

Grammar:

```
1 Goal → List
2 List → List Pair
3       | Pair
4 Pair → ( Pair )
5       | ( )
```

First sets:

```
First( Goal ) = { ( }
First( List ) = { ( }
First( Pair ) = { ( }
```

Closure procedure:

```
closure(s):
while (s is changing)
  for each item [A→β•Cδ, a] in s
    for each production C → γ
      for each b in First(δa)
        s ← s ∪ {[C → •γ, b]}
return s
```

S = {[Goal -> • List, eof]}