UNA. Escuela de Informática
EIF400 Paradigmas de Programación II-2017
Dr. Carlos Loría-Sáenz

Ejercicio Prolog Kokoslan

Preguntas:

Queremos extender el demo sobre `Kokoslan` para prototipar conceptos del proyecto programado asociado y practicar `Prolog`. Este ejercicio asume que Ud. ya está familiarizado con el proyecto `Kokoslan`. La idea es que le sirva para realizarlo mejor y estudiar para el examen. Sirva también para contrastar `Prolog` vs `Kotlin/Java`.

Considere el código en directorios `work/src/` Ahí se implementa como demo un emisor (`emiter.pl`), un evaluador (`evaluator.pl`), un contexto de evaluación (`context.pl`) y un caso de uso (`main.pl`). Además se incluye sólo el cascarón de un compilador (`compiler.pl`) que actualmente no hace nada más que dar un warning. El emisor genera en `output` la salida del caso probado según la produzca el compilador.

Queremos extender el evaluador y el compilador para manejar más casos de prueba. Por ahora sólo se maneja el caso `simple.kl` que está codificado "a mano" en `main.pl`. Este caso representa el siguiente programa (en `cases/simple.kl`)

```
let x = 666
let y = x
x + y
```

El compilador está pensado para que compile patrones en lambdas antes de poder evaluarlas. El emisor imprimiría el código compilado para su inspección y revisión.

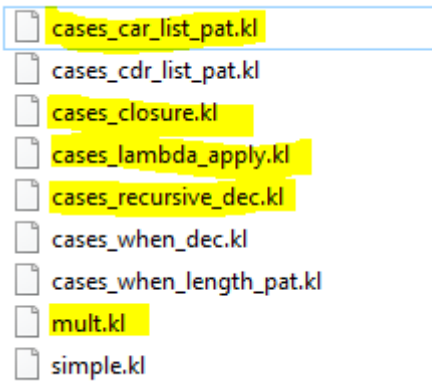
La forma de proceder es que Ud. codifica a mano un caso de prueba en `main.pl` (siga el modelo ahí montado según le explica el docente). Y extiende el compilador y/o el evaluador según se requiera.

La codificación se hace manualmente porque este demo no cuenta aún con un parser.

Corrida de `simple.kl` . En directorio `work`:

```
PP:swipl -q -g main('simple.kl'),halt -s src\main.pl
>>> Starting Kokoslan Demo in Prolog for 'simple.kl' case <<<
***[WARNING] Compiler must be implemented! ***
>>> Evaluator starts <<<
>>> 1332
>>> Ending Demo in Prolog <<<
PP:
```

Usando los casos en cases marcados en amarillo. En cada caso primero codifique en `main.pl` el `AST` en `Prolog`. El caso `mult.kl` vale 0 puntos (se hará con el docente), los demás valen 25 cada uno.



1) Logre el caso `mult.kl`
2) Logre `cases_closure.kl`
3) Logre `lambda_apply.kl`
4) Logre `cases_car_list_pat.kl`
5) Logre `cases_recursive_dec.kl`

El resto queda como material del examen.