**A running example:**

**Slide 6:**

Algebra (state) Q:

Universe UQ = {1,2,3}, consisting of two nullary functions aQ and bQ, and two unary functions vQ and nextQ:

aQ = 1

bQ = 1

vQ(1) = 1

vQ(2) = 2

vQ(3) = 3

nextQ(1) = 2

nextQ(2) = 3

nextQ(3) = 1

The signature ∑Q consists of the function symbols a,b,v,next.

**Slide 14:**

A set of molecules:

Q = {(a,[],1), (b,[],2),(v,[1],1),(v,[2],2),(v,[3],3),(next,[1],2),

(next, [2],3),(next,[3],1)}

**Slide 15:**

MQ =def {(a,[],1), (v,[2],2)}

M'Q =def {(a,[],2), (v,[2],3)}

(MQ, M'Q) is a substep which changes the substate MQ to the substate M'Q

**Slide 16**

|  |
| --- |
| M'Q (a,[],2) (v,[2],3) |
| (b,[],2) (next,[1],2) E (v,[1],1) (next,[2],3)  (v,[3],3) (next,[3],1) |

Q Q'

|  |
| --- |
| MQ (a,[],1) (v,[2],2) |
| (b,[],2) (next,[1],2) E (v,[1],1) (next,[2],3)  (v,[3],3) (next,[3],1) |

**Slide 22**

An example built from the signature ∑Q, the assignment rule ExAssign:

v(a):=next(b).

Executing this rule at state Q assigns to the function symbol v at the argument aQ=1, the value next(b) Q=3.

The result is a new state Q', equal to Q except for the value at location (v,[1]).

**Slide 23**

The conditional rule ExCond:

If (a=b) then a:=next(a)

Executes the assignment rule at a state Q only if condition a=b holds in A.

**Slide 24**

The parallel rule ExPar:

par a:=b b:=a endpar

Executing ExPar at a state Q will result a new state Q' where the values of a and b are swapped.

**Slide 26**

The choice rule ExChoice:

Choose x,y with v(x)≠y do v(x):=y

The rule assigns the nondeterministically chosen value y to the function symbol v at the nondeterministically chosen argument x and ensures that the newly assigned value differs from the old one.