

Práctica Uno

Especificación y TADS. AyEd2

Agosto 2022

1. Ejercicio 1

a) duplicar: $sec(\alpha) \rightarrow sec(\alpha)$

```
duplicar(s) ≡ if vacia?(s)
               then <>
               else prim(s).prim(s).duplicar(fin(s))
```

b) $\bullet \leq \bullet$: $sec(\alpha) \times sec(\alpha) \rightarrow bool$

```
s ≤ t ≡ if vacio?(s)
          then true
          else if vacio?(t)
                then false
                else prim(s) < prim(t) ∨ ¬prim(s) > prim(t) ∧ fin(s) ≤ fin(t)
```

c) reverso: $sec(\alpha) \rightarrow sec(\alpha)$

```
reverso(s) ≡ if vacio?(s)
               then <>
               else ult(s).reverso(com(s))
```

d) capicua: $sec(\alpha) \rightarrow bool$

```
capicua(s) ≡ if vacio?(s)
               then true
               else (prim(s) = ult(s)) ∧ (capicua(fin(com(s))))
```

e) EsPrefijo?: $sec(\alpha) \times sec(\alpha) \rightarrow bool$

```
EsPrefijo?(s,t) ≡ if vacia?(s)
                   then true
                   else if vacia?(t)
                         then false
                         else EsPrefijo?(fin(s), fin(t))
```

f) Buscar: $sec(\alpha) \times sec(\alpha) \rightarrow nat$

```
Buscar(s,t) ≡ if EsPrefijo?(s,t)
                then 0
                else 1+Buscar(s, fin(t))
```

g) EstáOrdenada?: $sec(\alpha) \rightarrow bool$

```
EstáOrdenada?(s) ≡ ¬vacía?(fin(s)) ∧L
                  prim(s) < prim(fin(s)) ∧
                  EstáOrdenada(fin(s))
```

i) CantidadApariciones: $sec(\alpha) \times \alpha \rightarrow nat$

```
CantidadApariciones(s, a) ≡ if vacia?(s)
                              then 0
                              else if prim(s) = a
                                    then 1+CantidadApariciones(fin(s))
                                    else CantidadApariciones(fin(s))
```

g) EsPermutación?(s,t): $sec(\alpha) \times sec(\alpha) \rightarrow bool$

```
EsPermutación?(s,t)  $\equiv$  if vacia?(s)
    then true
    else CantidadApariciones(prim(s), s) = CantidadApariciones(prim(s), t)  $\wedge$ 
        EsPermutación(fin(s), t)
```

2. Ejercicio 2

a) esHoja?: $ab(\alpha) \rightarrow bool$

```
esHoja?(a)  $\equiv$  if nil?(a)
    then false
    else (nil?(izq(a)  $\wedge$  nil?(der(a))))
```

b) #Hojas: $ab(\alpha) \rightarrow nat$

```
#Hojas(a)  $\equiv$  if nil?(a)
    then 0
    else if esHoja(a)
        then 1 + #Hojas(izq(a)) + #Hojas(der(a))
        else #Hojas(izq(a)) + #Hojas(der(a))
```

c) DegeneradoAlzquierda: $ab(\alpha) \rightarrow bool$

```
DegeneradoAlzquierda(a)  $\equiv$  if nil?(a)
    then false
    else if esHoja(a)
        then true
        else nil?(der(a))  $\wedge$  DegeneradoAlzquierda(izq(a))
```

g) EsSimetrico?: $ab(\alpha) \rightarrow bool$

```
EsSimetrico?(a)  $\equiv$  if nil?(a)
    then true
    else der(a) = espejo(izq(a))  $\wedge$ 
        izq(a) = espejo(der(a))
```

3. Ejercicio 3

a) PartesDe: $conj(nat) \rightarrow conj(conj(nat))$

```
PartesDe(c)  $\equiv$  if  $\emptyset?$ (c)
    then  $\{\emptyset\}$ 
    else Agt(dameUno(c), PartesDe(sinUno(c)))  $\cup$  PartesDe(sinUno(c))
```

Agt: $nat \times conj(conj(nat)) \rightarrow conj(conj(nat))$

```
Agt(a, c)  $\equiv$  if  $\emptyset?$ (c)
    then  $\{\}$ 
    else Agt(a, dameUno(c))  $\cup$  Agt(a, sinUno(c))
```

b) combinacionesDeK: $conj(\alpha) \times nat \rightarrow conj(conj(\alpha))$

```
combinacionesDeK(c, n)  $\equiv$  if  $\emptyset?$ (c)
    then  $\emptyset$ 
    else filtrarPorN(PartesDe(c), n)  $\cup$  combinacionesDeK(sinUno(c), n)
```

filtrarPorN: $conj(conj(\alpha)) \times nat \rightarrow conj(conj(\alpha))$

```
filtrarPorN(c, n)  $\equiv$  if  $\emptyset?$ (c)
    then  $\emptyset$ 
    else if #dameUno(c) = n
        then Agt(dameUno(c), filtrarPorN(sinUno(c), n))
        else filtrarPorN(sinUno(c), n)
```

Ejercicio 4

a) $\text{tieneValor}: \text{dicc}(\alpha, \text{nat}) \times \text{nat} \rightarrow \text{bool}$

```
tieneValor(d, n)  $\equiv$   $\emptyset?$ (claves(d))  
    then false  
    else obtener(dameUno(claves(d)), d) = n  $\vee$   
        tieneValor(borrar(dameUno(claves(d)), d), n)
```

b) $\text{clavesMayoresA}: \text{dicc}(\alpha, \text{nat}) \times \text{nat} \rightarrow \text{conj}(\alpha)$

```
clavesMayoresA(d, n)  $\equiv$  if  $\emptyset?$ (claves(d))  
    then  $\emptyset$   
    else if dameUno(claves(d))  $\geq$  n  
        then Ag(dameUno(claves(d)),  
            clavesMayoresA(borrar(dameUno(d)), d))  
        else clavesMayoresA(borrar(dameUno(d)), d)
```