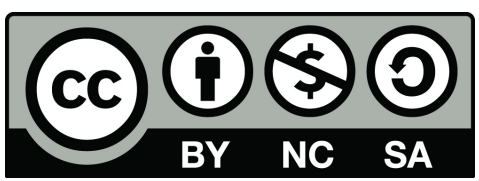
# Inteligencia Artificial: Prolog

## Unificación en Prolog

Universidad de La Laguna

Christopher Expósito-Izquierdo[1],
Belén Melián-Batista[2]

{cexposit[1], mbmelian[2]}@ull.es
Universidad de La Laguna (España)



# Contenidos

- Satisfacer objetivos
- Concepto de unificación
- Casos de unificación

# Satisfacer objetivos

Comprender cómo Prolog resuelve las consultas realizadas a través del intérprete del lenguaje usado permite reducir el esfuerzo empleado en la escritura y entendimiento de programas. En este sentido, considera una consulta como la siguiente:

```
? padre(X,Y), mujer(Y), propietario(X).
```

Prolog tratará de dar respuesta a esta consulta. Para ello, trata de satisfacer los objetivos establecidos en la consulta (3 en este caso particular). Los objetivos tratan de ser satisfechos de izquierda a derecha. Esto es, en primer lugar `padre(X,Y)`, luego `mujer(Y)` y finalmente `propietario(X)`.

# Satisfacer objetivos

- Cuando Prolog trata de satisfacer un determinado objetivo donde éste involucra variables (por ejemplo: `padre(X,Y)`), lo que trata de hacer es enlazar dichas variables a valores concretos establecidos en la base de conocimiento. Por ejemplo, `X` a `antonio` e `Y` a `andrea`.
- La secuencia completa de objetivos es satisfecha siempre que todos y cada uno de los objetivos individuales tengan éxito.
- En caso de que una consulta sea satisfecha, Prolog retorna como salida todos aquellos valores que pueden tomar las variables empleadas.

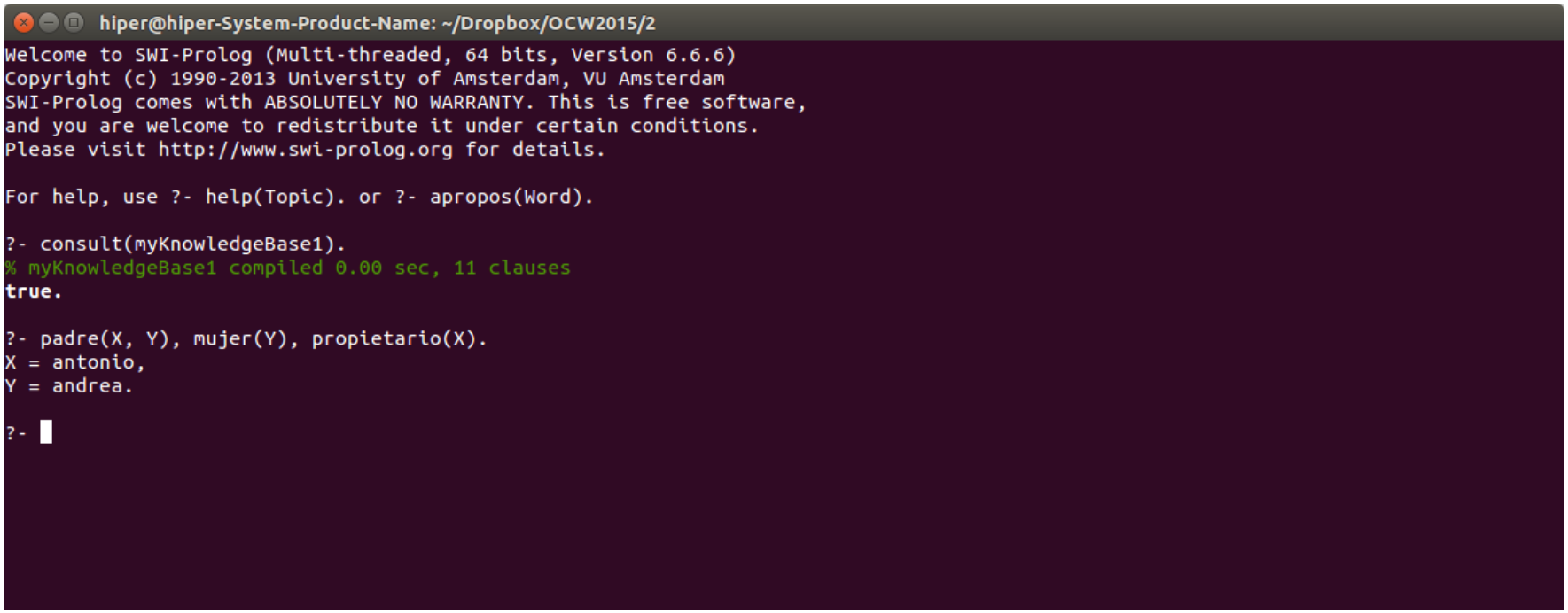
# Satisfacer objetivos

Consideremos la siguiente base de conocimiento:

