**RuleR Tool Rule Systems**

1. **Open Close File**

ruler OpenCloseFile {

Always Open() {

open(file) -> isOpen(file);

close(file), !isOpen(file) -> Fail;

}

State isOpen(file) {

open(file) -> Fail;

close(file) -> Ok;

}

}

1. **State Machine**

ruler StateMachine {

Step state1() {

a -> state2();

b -> Fail;

}

Step state2() {

a -> Fail;

b -> state1();

}

}

1. **Auction**

ruler Auction {

Always Start() {

create(item, min) -> Created(item, min, 0);

}

State Created(item, min, max) {

bid(item, amount),amount <= max -> Fail;

bid(item, amount),amount > max -> Created(item, min, amount);

sell(item),max < min -> Fail;

sell(item),max >= min -> Sold(item);

}

State Sold(item) {

create(item, min) -> OK;

bid(item, amount) -> Fail;

sell(item) -> Fail;

}

}

1. **Interface**

ruler Interface {

Always Open() {

openwindow(windName) -> Opened(windName);

close(windName), !Close(windName) -> Fail;

loaddata(windName,data), !Opened(windName) -> Fail;

}

Forbidden State Opened(windName) {

loaddata(windName,data) -> Close(windName)

}

Forbidden State Close(windName) {

close(windName) -> OK;

}

}

1. **Lottery**

ruler Lottery {

Always Spin() {

getNumber(account,n), Compare(account,n),Take(account,a),Pay(account,a) -> Fail;

compare(account,m), !Compare(account,n) -> Fail;

pay(account,a), !Win(account,a) -> Fail;

take(account,a), !Lose(account,a) -> Fail;

getNumber(account,n) -> Compare(account,n)

}

State Compare(account,n) {

compare(account,m), m > n -> Win(account,m+1);

compare(account,m), m <= n -> Lose(account,n+1);

}

State Lose(account,amount) {

take(account, amount) -> OK;

}

State Win(account,amount) {

pay(account, amount) -> OK;

}

}