

# **Obtenção de autômatos a partir de gramáticas em Notação de Wirth**

# Gramática Wirth do Dartmouth Basic

Program = BStatement { BStatement } int "END" .

BStatement = int ( Assign | Read | Data | Print | Goto | If  
| For | Next | Dim | Def | Gosub | Return | Remark ) .

Assign = "LET" Var "=" Exp .

Var = letter digit | letter [ "(" Exp { "," Exp } ")" ] .

Exp = { "+" | "-" } Eb { ( "+" | "-" | "\*" | "/" | "↑" ) Eb } .

Eb = "(" Exp ")" | Num | Var |

( "FN" letter | Predef ) "(" Exp ")" .

Predef = "SIN" | "COS" | "TAN" | "ATN" | "EXP" | "ABS"  
| "LOG" | "SQR" | "INT" | "RND" .

Read = "READ" Var { "," Var } .

Data = "DATA" Snum { "," Snum } .

Print = "PRINT" [ Pitem { "," Pitem } [ "," ] ] .

Pitem = Exp | "" Character { Character } "" [ Exp ] .

Goto = ( "GOTO" | "GO" "TO" ) int .

If = "IF" Exp ( ">=" | ">" | "<>" | "<" | "<=" | "=" ) Exp  
"THEN" int .

For = "FOR" letter [ digit ] "=" Exp "TO" Exp  
[ "STEP" Exp ] .

Next = "NEXT" letter [ digit ] .

Dim = "DIM" letter "(" int { "," int } ")"  
{ "," letter "(" int { "," int } ")" } .

Def = "DEF FN" letter "(" letter [ digit ] ")" "=" Exp .

Gosub = "GOSUB" int .

Return = "RETURN" .

Remark = "REM" { Character } .

Int = digit { digit } .

Num = ( Int [ "." { digit } ] | "." Int )  
[ "E" [ "+" | "-" ] Int ] .

Snum = [ "+" | "-" ] Num .

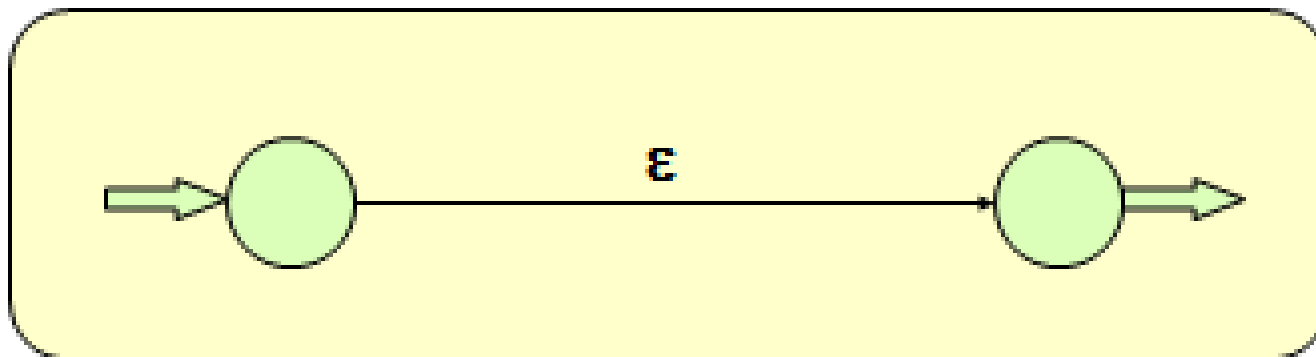
Character = letter | digit | special .

# REGRAS DE CONSTRUÇÃO DO AUTÔMATO

# 1 VAZIO

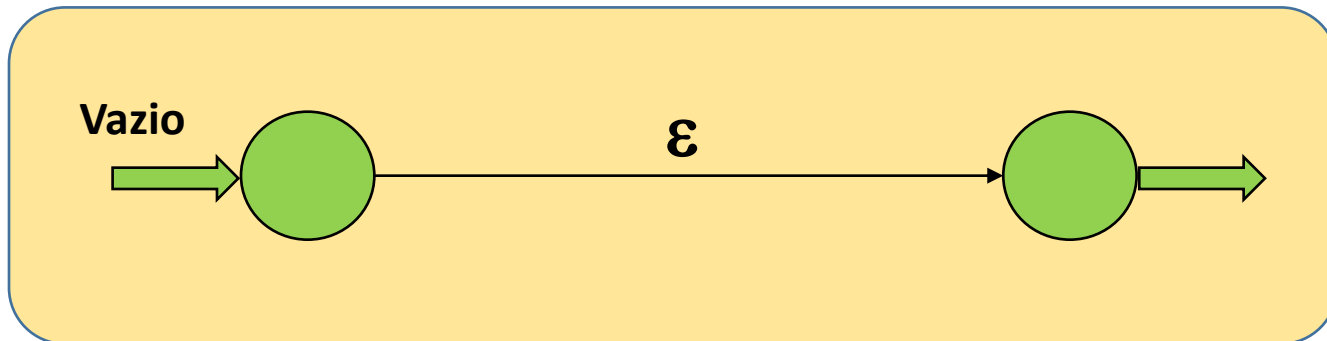
(transição interna, sem consumo de símbolo)

