Contestaciones: Proyecto 1b

Aramis E. Matos Nieves

a) El programa genera las descripciones estructurales adecuadas en el sentido que a base de unas restricciones estructurales (es decir a base de reglas creadas por clase naturales y no especificas a un fono en particular) y unas reglas de combinación se generan silabas lícitas del español. Por ejemplo, las silabas están compuestas de un ataque y una rima y esa rima está compuesta de un núcleo y una coda. El programa modela este comportamiento mediante hechos que asertan que una silaba es silaba si y solo si tiene una ataque A y una rima B.

b) El modelo genera $41,000$ silabas. Esto es una sobregeneración masiva. Esto se deba a las reglas de inferencia para las rimas y los ataques son demasiado permisibles debido a errores lógicos.

   Por ejemplo, la cantidad de rimas posibles con 3 fonos, se supone que que sea 165. Sin embargo, el programa genera 270.

   Lo mismo ocurre con la cantidad de silabas posibles con 2 fonos, se supone que sea 269 y salen 273.

c) Si se considera que la manera en la cual se están generando las no es mediante una lista exhaustiva de posibles silabas sino unas reglas inferencia, el tiempo de rendimiento no es malo. Por ejemplo:

   - *Silbas/2*

     ```
     ?- time(silabas(_,_)).
     % 70,740,116 inferences, 18.250 CPU in 18.252 seconds
         (100% CPU, 3876199 Lips)
     ```

   - *rimas/3*

     ```
     ?- time(rimas(_,_,_)).
     ```

```
% 366,100 inferences, 0.206 CPU in 0.206 seconds (100% CPU
    , 1775627 Lips)
true.
```

- *ataques/3*

```
?- time(ataques(_,_,_)).
% 2,239,528 inferences, 0.723 CPU in 0.723 seconds (100%
    CPU, 3097738 Lips)
true.
```

d) Los problemas posteriormente mencionados pueden ser arreglados por revisando las reglas de inferencia para asegurar que se traducienron correctamente y alejandose de una descripción estructural a una descripción extensional.

Debido a que el programa intenta describir estructuralmente la silaba utilizando reglas de inferencia y no hechos de todas las posibles combinaciones de fonemas de español listadas extencionalmente. Esto incurre un penalidad substancial en el rendimiento del programa.

Por ejemplo, si se describe estructuralmente la limitación del español de no tener las subcadenas *ji* o *wu* el programa presenta lo siguiente:

```
?- time(silabas(_,_)).
% 70,740,116 inferences, 18.250 CPU in 18.252 seconds (100%
    CPU, 3876199 Lips)
```

Sin embargo, si se describen extencionalmente, el programa presenta lo siguiente:

```
?- time(silabas(_,_)).
% 51,598,708 inferences, 15.624 CPU in 15.626 seconds (100%
    CPU, 3302579 Lips)
```

Es claro entonces que la descripción estructural puede sacrificar rendimiento.