Salvador Marti Roman
Ramon Ruiz Dolz

Computabilidad y complejidad: 3CO21

```
(*Ejercicio 1*)
Ejercicio1[grammar_] :=
 Module[{aux, n, i,  p, t, pright, pleft, rightlist,  k, j, sol, element},
  aux = grammar;
  n = aux[[1]]; (*N, auxiliares*)
  p = aux[[3]]; (*P, producciones*)
  sol = {};
  For[k = 1, k ≤ Length[p], k++,

   pright = p[[k]][[2]]; (*Parte derecha de la producción*)
   pleft = p[[k]][[1]]; (*Parte izquierda de la producción*)

   For[j = 1, j ≤ Length[pright], j++,

    rightlist = pright[[j]]; (*Una lista de la parte derecha*)
    t = True;

    For[i = 1, i ≤ Length[rightlist], i++,

     element = rightlist[[i]];
     (*Comprueba si algún elemento
      de la lista de la parte derecha pertenece a N*)
     If[MemberQ[n, element] == True, t = False];

    ];
    If[t == True, AppendTo[sol, pleft]];
   ];
  ];
  sol = Union[Flatten[sol]];
  Return[sol];
 ]


Ejercicio1[{{S, U, V, T}, {a, b}, {{{S}, {{U}, {V}}}, {{U}, {{a, a, b}, {a, a, b}}},
   {{V}, {{b, V}, {T, b, T}}}, {{T}, {{a, T, b, T}, {b, T, a, T}, {{}}}}}, S}]
{T, U}
```

```
(*Ejercicio 2*)
Ejercicio2[grammar_] := Module[{aux, n, p, pright, pleft, rightlist, j, k, sol},
  aux = grammar;
  n = aux[[1]]; (*N, auxiliares*)
  p = aux[[3]]; (*P, producciones*)
  sol = {};
  For[k = 1, k ≤ Length[p], k++,

   pright = p[[k]][[2]]; (*Parte derecha de la producción*)
   pleft = p[[k]][[1]]; (*Parte izquierda de la producción*)

   For[j = 1, j ≤ Length[pright], j++,
    rightlist = pright[[j]]; (*Una lista de la parte derecha*)

    (*Comprueba si el símbolo auxiliar de la
     parte izquierda pertenece a la lista de la parte derecha*)
    If[Intersection[rightlist, pleft] == pleft, AppendTo[sol, pleft]];

   ];
  ];
  sol = Union[Flatten[sol]];
  Return[sol];
 ]
```

```
Ejercicio2[{{S, U, V, T}, {a, b}, {{{S}, {{U}, {V}}}, {{U}, {{a, a, b}, {a, a, b}}},
   {{V}, {{T, b, V}, {T, b, T}}}, {{T}, {{a, T, b, T}, {b, T, a, T}, {{}}}}}, S}]
```

```
{T, V}
```

```
(*Ejercicio 3*)
Ejercicio3[grammar_] :=
 Module[{aux, n, p, t, pright, pleft, rightlist, i, j, k, sol },
  aux = grammar;
  n = aux[[1]]; (*N, auxiliares*)
  t = aux[[2]]; (*T, terminales*)
  p = aux[[3]]; (*P, producciones*)
  sol = True;
  For[k = 1, k ≤ Length[p], k++,

   pright = p[[k]][[2]]; (*Parte derecha de la producción*)
   pleft = p[[k]][[1]]; (*Parte izquierda de la producción*)

   For [j = 1, j ≤ Length[pright], j++,

    rightlist = pright[[j]]; (*Una lista de la parte derecha*)

    (*Comprueba que el tamaño de la lista de
     la parte derecha respeta la forma normal de Chomsky*)
    If[Length[rightlist] > 2 || Length[rightlist] < 1, sol = False];
    (*Comprueba si la lista de la parte
     derecha de longitud 2 contiene un símbolo terminal*)
    If[Length[rightlist] == 2, If[Intersection[rightlist, t] ≠ {}, sol = False]];
    (*Comprueba si la lista de la parte
     derecha de longitud 1 contiene un símbolo auxiliar*)
    If[Length[rightlist] == 1, If[Intersection[rightlist, n] ≠ {}, sol = False]];

    If[sol == False, Break[]];
   ];
   If[sol == False, Break[]];
  ];
  Return[sol];
 ]


Ejercicio3[{{S, A, B, C}, {a, b}, {{{S}, {{A, B}, {B, C}}},
   {{A}, {{B, A}, {a}}}, {{B}, {{C, C}, {b}}}, {{C}, {{a}, {A, B}}}}, S}]