$\epsilon$



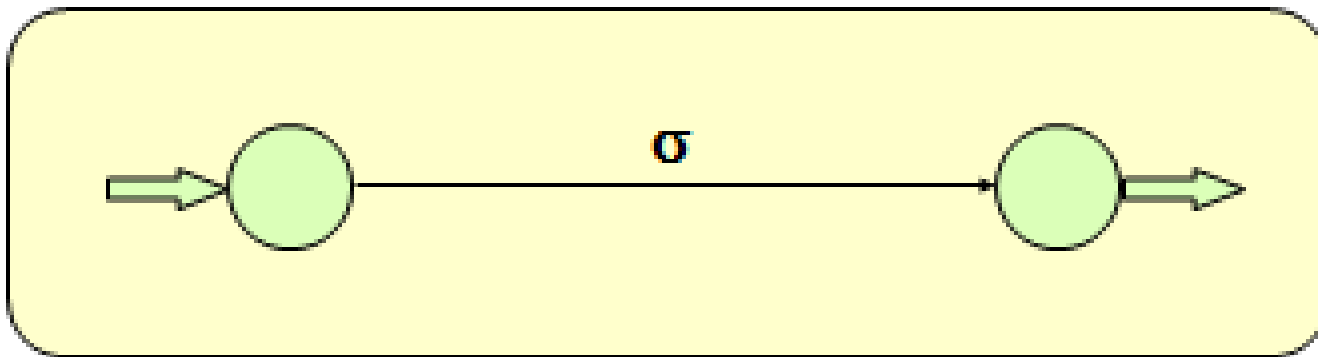
Vazio =  $\varepsilon$  .

$\varepsilon$



## 2 OCORRÊNCIAS DE TERMINAIS ISOLADOS (consumo de um símbolo do alfabeto)

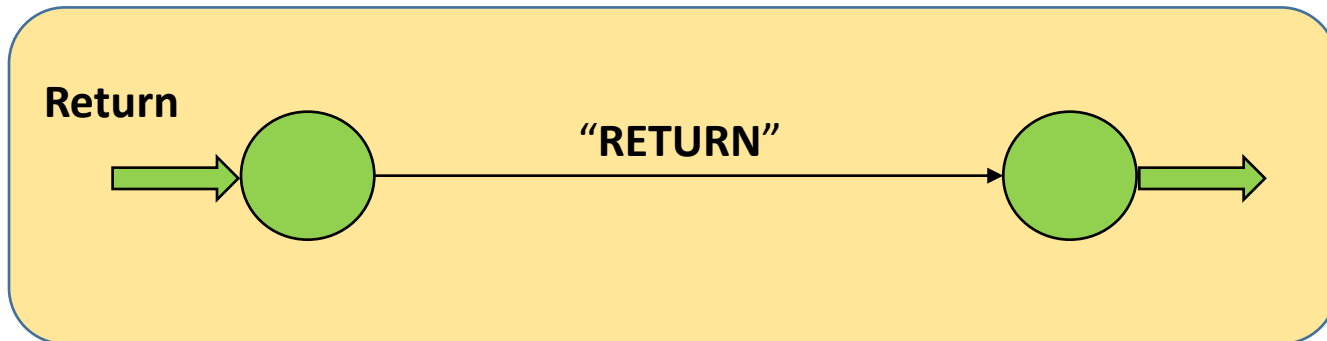
$\sigma$



# Exemplo

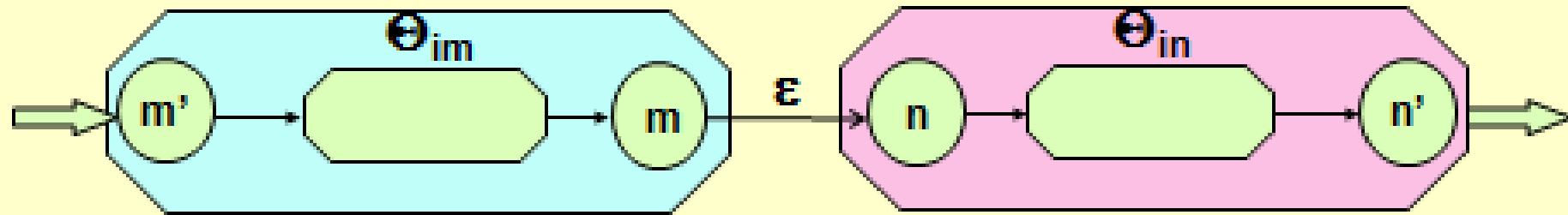
Return = **"RETURN"** .

**"RETURN"**



### 3 CONCATENAÇÃO DE EXPRESSÕES (concatenação de linguagens)

$\Theta_{im} \Theta_{in}$

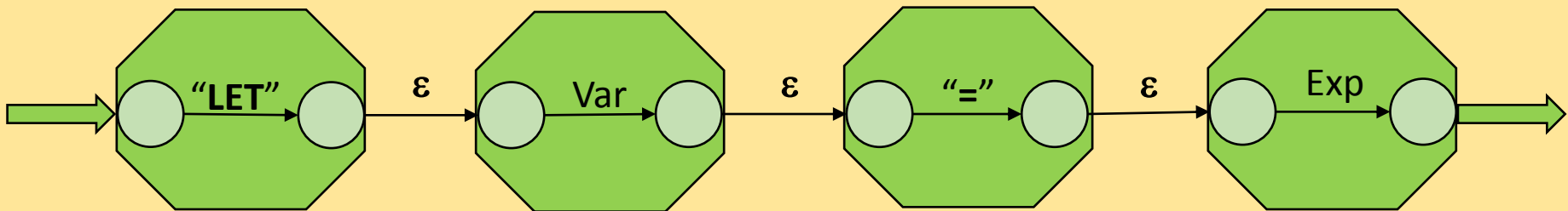


- Neste diagrama,  $m$  e  $n$  representam respectivamente os estados de saída de  $\Theta_{im}$  e de entrada de  $\Theta_{in}$
- Analogamente,  $m'$  e  $n'$  correspondem aos estados de entrada de  $\Theta_{im}$  e de saída de  $\Theta_{in}$ , respectivamente.



