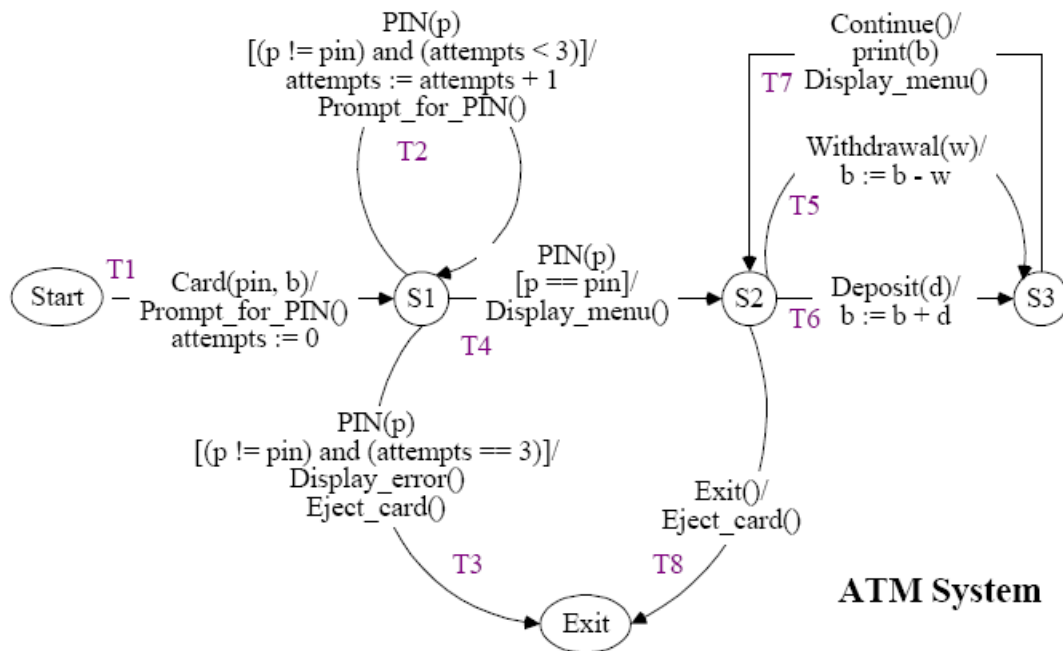


```

<mod> ::= <mtransitions>
<mtransitions> ::=
    <mtransition> | <mtransitions> <mtransition>
<mtransition> ::=
    mtransition mType transitionId
    sourceStateIndex destinationStateIndex
    <requirement>
<requirement> ::=
    [<input>]
    [<enablingPredicate>]
    /
    [<actions>]
<actions> ::=
    <action> | <actions> <action>
<action> ::=
    <output> | <assignment> | <set> | <reset> | <procedureCall>
<input> ::=
    inputId ( [<parameters>] )
<output> ::=
    outputId ( [<parameters>] )
<enablingPredicate> ::=
    <variableIds> [/* booleanExpression */]
<assignment> ::=
    <variableId> := <expression>
<set> ::=
    set ( constant , timerId )
<reset> ::=
    reset ( timerId )
procedureCall ::=
    procedure ( procedureId ( <variableIds> [; <variableIds>]
    ) ) { <pbrDefs> }
<parameters> ::=
    <parameter> { , <parameter> } *
<parameter> ::=
    <variableId> | constant
<variableIds> ::=
    <variableId> { , <variableId> } *
<pbrDefs> ::=
    <pbrDef> | <pbrDefs> <pbrDef>
<pbrDef> ::=
    <variableId> := <expression> ;
<expression> ::=
    function ( <variableIds> ) | constant
<variableId> ::=
    id

```



mtransition 0 T9

2 3

Balance(b) /

print(b)

mtransition 1 T6

2 3

Deposit(d) /

b := function(b, d)