Universidad de los Andes
Facultad de Ingeniería

Ingeniería de Sistemas y Computación
ISIS-1106 Lenguajes y Máquinas
2024-20
Practical Quiz
Individual

We can specify Petri net with the language defined below.

- A Petri net specification begins with keyword `PetriNet` followed by an identifier and ends with the keyword `END`.
- Between `PetriNet` name and `END` we have the following:
  - Place specifications: The keyword `PLACES` followed place specifications separated by commas. A place specification is an identifier that may be followed by its capacity within parenthesis. The capacity is a number.
  - Transition specifications: The keyword `TRANSITIONS` followed by transition specifications separated by semicolons. A transition specification is an identifier followed by the inputs and the outputs separated by a comma and within parenthesis,
    - Both the inputs and the outputs are sets of place names: sequences of zero or more places separated by commas withing {}'s. Place names may be followed by the weight of the transition within parenthesis. The weight is a number.

Terminals: `PetriNet`, `End`, `PLACES`, `TRANSITIONS`, ",", ";", "(", ")",, "{", "}", ids.

An identifier is a sequence of alphanumeric characters that begin with a letter.

A number is a sequence of digits:

TASK:

Define the grammar for Petri net specifications; implement a JavaCC yes/no parser; and integrate it to the attached parserTester project. You must verify that the elements in the sets in transition specifications are in fact place names defined previously.

EXAMPLE

```
PetriNet myNet
Places
  Waiting(100), ReadyB,  ReadyA, WorkingA, WorkingB,
  CountA(100),CountB(100)
Transitions
   Arrive({},{Waiting});
   StartA({Waiting, ReadyA},{WorkingA});
  FinishA({WorkingA},{ReadyA, CountA(2)});
  StartB({Waiting(2), ReadyB},{WorkingB});
FinishB({WorkingB(2),{ReadyB, CountB(2)})
End
```