Assign = "LET" Var "=" Exp .

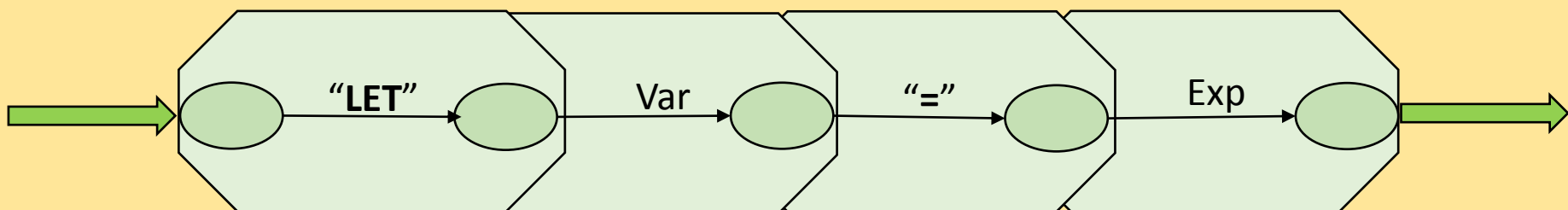
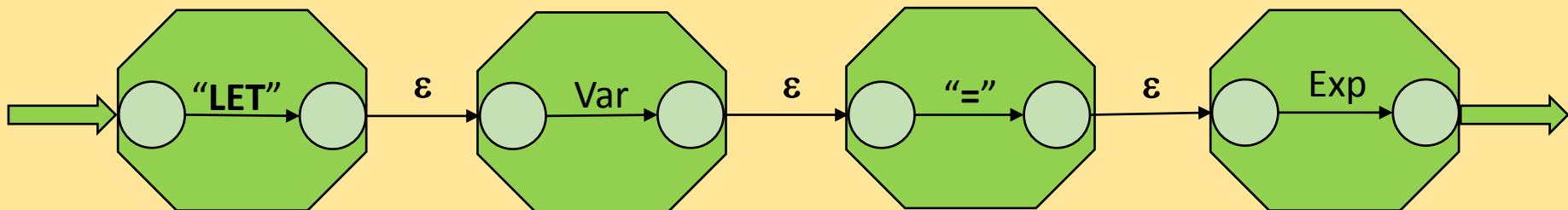
"LET" Var "=" Exp



# Uma simplificação possível

Em casos como este, em que sub-autômatos são concatenados por transições em vazio que ligam estados, dos quais não emergem outras transições, a estados nos quais se inicia o reconhecimento da estrutura sintática seguinte, não há impedimentos em remover a transição em vazio e fundir os estados que se conectam. Esta manobra deve ser feita com cuidado para que a linguagem reconhecida pelo autômato final não se altere por efeito de uma simplificação incorreta ou exagerada.

