

Nombre: Luis Orellana P.

Completar el ejemplo de clips sobre la familia agregar:

Tíos

Sobrinos

Primos

Divorcios

```
(defrule progenitor
```

```
(padre-de ?padre ?hijo)
```

```
(marido-de ?padre ?madre)
```

```
=>
```

```
(assert (progenitor-de ?madre ?hijo))
```

```
)
```

```
(defrule esposa
```

```
(marido-de ?hombre ?mujer)
```

```
=>
```

```
(assert (esposa-de ?mujer ?hombre))
```

```
(printout t ?mujer " es esposa de " ?hombre crlf)
```

```
)
```

```
(defrule padre
```

```
(progenitor-de ?padre ?hijo)
```

```
(hombre ?padre)
```

```
=>
```

```
(assert (padre-de ?padre ?hijo))
```

```
(printout t ?padre " es padre de " ?hijo crlf)
```

```
)
```

```
(defrule madre
```

```
(progenitor-de ?madre ?hijo)
```

```
(mujer ?madre)
```

```
=>
```

```
(assert (madre-de ?madre ?hijo))
```

```
(printout t ?madre " es madre de " ?hijo crlf)
)
```

```
(defrule abuelos
(progenitor-de ?padre ?hijo)
(progenitor-de ?hijo ?nieto)
=>
(assert (abuelos-de ?padre ?nieto))
)
```

```
(defrule abuelo
(abuelos-de ?padre ?nieto)
(hombre ?padre)
=>
(assert (abuelo-de ?padre ?nieto))
(printout t ?padre " es abuelo de " ?nieto crlf)
)
```

```
(defrule abuela
(abuelos-de ?madre ?nieto)
(mujer ?madre)
=>
(assert (abuela-de ?madre ?nieto))
(printout t ?madre " es abuela de " ?nieto crlf)
)
```

```
(defrule hermanos-padre
(padre-de ?padre ?hijo1) (padre-
de ?padre ?hijo2)
```

```
(test (neq ?hijo1 ?hijo2))
=>
```

```
(assert (hermanos ?hijo1 ?hijo2))  
)
```

```
(defrule hermanos-madre  
  (madre-de ?madre ?hijo1)  
  (madre-de ?madre ?hijo2)  
  (test (neq ?hijo1 ?hijo2))  
=>  
  (assert (hermanos ?hijo1 ?hijo2))  
)
```

```
(defrule hermana  
  (hermanos ?hijo1 ?hijo2)  
  (mujer ?hijo1)  
=>  
  (assert (hermana-de ?hijo1 ?hijo2))  
  (printout t ?hijo1 " es hermana de " ?hijo2 crlf)  
)
```

```
(defrule hermano  
  (hermanos ?hijo1 ?hijo2)  
  (hombre ?hijo1)  
=>  
  (assert (hermano-de ?hijo1 ?hijo2))  
  (printout t ?hijo1 " es hermano de " ?hijo2 crlf)  
)
```

```
(defrule tio  
  (hermanos ?hijo1 ?hijo2)  
  (hombre ?hijo2)  
  (progenitor-de ?hijo1 ?hijo)
```

=>

```
(assert (tio-de ?hijo2 ?hijo))  
(printout t ?hijo2 " es tio de " ?hijo crlf)  
)
```

```
(defrule tia  
(hermanos ?hijo1 ?hijo2)  
(mujer ?hijo2)  
(progenitor-de ?hijo1 ?hijo)
```

=>

```
(assert (tia-de ?hijo2 ?hijo))  
(printout t ?hijo2 " es tia de " ?hijo crlf)  
)
```

```
(defrule sobrino  
(hermanos ?hijo1 ?hijo2)  
(progenitor-de ?hijo1 ?hijo)  
(hombre ?hijo)
```

=>

```
(assert (sobrino-de ?hijo ?hijo2))  
(printout t ?hijo " es sobrino de " ?hijo2 crlf)  
)
```

```
(defrule sobrina  
(hermanos ?hijo1 ?hijo2)  
(progenitor-de ?hijo1 ?hijo)  
(mujer ?hijo)
```

=>

```
(assert (sobrina-de ?hijo ?hijo2))  
(printout t ?hijo " es sobrina de " ?hijo2 crlf) )  
(defrule primo
```

```
(hermanos ?hijo1 ?hijo2)
(progenitor-de ?hijo1 ?hijo3)
(progenitor-de ?hijo2 ?hijo4)
(hombre ?hijo3)
=>
(assert (primo-de ?hijo3 ?hijo4))
(printout t ?hijo3 " es primo de " ?hijo4 crlf)
)
```

```
(defrule prima
(hermanos ?hijo1 ?hijo2)
(progenitor-de ?hijo1 ?hijo3)
(progenitor-de ?hijo2 ?hijo4)
(mujer ?hijo3)
=>
(assert (prima-de ?hijo3 ?hijo4))
(printout t ?hijo3 " es prima de " ?hijo4 crlf)
)
```

```
(deffacts inicio
(hombre Diego)
(mujer Lili)
(hombre Juan)
(mujer Mayra)
(hombre Maria)
(mujer Rosa)
(hombre Jose)
(mujer Luisa)
(hombre Roberto)
(mujer Luisana)
(mujer Danny)
(progenitor-de Diego Danny) (progenitor-de Luisana Diego)
(marido-de Diego Lili)
(progenitor-de Luisana Juan)
```

(progenitor-de Juan Mayra)

(tio-de Juan Danny)

(sobrina-de Danny Juan)

(prima Danny Mayra)

)