```
In[•]:= Ejemplo1 = {{S, A, B}, {a, b}, {{{S}, {{a, A}, {b, B, A}, {a, b}}},
          {{A}, {{a, A}, {B, B, A}}}, {{B}, {{{}}, {S, a}, {B, S}}}}, S}

Out[•]= {{S, A, B}, {a, b}, {{{S}, {{a, A}, {b, B, A}, {a, b}}},
          {{A}, {{a, A}, {B, B, A}}}, {{B}, {{{}}, {S, a}, {B, S}}}}, S}

In[•]:= Ejemplo2 = {{S, A, B, C}, {a, b}, {{{S}, {{B}, {A}, {C}, {S, A}}},
                                  constante                 constante
          {{A}, {{S}, {a, b, b, a}}}, {{B}, {{A, b, B, C}, {B, B, A}, {B, S, B}}},
                                                    constante
          {{C}, {{C, S}, {S, C, S, A}, {A, A, b, B, C, B}}}}, S}
            cons···  constante  constante          constante

Out[•]= {{S, A, B, C}, {a, b}, {{{S}, {{B}, {A}, {C}, {S, A}}},
          {{A}, {{S}, {a, b, b, a}}}, {{B}, {{A, b, B, C}, {B, B, A}, {B, S, B}}},
          {{C}, {{C, S}, {S, C, S, A}, {A, A, b, B, C, B}}}}, S}

In[•]:= Ejemplo3 = {{S, A, B, C}, {a, b},
                    constante
          {{{S}, {{C, B}, {A, B}}}, {{A}, {{a}}}, {{B}, {{b}}}, {{C}, {{A, S}}}}, S}
                   constante                                    constante

Out[•]= {{S, A, B, C}, {a, b},
          {{{S}, {{C, B}, {A, B}}}, {{A}, {{a}}}, {{B}, {{b}}}, {{C}, {{A, S}}}}, S}

In[•]:= Ejemplo4 = {{S, A, B, C, D, E, F}, {a, b}, {{{S}, {{C, A}, {D, B}, {a}, {b}, {E, F}}},
                           ···  ···  número e          consta···  deriva          número e
          {{A}, {{a}}}, {{B}, {{b}}}, {{C}, {{A, S}}}, {{D}, {{B, S}}},
                                              constante          deriva
          {{E}, {{F, F}, {F, E}}}, {{F}, {{F, F}, {F, E}}}}, S}
            número e          número e                número e

Out[•]= {{S, A, B, C, D, e, F}, {a, b},
          {{{S}, {{C, A}, {D, B}, {a}, {b}, {e, F}}}, {{A}, {{a}}}, {{B}, {{b}}},
          {{C}, {{A, S}}}, {{D}, {{B, S}}}, {{e}, {{F, F}, {F, e}}}, {{F}, {{F, F}, {F, e}}}}, S}

In[•]:= (*Entregable de Ignacio Diago Valeta*)
```

In[•]:=
```
directamenteGenerativos[gram_] :=
 Module[{sol, i, j, k, prod, aux, compruebaConsecuente},
 módulo
 sol = {};
 aux = gram[[1]];
 prod = gram[[3]];
 For[j = 1, j ≤ Length[prod], j++,
 para cada  longitud
  For[i = 1, i ≤ Length[prod[[j, 2]]], i++,
  para cada  longitud
   compruebaConsecuente = True;
                          verdadero
   For[k = 1, k ≤ Length[aux], k++,
   para cada  longitud
    If[MemberQ[prod[[j, 2, i]], aux[[k]]],
    si  ¿contenido en?
     compruebaConsecuente = False;
                            falso
     Break[]
     finaliza iteración
    ];
   ];
   If[compruebaConsecuente === True,
   si  verdadero
    AppendTo[sol, prod[[j, 1, 1]]];
    añade al final
    Break[]
    finaliza iteración
   ];
  ];
 ];
 Return[Union[Flatten[sol]]]
 retorna  unión  aplana
]
```

