```
MODULE GFXSpec
```

The EXTENDS statement imports the standard module *FiniteSets*, which defines a few useful operators for reasoning about finite sets, including *Cardinality*.

```
EXTENDS FiniteSets
```

The CONSTANT statement declares *Proc* to be an unspecified constant. There's no need (and no way) to specify that *Proc* is a set because TLA+ is based on *ZF* set theory, so value is a set.

```
CONSTANT Proc
```

```
(***************************************************************************
--algorithm GFXSpec
{ variable result = [p ∈ Proc ↦ {}]
  process (Pr ∈ Proc)
    {A: with (P ∈ {Q ∈ SUBSET Proc :
                          ∧ self ∈ Q
                          ∧ ∀ p ∈ Proc \ {self} :
                              ∨ Cardinality(result[p]) ≠ Cardinality(Q)
                              ∨ Q = result[p]
                  })
            {result[self] := P}
    }
}
***************************************************************************)
```

The algorithm is automatically translated to the stuff between the BEGIN TRANSLATION and END TRANSLATION comments.

```
BEGIN TRANSLATION
VARIABLES result, pc

vars ≜ ⟨result, pc⟩

ProcSet ≜ (Proc)

Init ≜ Global variables
        ∧ result = [p ∈ Proc ↦ {}]
        ∧ pc = [self ∈ ProcSet ↦ "A"]

A(self) ≜ ∧ pc[self] = "A"
          ∧ ∃ P ∈ {Q ∈ SUBSET Proc :
                          ∧ self ∈ Q
                          ∧ ∀ p ∈ Proc \ {self} :
                              ∨ Cardinality(result[p]) ≠ Cardinality(Q)
                              ∨ Q = result[p]
                  } :
              result' = [result EXCEPT ![self] = P]
          ∧ pc' = [pc EXCEPT ![self] = "Done"]
```

$Pr(self) \triangleq A(self)$

$Next \triangleq (\exists\, self \in Proc : Pr(self))$
$\quad\quad \lor$ Disjunct to prevent deadlock on termination
$\quad\quad\quad ((\forall\, self \in ProcSet : pc[self] = \text{“Done”}) \land \text{UNCHANGED } vars)$

$Spec \triangleq Init \land \Box[Next]_{vars}$

$Termination \triangleq \Diamond(\forall\, self \in ProcSet : pc[self] = \text{“Done”})$

END TRANSLATION

We can check the specification for trivial “type errors” with *TLC* by having it check that the following predicates *TypeOK* and *GFXCorrect* are invariants.

$TypeOK \triangleq result \in [Proc \rightarrow \text{SUBSET } Proc]$

$Done(i) \triangleq pc[i] = \text{“Done”}$

$GFXCorrect \triangleq \forall\, i, j \in Proc :$
$\quad \land Done(i) \land Done(j)$
$\quad \land Cardinality(result[i]) = Cardinality(result[j])$
$\quad \Rightarrow (result[i] = result[j])$

---

* Modification History
* Last modified *Fri Jul* 20 14:07:01 *PDT* 2012 by *lamport*
* Created *Fri Jul* 06 03:18:28 *PDT* 2012 by *lamport*