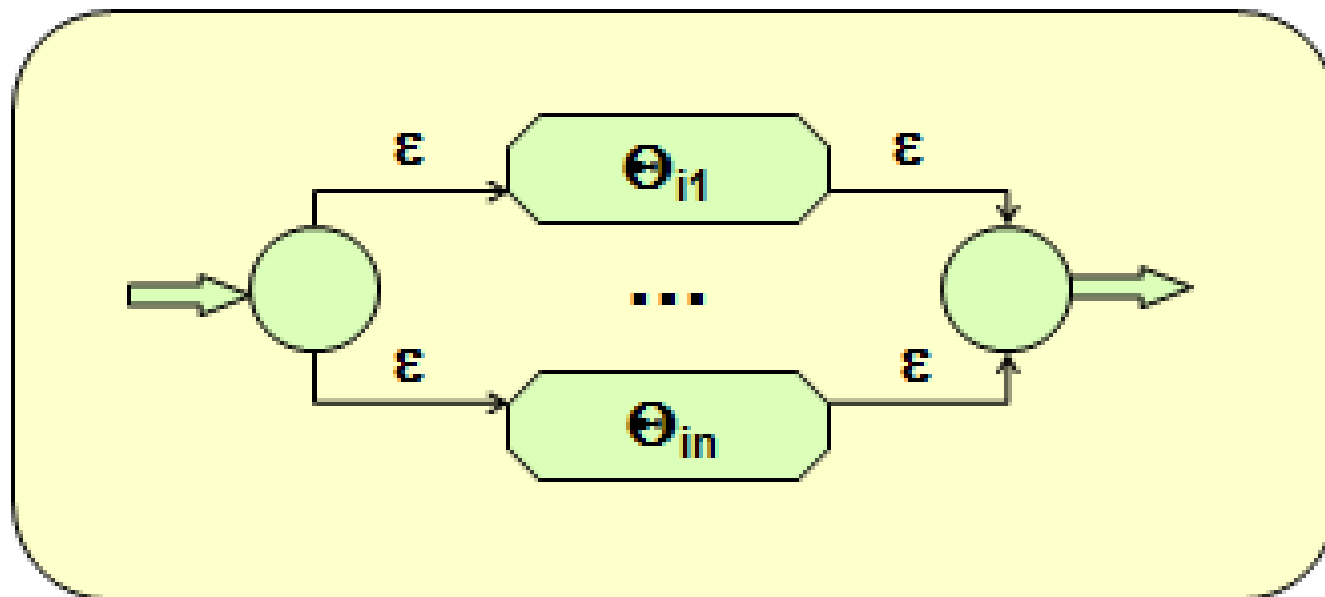
"LET" Var "=" Exp



# Interpretação da Notação de Wirth

## 4 AGRUPAMENTOS DE EXPRESSÕES ENTRE PARÊNTESES (instanciação obrigatória: uma e só uma instância)

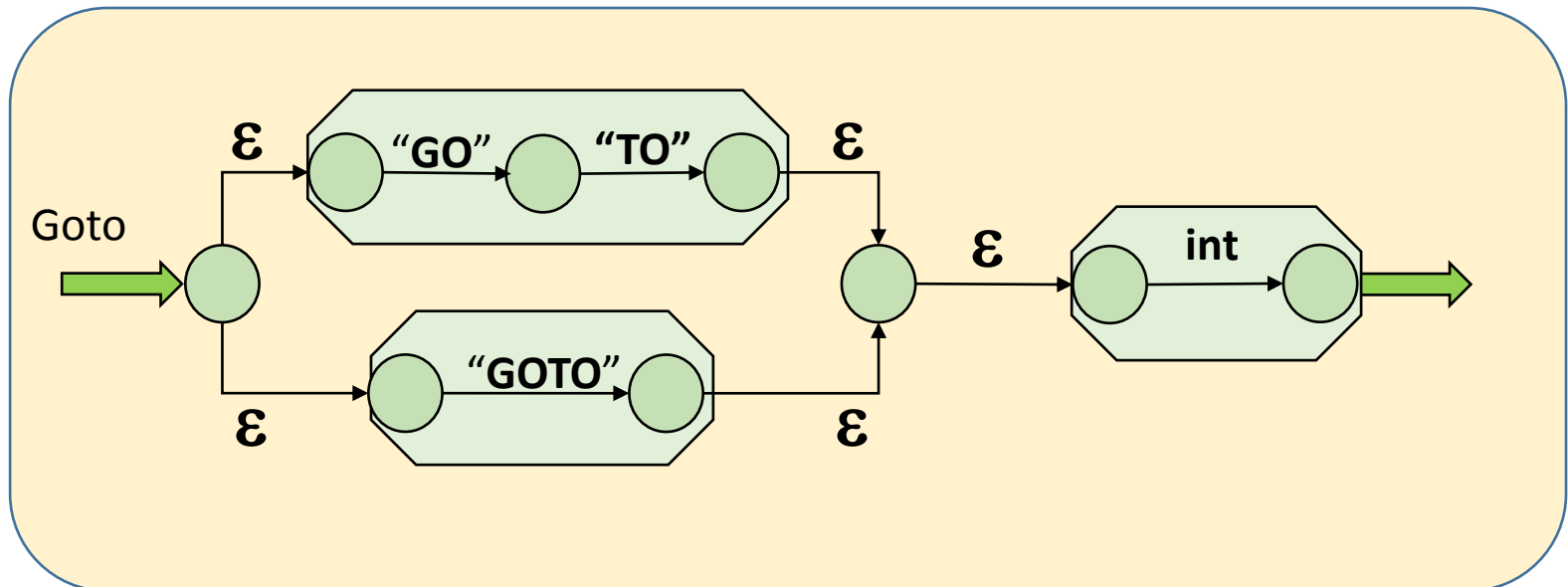
$( \Theta_{i1} \mid \Theta_{i2} \mid \dots \mid \Theta_{in} )$



