

 **EECRIVONS UNDSLL**

SCRIPTS AND

```
let (-->) cmd1 cmd2 = Chain(cmd1, cmd2)
```

```
let _ = Face East  
      --> Start  
      --> Stop  
      --> Face South  
      --> Start  
      --> Stop
```


MAGNET 2-DUALS - TRANSLES

SCRIPT DE COMMANDE

ÉCRIVONS UN DSL



```
let (-->) cmd1 cmd2 = Chain(cmd1, cmd2)

let _ = Face East
      --> Start
      --> Stop
      --> Face South
      --> Start
      --> Stop
```

TAKE AWAY

NOUS AVONS VU

Le type `command * command` se lit **commande ET commande**, c'est un tuple, donc un type « ET »

Le type `command` est récursif : s'exprime en terme de **Chain of `command * command`**

Un système qui a des types « ET », des types « OU » et des types récursifs s'appelle un système de données algébriques (ADT)

Un ADT permet de représenter les valeurs autorisées d'un programme et leurs transitions