- `mujer(andrea).`
- `mujer(sandra).`
- `hombre(pepe).`
- `hombre(luis).`
- `hombre(antonio).`
- `propietario(pepe).`
- `propietario(antonio).`
- `padre(luis, sandra).`
- `padre(antonio, pepe).`
- `padre(antonio, andrea).`

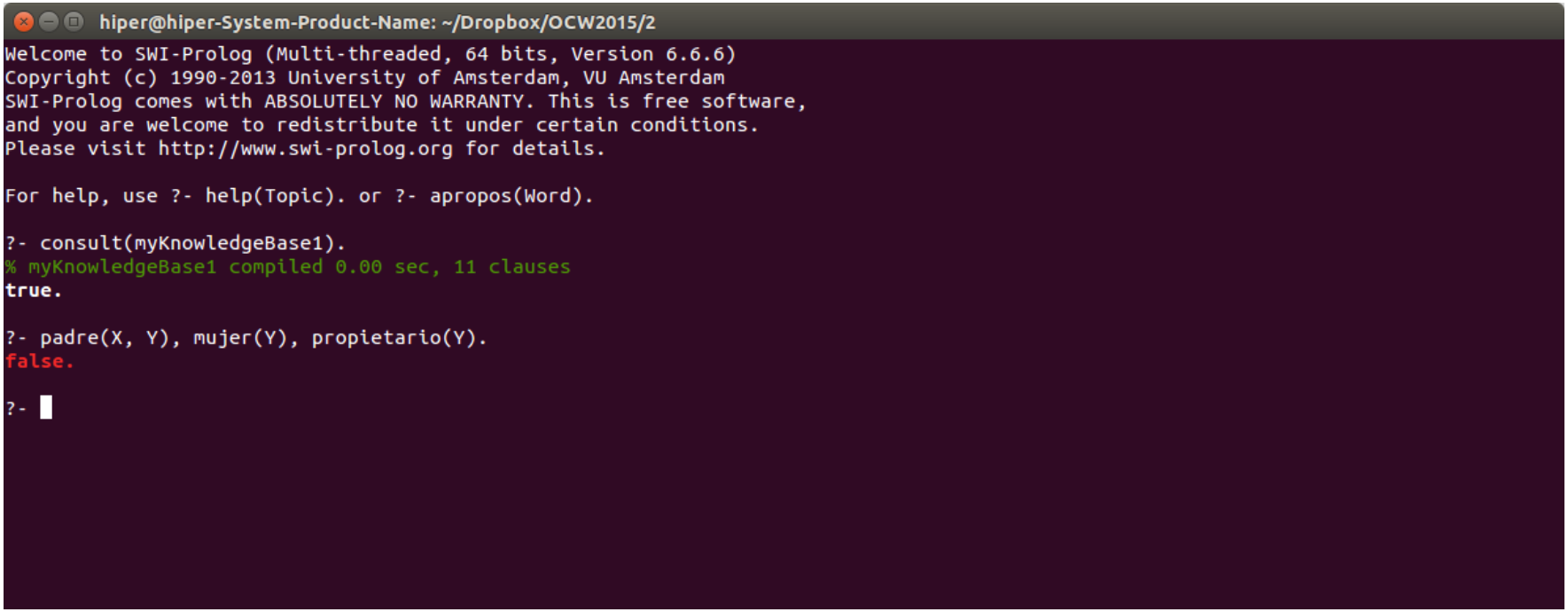
# Satisfacer objetivos

Si realizamos la consulta propuesta inicialmente obtendremos lo siguiente:



# Satisfacer objetivos

Cuando se realiza esta otra consulta obtenemos un resultado negativo:



# Satisfacer objetivos

- Los objetivos establecidos en las consultas realizadas a través del intérprete de Prolog pueden ser únicamente átomos y términos complejos. Elementos como variables, números, listas, etc. no pueden emplearse aisladamente como objetivos a satisfacer.