# Exemplo

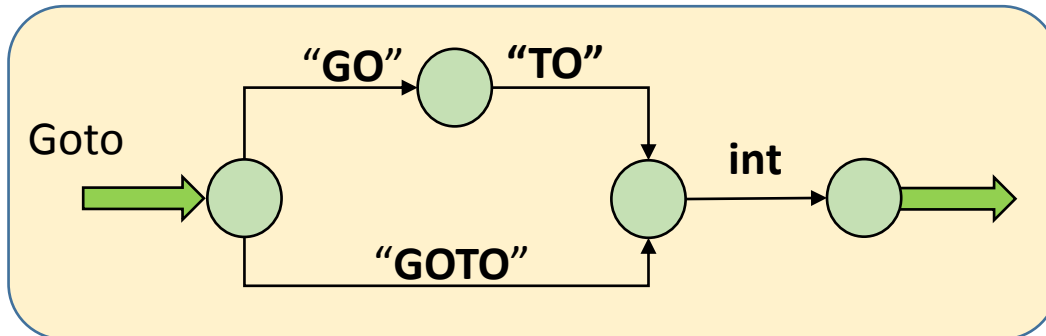
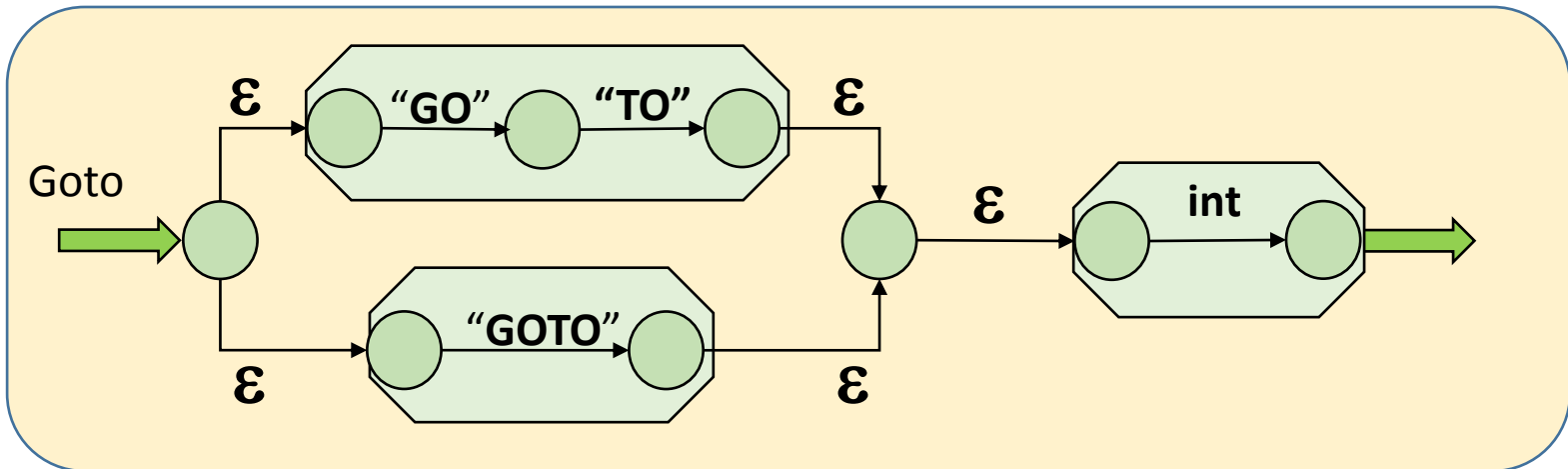
$\text{Goto} = ( \text{"GOTO"} \mid \text{"GO"} \text{"TO"} ) \text{int} .$

$( \text{"GOTO"} \mid \text{"GO"} \text{"TO"} ) \text{int}$



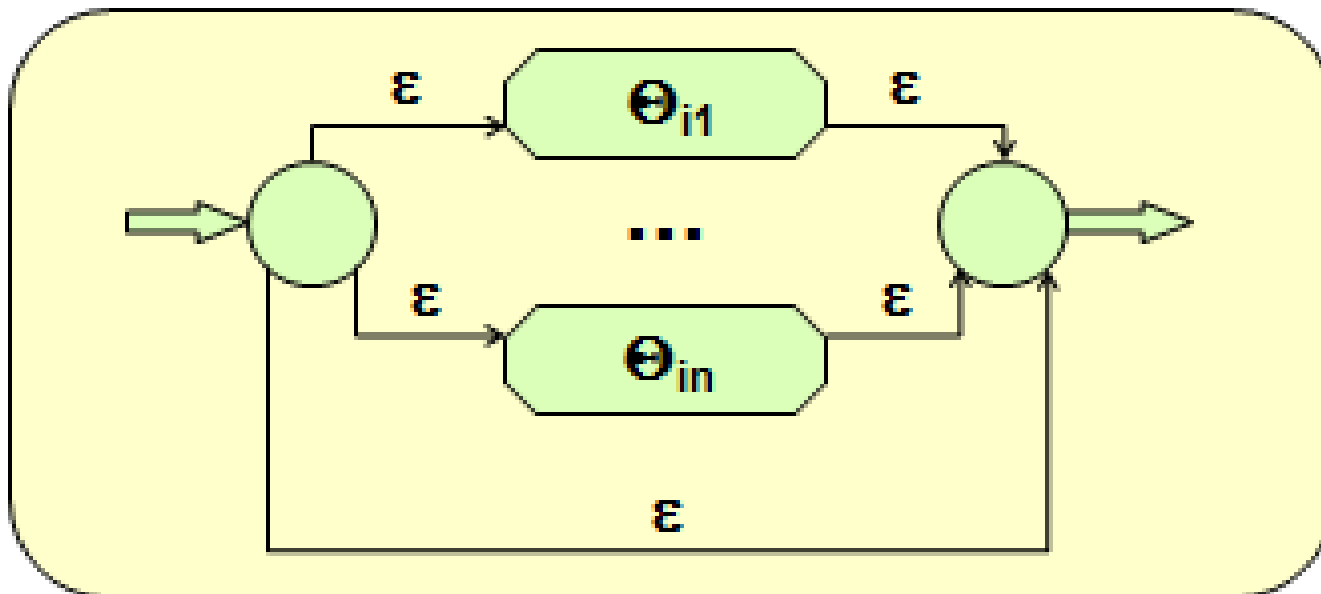
# Simplificando também neste caso

- Aqui também é possível fazer uma simplificação relativamente trivial, eliminando as transições em vazio do autômato, dado que não há nenhum impedimento:



