

Prolog

Particularización y Unificación

- Todas las variables están cuantificadas universalmente de forma implícita.
- Toda variable puede sustituirse por un término cualquiera siempre y cuando se haga la sustitución de forma consistente.
- Las variables son locales a la cláusula donde se definen.
- Una variable que aparezca una sola vez, se recomienda sustituirla por el símbolo '_'. Distintos '_' se pueden particularizar de distintas formas.

Ejemplos

```
hermanos(X,Y) :- padre(X,Z), padre(Y,Z).
```

```
capital(PAIS, _).
```

```
capital(_, CAP).
```

Objetivo de la Unificación

- Unificar una meta con un hecho, o
- Unificar una meta con el encabezado de una cláusula.

Ejemplos

Las siguientes consultas se resuelven por unificación

Base de datos

```
suma(X, 0, X).
```

Consultas

```
1 ?- suma(6, Y, Z).  
Y = 0,  
Z = 6.
```

```
2 ?- suma(9, 0, Z).  
Z=9.
```

```
3 ?- suma(X, 2, Z).  
false.
```

```
4 ?- suma(X, Y, Z).  
X = Z,  
Y = 0.
```

Unificación

Base de datos

```
padre(b,a). % El padre de b es a
padre(c,a).
padre(e,d).
padre(f,d).
padre(d,g).
padre(a,g).
hermanos(X,Y) :- padre(X,Z), padre(Y,Z), X\=Y.
tio(X,Y) :- padre(Y,Z), hermanos(X, Z).
```

Consultas

```
1 ?- tio(a,f).
true.
```

Observar como la unificación en cada cláusula es distinta.

Cláusula tio: $X=a$, $Y=f$, $Z=d$

Cláusula hermanos: $X=a$, $Y=d$, $Z=g$

Backtracking

- Es el proceso mediante el cual el Prolog halla soluciones o determina que no existen.
- Consideremos el siguiente ejemplo:

Base de datos

```
compra(X,Y) :- sevende(Y), gusta(X,Y), bueno(Y).  
sevende(vestido).  
sevende(sombrero).  
sevende(zapatos).  
gusta(jaime,zapatos).  
gusta(maria,vestido).  
gusta(maria,sombrero).  
bueno(sombrero).
```

Backtracking 2

- Se hace la consulta

```
1 ?- compra(Z,U).
```

- Como se resuelve?

- 1) Se unifican $\text{compra}(Z,U)$ con $\text{compra}(X,Y)$
 $\text{compra}(Z,U) \text{ :- se vende}(U), \text{ gusta}(Z,U), \text{ bueno}(U).$
- 2) Se intenta satisfacer las metas. $U=\text{vestido}$
satisface $\text{se vende}(U)$.
- 3) $\text{gusta}(Z,\text{vestido})$ se satisface con $Z=\text{maria}$
- 4) $\text{bueno}(\text{vestido})$ falla

Backtracking 3

- 5) Se retrocede a buscar más alternativas para `gusta(Z,vestido)` : No hay más, entonces falla.
- 6) Se retrocede para intentar alternativas para `sevende(U)`. Encuentra `U=sombrero` satisface la meta.
- 7) `gusta(Z,sombrero)` se satisface con `Z=maria`.
- 8) `bueno(U)` se satisface con `U=sombrero`. La cláusula tiene éxito

Backtracking 4

Cuando una cláusula tiene éxito, Prolog imprime el unificador.

Si el usuario oprime ';', Prolog retrocede nuevamente, e intenta encontrar otro unificador que cumpla la regla.

Resolución

- Si una cláusula **a** contiene como meta a una cláusula **b**, entonces haciendo las particularizaciones necesarias se puede reemplazar **b** por el cuerpo de **b** en **a**.
- La nueva cláusula resultante se llama el *resolvente* de **a** con **b**.

Resolución - Ejemplo

Base de datos

```
compra(X,Y) :- oportunidad(Y), gusta(X,Y).  
oportunidad(Z) :- preciobajo(Z), altacalidad(Z).  
oportunidad(Z) :- recomendado(Z).  
oportunidad(sombrero).
```

- El resolvente de la 1a cláusula con la segunda es
`compra(X,Y) :- preciobajo(Y), altacalidad(Y), gusta(X,Y).`
- El resolvente de la 1a cláusula con la tercera es
`compra(X,Y) :- recomendado(Y), gusta(X,Y).`
- El resolvente de 1 con 4 es
`compra(X,sombrero) :- gusta(X,sombrero).`
Un hecho no tiene cuerpo, se reemplaza por vacío.

Motor de Prolog

- Para resolver las consultas, Prolog utiliza el proceso de resolución:
 1. Se resuelven todas las metas
 2. La consulta tiene éxito cuando el cuerpo queda vacío
- Una consulta es una regla sin encabezado. La consulta se resuelve exitosamente cuando se llega a la cláusula vacía. De lo contrario falla.