- Cada objetivo establecido en una consulta trata de ser satisfecho buscando ordenadamente de arriba a abajo en la base de conocimiento (o en los predicados predefinidos en Prolog) de tal manera que se enlace con la cabeza de alguna regla. Por ejemplo, el objetivo `mujer(Y)` podría ser enlazado con el hecho `mujer(andrea)`. El resultado obtenido sería

```
Y = andrea.
```

# Satisfacer objetivos

- Tal como se vió anteriormente, cada vez que se establece un objetivo dentro de una consulta que no puede ser satisfecho por Prolog el resultado de dicha consulta falla.
- En base a lo anterior, puede decirse que Prolog asume una visión de *mundo cerrado*. Esto significa que toda conclusión que no pueda ser probada a partir de los hechos y reglas establecidas en la base de conocimiento se considera falsa por defecto.
- Prolog no emplea puntos de vista intermedios como *desconocido* o *no probado*.

# Concepto de unificación

Cuando el usuario establece un objetivo en una consulta, Prolog recorre todos los hechos y las reglas existentes en la base de conocimiento tratando de enlazar dicho objetivo con alguna cláusula. Las cláusulas son recorridas desde la primera hasta la última hasta que algún emparejamiento pueda ser realizado.

Si consideramos la base de conocimiento anteriormente mostrada y el objetivo `padre(X,Y)`, en primer lugar se analiza el hecho `mujer(andrea)`, luego `mujer(sandra)`, y así sucesivamente hasta llegar a `padre(luis, sandra)`. Este hecho satisface el objetivo establecido. Nótese que posteriormente se pasaría a analizar los restantes objetivos establecidos originalmente en la consulta.

# Concepto de unificación

La *unificación* es la forma empleada por Prolog para realizar el emparejamiento de objetivos y cláusulas. Habitualmente una o varias variables se encuentran involucradas en el proceso. Se persigue en estos casos obtener términos iguales. Este proceso se denomina enlazado de variables a valores.

Por ejemplo, los términos `mujer(Y)` y `mujer(andrea)` son perfectamente unificables mediante el enlazado de la variable `Y` al átomo `andrea`. Podemos decir por simplicidad que `Y` toma el valor `andrea`: `Y = andrea`

De forma similar, `X` toma el valor `antonio` e `Y` toma el valor `andrea` en la consulta inicialmente planteada: `X = antonio`

# Concepto de unificación

- Las variables empleadas en una consulta en Prolog están inicialmente no enlazadas. Esto significa que no tienen valor asignado alguno.
- A diferencia de la mayoría de los lenguajes de programación, una vez una valor ha sido asignado a una determinada variable, ésta puede tomar otro valor mediante un proceso de recursión.

# Concepto de unificación

El proceso de unificación llevado a cabo a la hora de comprobar si dos términos unifican se puede resumir tal como sigue:

1. Si los términos son constantes hay que comprobar si son el mismo término. Si lo son, la unificación tiene éxito. En otro caso, la unificación falla.

2. Si los términos son compuestos hay que comprobar en primer lugar sus funtores y aridades. Si tienen el mismo funtor y aridad, hay que comprobar si los argumentos unifican por pares. Si esto sucede, la unificación tiene éxito. En otro caso, la unificación falla.

# Casos de unificación I

El caso más sencillo es aquel en que un átomo se unifica con otro átomo. En este caso, la unificación tiene éxito si y solo si ambos átomos son el mismo. De esta manera, se deben considerar los siguientes ejemplos:

- Los átomos `andrea` y `andrea` unifican
- Los átomos `andrea` y `'andrea'` unifican debido a que las comillas no son consideradas como parte del átomo
- Los átomos `andrea` y `sandra` no unifican debido a que son átomos distintos

# Casos de unificación II

Un segundo caso a considerar en el proceso de unificación es aquel en que un átomo trata de ser unificado con un término complejo. Consideremos el caso en que el átomo `andrea` quiere ser unificado con el término complejo `gusta(sandra, antonio)`. Este tipo de unificaciones siempre fallan.

# Casos de unificación III

El último de los casos a considerar en un proceso de unificación es aquel en que dos términos complejos tratan de unificarse. Consideremos el caso en que el término complejo `gusta(X, Y)` y `gusta(sandra, antonio)` tratan de ser unificados. Otro ejemplo sería la unificación de `gusta(sandra, Y)` y `mujer(andrea)`. Tal como se describió anteriormente, la unificación falla a menos que los dos términos complejos tengan el mismo funtor y la misma aridad. Por tanto, la unificación en el primero de los casos es exitosa mientras que falla en el segundo caso.

# Inteligencia Artificial: Prolog

## Unificación en Prolog

Universidad de La Laguna

Christopher Expósito-Izquierdo[1],
Belén Melián-Batista[2]

{cexposit[1], mbmelian[2]}@ull.es
Universidad de La Laguna (España)