

Lenguajes de Programación 2020-1

Facultad de Ciencias UNAM

Ejercicio Semanal 11

Sandra del Mar Soto Corderi
Edgar Quiroz Castañeda

7 de noviembre de 2019

1. Considera el siguiente programa

```
exception Excpt of int ;  
fun twice (f , x ) = handle f ( f ( x ) ) with Excpt ( x ) = > x ;  
fun pred ( x ) = if iszero x then raise Excpt ( x ) else x - 1;  
fun smart ( x ) = handle 1 + pred ( x ) with Excpt ( x ) = > 1;
```

Evalúa las siguientes expresiones utilizando la semántica operacional de la máquina K extendida para manejar excepciones con valor. (pueden obviarse algunos pasos pero no los que involucren manejo de excepciones. En cada caso describe qué excepción fue lanzada y en dónde.)

- `twice(smart, 0)`

Cuando se tenga una aplicación con valores bloqueados, se saltarán los pasos hasta el cálculo que los concierna, ignorando los pasos intermedios para verificar que son valores.

```
□ > app(twice, (smart, 0))  
→k □ > twice[f := twice, x := 0]  
→β □ > handle(app(smart, app(smart, 0)), x.x)  
→k handle(_, x.x) > app(smart, app(smart, 0))  
→k app(smart, _), handle(_, x.x) > app(smart, 0)  
→k app(smart, _), handle(_, x.x) > smart[x := 0]  
→β app(smart, _), handle(_, x.x) > handle(sum(1, app(pred,0)), x.1)  
→k handle(_, x.1), app(smart, _), handle(_, x.x) > sum(1, app(pred,0))  
→k sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) > app(pred,0)  
→k sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) > pred[x:=0]  
→β sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) > if(iszero(0), raise(0), dif(0, 1))  
→k if(_, raise(0), dif(0, 1)), sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) > iszero(0)  
→k if(_, raise(0), dif(0, 1)), sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) < true  
→k sum(1, _), handle(_, x.1), app(smart, _), handle(_, x.x) >> raise(0)
```

En este punto se lanza una excepción de tipo `Excpt` con un valor de 0.

$\rightarrow_k \text{handle}(_, x.1), \text{app}(\text{smart}, _), \text{handle}(_, x.x) \gg \text{raise}(0)$
 $\rightarrow_k \text{app}(\text{smart}, _), \text{handle}(_, x.x) \succ 1[x:=0]$
 $\rightarrow_\beta \text{app}(\text{smart}, _), \text{handle}(_, x.x) \succ 1$
 $\rightarrow_k \text{handle}(_, x.x) \succ \text{smart}[x:=1]$
 $\rightarrow_\beta \text{handle}(_, x.x) \succ \text{handle}(\text{sum}(1, \text{app}(\text{pred}, 1)), x.1)$
 $\rightarrow_k \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{sum}(1, \text{app}(\text{pred}, 1))$
 $\rightarrow_k \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{app}(\text{pred}, 1)$
 $\rightarrow_k \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{pred}[x:=1]$
 $\rightarrow_\beta \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{if}(\text{iszero}(1), \text{raise}(1), \text{dif}(1, 1))$
 $\rightarrow_k \text{if}(_, \text{raise}(1), \text{dif}(1, 1)), \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{iszero}(1)$
 $\rightarrow_k \text{if}(_, \text{raise}(1), \text{dif}(1, 1)), \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \prec \text{false}$
 $\rightarrow_k \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{dif}(1, 1)$
 $\rightarrow_k \text{sum}(1, _), \text{handle}(_, x.1), \text{handle}(_, x.x) \prec 0$
 $\rightarrow_k \text{handle}(_, x.1), \text{handle}(_, x.x) \succ \text{sum}(1, 0)$
 $\rightarrow_k \text{handle}(_, x.1), \text{handle}(_, x.x) \prec 1$
 $\rightarrow_k \text{handle}(_, x.x) \prec 1$
 $\rightarrow_k \square \prec 1$

- `twice(pred, 1)`