A

```
rs = @reaction_network begin
    (kB,kD), 2A <--> B
end kB kD
```

```
@parameters kB kD
@variables t A(t) B(t)

reactions = [Reaction(kB, [A], [B], [2], [1]),
                    Reaction(kD, [B], [A], [1], [2])]
@named rs  = ReactionSystem(reactions, t)
```

ReactionSystem

ps	states	eqs
kB	A(t)	kB, 2*A --> B
kD	B(t)	kD, B --> 2*A

B

```
os = convert(ODESystem,rs)
```

ODESystem

ps	states	eqs
kB	A(t)	Differential(t)(A(t)) ~ 2kD*B(t) - kB*(A(t)^2)
kD	B(t)	Differential(t)(B(t)) ~ (1//2)*kB*(A(t)^2) - kD*B(t)

C

```
u0 = [A => 1.0, B => 1.0]
p = [kD => 1.0, kB => 1.0]
tspan = (0.0,10.0)
```

```
oprob = ODEProblem(os,u0,tspan,p)
sol = solve(oprob)
```

```
function (__out, __arg1, __arg2, t)
  begin
      begin
          @inbounds begin
              __out[1] = (+)((*)((*)(-1//1, __arg2[1]), (^)((getindex)(__arg1, 1), 2)), (*)((*)(2, __arg2[2]), (getindex)(__arg1, 2)))
              __out[2] = (+)((*)((*)(1//2, __arg2[1]), (^)((getindex)(__arg1, 1), 2)), (*)((*)(-1, __arg2[2]), (getindex)(__arg1, 2)))
              nothing
          end
      end
  end
end
```

Solution

t

```
[0.0, 0.002,...,10.0]
```

u

```
[[1.0,1.0], [1.002, 0.998], ... [1.30, 0.84]]
```