## 5 AGRUPAMENTOS DE EXPRESSÕES ENTRE COLCHETES (opcional: zero ou uma instância)

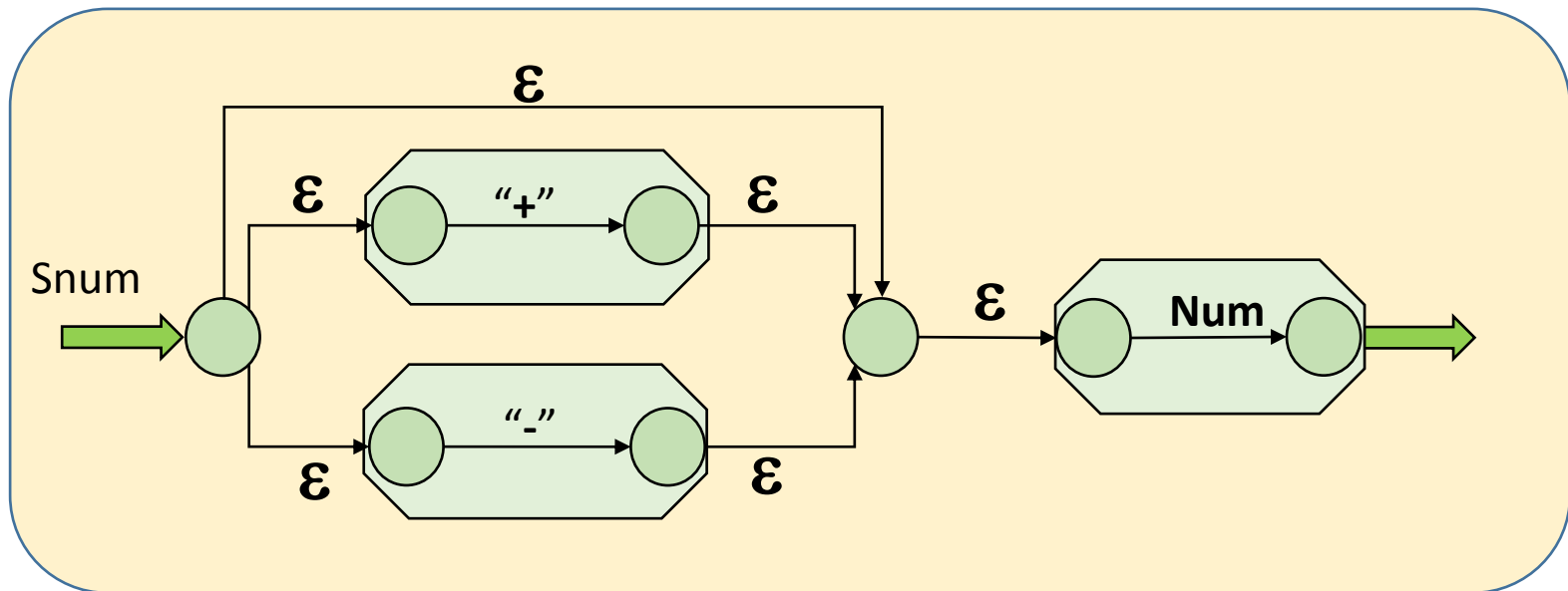
$[ \Theta_{i1} \mid \Theta_{i2} \mid \dots \mid \Theta_{in} ]$



# Exemplo

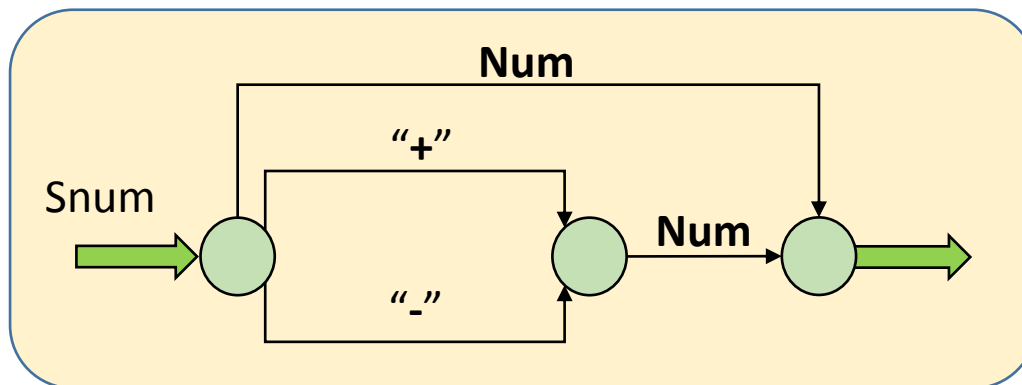
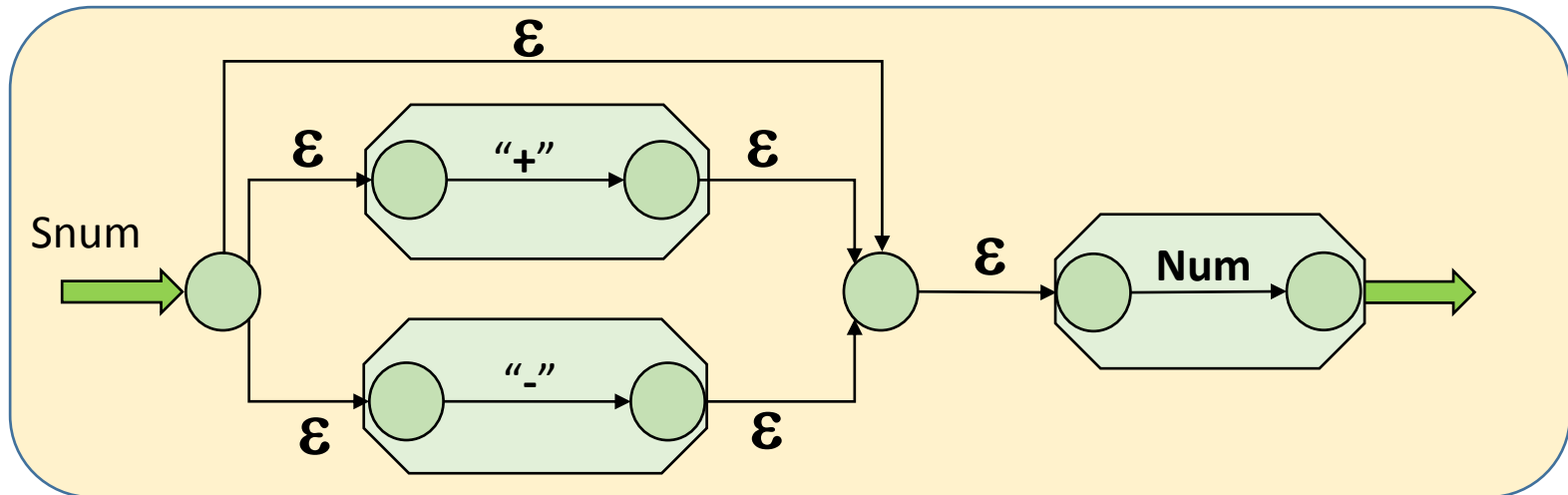
$S_{\text{num}} = [ "+" \mid "-" ] \text{Num} .$

