BNF Extendido

Ejemplo

Gramática Para convertir

```
G \rightarrow Graph (name, nodes(Verts); arcs(Arcs))
Verts \rightarrow Verts . Vert
Verts→ Vert
Vert \rightarrow Id
Vert → Id: Props
Arcs \rightarrow Arcs, Arc
Arcs \rightarrow Arc
Arc \rightarrow (Id, Id)
Arc \rightarrow (Id, Id, value)
Id \rightarrow num
Id \rightarrow name
value \rightarrow num
value \rightarrow - num
Props → Props , Prop
Props → Prop
Prop \rightarrow < name, value>
```

Unimos reglas con el mismo lado izquierdo

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Verts , Vert | Vert
    Vert → Id | Id : Props
    Arcs → Arcs , Arc | Arc
    Arc → (Id , Id) | (Id, Id, value)
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Ley de Arden a (1)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id | Id : Props
    Arcs → Arcs , Arc | Arc
    Arc → (Id , Id) | (Id , Id , value)
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Factorizamos Id en (3)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id (λ | : Props)
    Arcs → Arcs , Arc | Arc
    Arc → (Id , Id) | (Id, Id, value)
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

$(\lambda \mid \beta) \rightarrow [\beta] \text{ en } (3)$

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [: Props]
    Arcs → Arcs , Arc | Arc
    Arc → (Id , Id) | (Id, Id, value)
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Ley de Arden a (4)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [: Props]
    Arcs → Arc {, Arc }
    Arc → (Id , Id) | (Id, Id, value)
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Factorizar (Id, Id en (5)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [: Props]
    Arcs → Arc {, Arc }
    Arc → (Id , Id () | , value))
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Factorizar) en (5)

```
G 	o Graph (name, nodes(Verts); arcs(Arcs))

Verts 	o Vert {, Vert }

Vert 	o Id [Id : Props]

Arcs 	o Arc {, Arc }

Arc 	o (Id , Id (\lambda |, value))

Id 	o num | name

value 	o num | - num

Props 	o Props , Prop | Prop

Prop 	o < name, value >
```

$(\lambda \mid \beta) \rightarrow [\beta] \text{ en } (5)$

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [Id: Props]
    Arcs → Arc {, Arc }
    Arc → (Id, Id[, value])
    Id → num | name
    value → num | - num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Saltandónos pasos – es opcional en (7)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [Id: Props]
    Arcs → Arc {, Arc }
    Arc → (Id, Id[, value])
    Id → num | name
    value → [-]num
    Props → Props , Prop | Prop
    Prop → < name, value>
```

Ley de Arden en (8)

```
    G → Graph (name, nodes(Verts); arcs(Arcs))
    Verts → Vert {, Vert }
    Vert → Id [Id: Props]
    Arcs → Arc {, Arc }
    Arc → (Id, Id[, value])
    Id → num | name
    value → [-] num
    Props → Prop {, Prop}
    Prop → < name, value>
```