

## HW#2 FAQ:

**Q:** How do I work with the values returned by UNIQUE-GAP? (UNIQUE-GAP 'HUMAN) returns something like #:HUMAN3211, but when I type that at the prompt, Lisp says the variable is undefined!

**A:** This is an annoying technical limitation of UNIQUE-GAP. The return value is an “uninterned symbol,” which is not accessible by name. However, as long as we take care to save the return value somewhere, we can still assign/retrieve values from it. The following example Lisp session illustrates this:

```
(setq a (unique-gap 'human)))  
->#:HUMAN3070
```

```
a  
->#:HUMAN3070
```

```
(set a 'test)  
->TEST
```

```
a  
-> #:HUMAN3070
```

```
(eval a)  
->TEST
```

```
(setq b (list 'slot a))  
->(SLOT #:HUMAN3070)
```

```
(eval (second b))  
->TEST
```

(notice that even though I am unable to reference the unique gap variable directly, I was able to put it in a slot-filler list, and access its value using EVAL)

You can still test your work using the examples in the prompt (being able to actually access the variables at the interpreter will help with debugging), but be aware that your code must still work even if the GAPs are uninterned.

**Q:** Why does the second example for D-FILL bind (VEHICLE) when the PRED is **not** “PHYS-OBJ”?

**A:** This wasn't clear in the specs (now updated). D-FILL should be looking for a frame whose predicate “ISA” PRED. Since according to the ONTology, VEHICLE ISA PHYS-OBJ, D-FILL will successfully bind CON888.

**Q:** Are there typos in the specs for D-LAST-NAME, TOKENIZE?

**A:** Yes. Please see the corrected PDF for the assignment on CourseWeb.

**Q:** How do I execute my demons without evaluating their CON arguments?

**A:** See the function “apply”: <http://www.cs.cmu.edu/Groups/AI/html/cltl/clm/node81.html>