$[ "+" \mid "-" ] \text{Num}$



# Simplificação

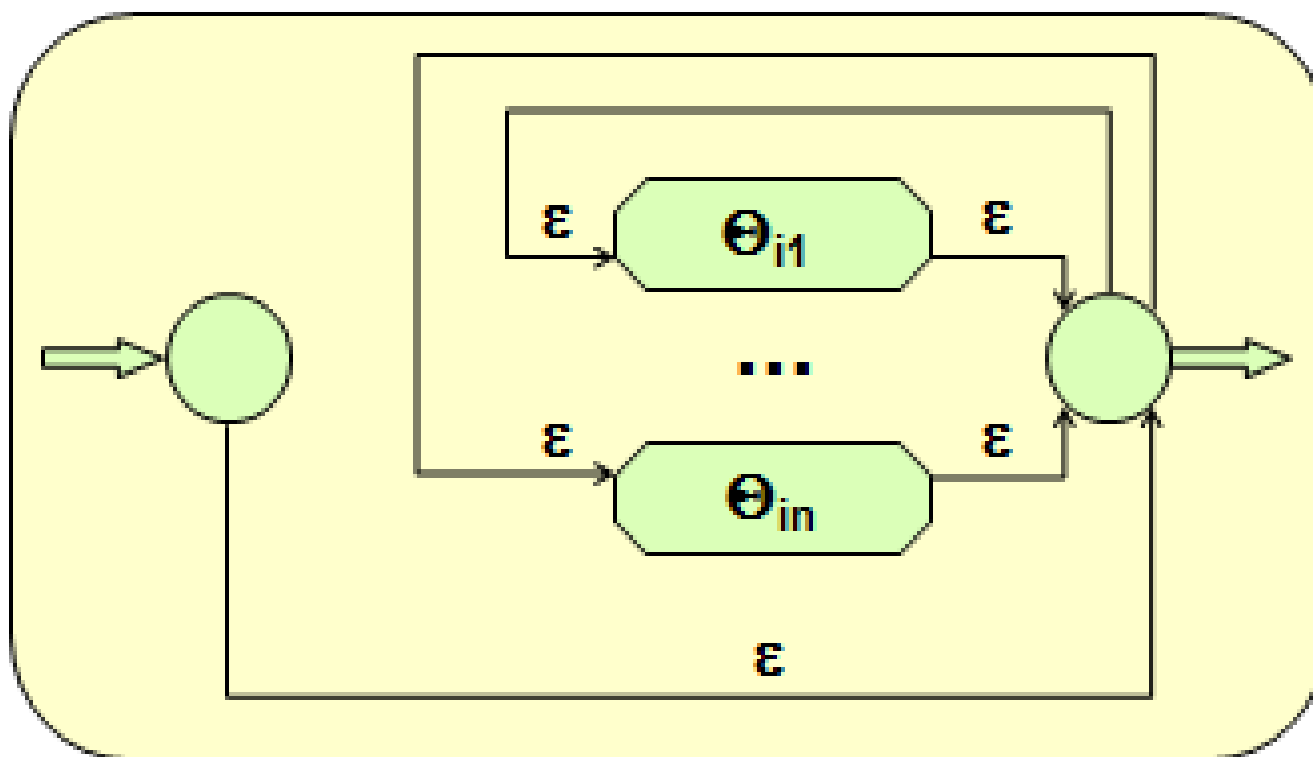
A simplificação neste caso é um pouco menos imediata, devido à transição em vazio responsável pela omissão do sinal. Notar a fusão com a transição com Num, a qual é antecipada para o primeiro estado.