True

(*Ejercicio 4*)
Ejercicio4[grammar_] :=
 Module[{p, N, t, aux, pright, pleft, i, m, n, production, extras},
  N = grammar[[1]]; (*N, auxiliares*)
  t = grammar[[2]]; (*T, terminales*)
  p = grammar[[3]]; (*P, producciones*)
  aux = grammar;
  extras = {};
```

```
aux[[3]] = {}; (*Borrando las producciones
 de aux dado que en el resultado serán distintas*)
For[i = 1, i ≤ Length[p], i++,
 pright = p[[i]][[2]];
 pleft = p[[i]][[1]];
 AppendTo[aux[[3]], {pleft, {}}];
 n = 2; (*Indice para apuntar a
  la segunda posicion de los simbolos consecuentes*)
 For[production = 1, production <= Length[pright], production++,
  n = 2;
  If[Length[pright[[production]]] == 1 ,
   AppendTo[aux[[3]][[i]][[2]], {pright[[production]][[1]]}];
   Continue[];
  ];
  If[Length[pright[[production]]] == 2  && MemberQ[t,
     pright[[production]][[1]]] && MemberQ[N, pright[[production]][[2]]],
   AppendTo[aux[[3]][[i]][[2]], {pright[[production]]}];
   Continue[];
  ];
  For[m = 1, m < Length[pright[[production]]], m++,
   (* m apunta a la primera posición de los simbolos consecuentes *)
   If[MemberQ[t, pright[[production]][[n]]],
    (* Comprueba si el siguiente simbolo consecuente es terminal *)
    If[m == 1, (* Comprobamos si estamos en el
      primer simbolo consecuente *)
     (*True*)
     AppendTo[aux[[3]][[i]][[2]],
      {pright[[production]][[m]], Subscript[X, i, production, m + 1]}];
     AppendTo[aux[[1]], Subscript[X, i, production, m + 1]];,


     (*False*)
     AppendTo[extras, {{Subscript[X, i, production, m]},
       {{pright[[production]][[m]], Subscript[X, i, production, m + 1]}}}];
     AppendTo[aux[[1]], Subscript[X, i, production, m + 1]];
    ];
    If[n == Length[pright[[production]]],
     (* Comprobamos si el siguiente simbolo consecuente es el último *)
     AppendTo[extras, {{Subscript[X, i, production, m + 1]},
        {{pright[[production]][[n]]}}}];
    ];,
    If[m == 1,
      (* Comprobamos si estamos en el primer simbolo consecuente *)
      (*True*)
     AppendTo[aux[[3]][[i]][[2]],
        {pright[[production]][[m]], pright[[production]][[n]]}];,
      (*False*)
```

```
        AppendTo[extras, {{Subscript[X, i, production, m]},
           {{pright[[production]][[m]], pright[[production]][[n]]}}}];
       ];
     ];
     n++;
    ];
   ];
  ];
  AppendTo[aux[[3]], extras];
  Return[aux];
 ]
```

```
grammar =
  {{S, A, B, C}, {a, b, c, d, e, f}, {{{S}, {{a, A}, {b, b}}}}, {{C}, {{b}}}, S};
Ejercicio4[grammar]
```

```
{{S, A, B, C, X_{1,2,2}}, {a, b, c, d, e, f},
 {{{S}, {{{a, A}}, {b, X_{1,2,2}}}}, {{{X_{1,2,2}}, {{b}}}}}, {{C}, {{b}}}, S}
```

```
grammar = {{S, A, B, C}, {a, b, c, d, e, f, g},
  {{{S}, {{a, a}, {b, b}}}, {{A}, {{B}, {c, c}}}, {{C}, {{A, b}, {b, A}}}}, S}
Ejercicio4[grammar]
```

```
{{S, A, B, C}, {a, b, c, d, e, f, g},
 {{{S}, {{a, a}, {b, b}}}, {{A}, {{B}, {c, c}}}, {{C}, {{A, b}, {b, A}}}}, S}
```

```
{{S, A, B, C, X_{1,1,2}, X_{1,2,2}, X_{2,2,2}, X_{3,1,2}}, {a, b, c, d, e, f, g},
 {{{S}, {{a, X_{1,1,2}}, {b, X_{1,2,2}}}}, {{A}, {{c, X_{2,2,2}}}}, {{C}, {{A, X_{3,1,2}}, {{b, A}}}},
  {{{X_{1,1,2}}, {{a}}}, {{X_{1,2,2}}, {{b}}}, {{X_{2,2,2}}, {{c}}}, {{X_{3,1,2}}, {{b}}}}}, S}
```