive(...), para eso los encierro entre paréntesis, queda

```
esPeculiar(A):-
    forall(
        leGusta(P,A),
        (colorDePelo(P,colorado), vive(P,buenosAires))).
```

Pregunto: ¿está bien

```
esPeculiar(A):-
    forall(
        (leGusta(P,A), colorDePelo(P,colorado)),
        vive(P,buenosAires)).
```

?

No, porque estaría pidiendo que todos los colorados a los que les gusta el alimento vivan en Buenos Aires.

8.

Una que les queda para pensar: ahora tengo que decir que un alimento es marketinable si todas las personas a las que les gusta viven en ciudades grandes. No me interesa que el predicado sea inversible.

Tiro tres opciones: elijan la correcta y piensen por qué eligieron esa.

```
esMarketinable(A):-
    forall(leGusta(P,A), vive(P,C), ciudadGrande(C)). % opc 1
esMarketinable(A):-
    forall((leGusta(P,A), vive(P,C)), ciudadGrande(C)). % opc 2
esMarketinable(A):-
    forall(leGusta(P,A), (vive(P,C), ciudadGrande(C))). % opc 3
```