In[•]:=
```
directamenteGenerativos[Ejemplo1]
```

Out[•]= {B, S}

In[•]:=
```
directamenteGenerativos[Ejemplo2]
```

Out[•]= {A}

In[•]:=
```
directamenteGenerativos[Ejemplo3]
```

Out[•]= {A, B}

In[•]:=
```
directamenteGenerativos[Ejemplo4]
```

Out[•]= {A, B, S}

In[•]:=
```
(*Entregable de Ignacio Diago Valeta*)
```

```
directamenteNoGenerativos[gram_] :=
 Module[{sol, i, j, prod, generativos, esNoGenerativo, noGenerativos},
 módulo
  sol = {};
  noGenerativos = {};
  generativos = directamenteGenerativos[gram];
  prod = gram[[3]];
  For[i = 1, i ≤ Length[prod], i++,
  para cada          longitud
   If[! MemberQ[generativos, prod[[i, 1, 1]]],
   si     ¿contenido en?
     esNoGenerativo = True;
                      verdadero
     For[j = 1, j ≤ Length[prod[[i, 2]]], j++,
     para cada          longitud
      If[! MemberQ[prod[[i, 2, j]], prod[[i, 1, 1]]],
      si     ¿contenido en?
        esNoGenerativo = False;
                         falso
        Break[]
        finaliza iteración
       ];
     ];
     If[esNoGenerativo === True,
     si                    verdadero
      AppendTo[noGenerativos, prod[[i, 1, 1]]]
      añade al final
    ];
  ];
  Return[DeleteDuplicates[noGenerativos]]
  retorna   elimina repeticiones
 ]
```

In[ ]:= `directamenteNoGenerativos[Ejemplo1]`

Out[ ]= {A}

In[ ]:= `directamenteNoGenerativos[Ejemplo2]`

Out[ ]= {B, C}

In[ ]:= `directamenteNoGenerativos[Ejemplo3]`

Out[ ]= { }

In[ ]:= `directamenteNoGenerativos[Ejemplo4]`

Out[ ]= {F}

In[ ]:= `(*Entregable de Ignacio Diago Valeta*)`

```
In[•]:= formaNormalChomsky[gram_, w_] := Module[{n, r, table, i, j, k, l, b, c, prods},
                                          módulo
 n = Length[w];
     longitud
 r = Length[gram[[1]]];
     longitud
 table = Table[{}, {n}, {n}];
         tabla
 prods = Flatten[Table[{prod[[1, 1]], #} & /@ prod[[2]], {prod, gram[[3]]}], 1];
         aplana    tabla
 For[i = 1, i ≤ n, i++,
 para cada
  table[[i, i]] = Flatten[Select[prods, #[[2]] == {w[[i]]} &][[All, 1]]]
                  aplana    selecciona                        todo
 ];
 For[l = 2, l ≤ n, l++,
 para cada
  For[i = 1, i ≤ n - l + 1, i++,
  para cada
   j = i + l - 1;
   table[[i, j]] = {};
   For[k = i, k ≤ j - 1, k++,
   para cada
    Do[Do[table[[i, j]] =
    r··· repite
          Union[table[[i, j]], Flatten[Select[prods, #[[2]] == {b, c} &][[All, 1]]],
          unión                aplana    selecciona                    todo
         {c, table[[k + 1, j]]}], {b, table[[i, k]]}]
   ];
  ];
 ];
 Return[MemberQ[table[[1, n]], gram[[4]]]]
 retorna  ¿contenido en?
]

In[•]:= formaNormalChomsky[Ejemplo3, {a, a, b, b}]
Out[•]= True

In[•]:= formaNormalChomsky[Ejemplo3, {a, a, a, b}]
Out[•]= False

 formaNormalChomsky[Ejemplo3, {a, b, a, b}]
Out[•]= False

In[•]:= formaNormalChomsky[Ejemplo4, {a, b, a, b, a}]
Out[•]= True

In[•]:= formaNormalChomsky[Ejemplo4, {a, a, a, b}]
Out[•]= False
```

In[ ]:= **formaNormalChomsky[Ejemplo4, {b, b, b, a, b, b, b}]**

Out[ ]= True

7 lines of corrupt data deleted.