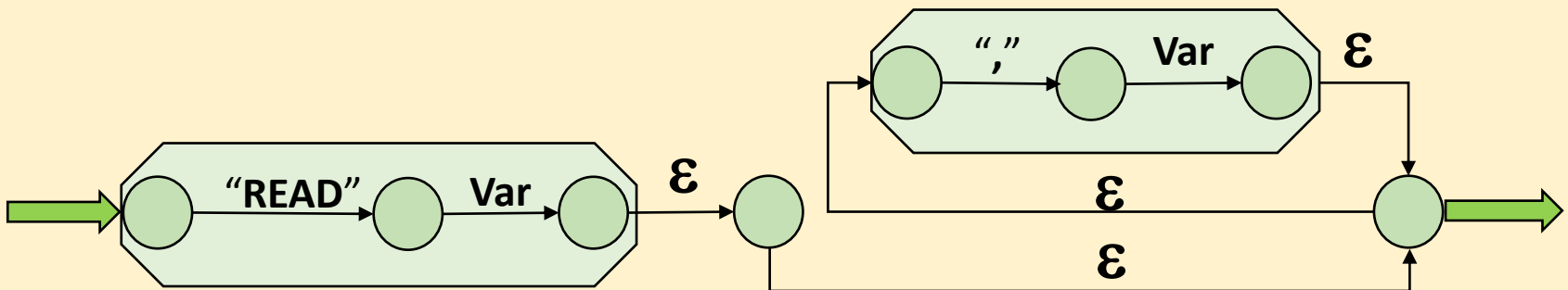
## 6 AGRUPAMENTOS DE EXPRESSÕES ENTRE CHAVES (fecho de Kleene: zero ou mais instâncias)

$\{ \Theta_{i1} \mid \Theta_{i2} \mid \dots \mid \Theta_{in} \}$



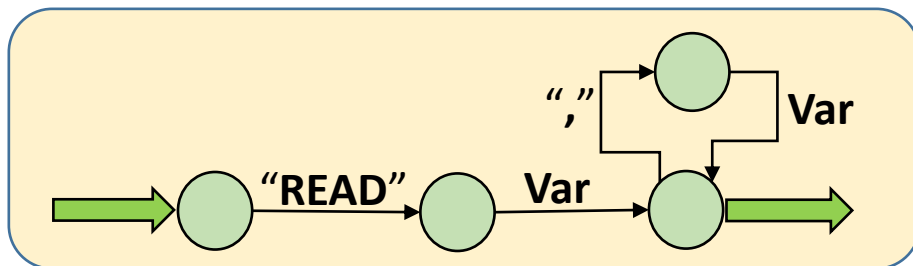
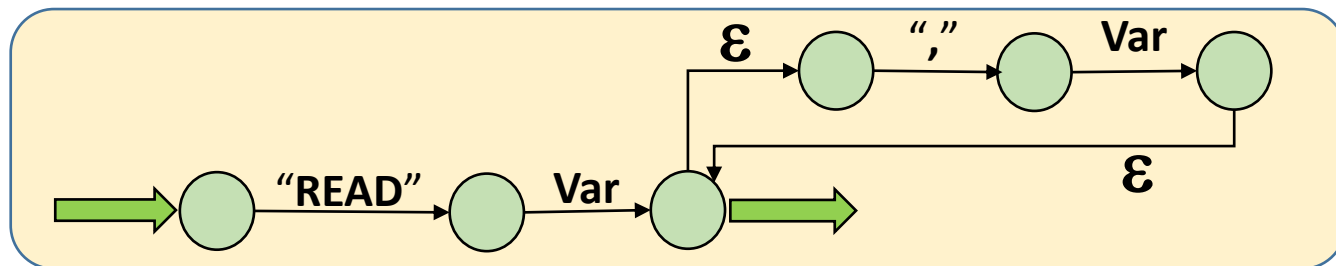
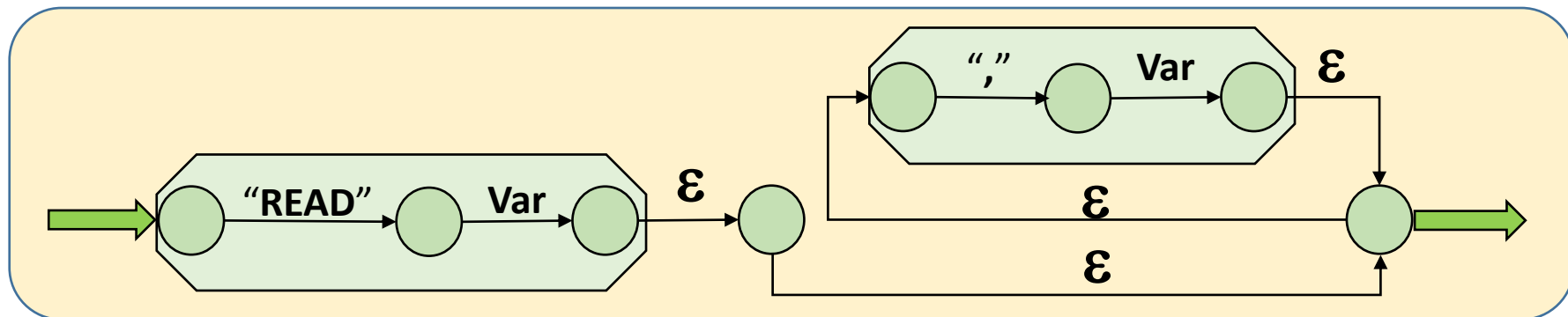
Read = "READ" Var { "," Var } .

"READ" Var { "," Var }



# Simplificação

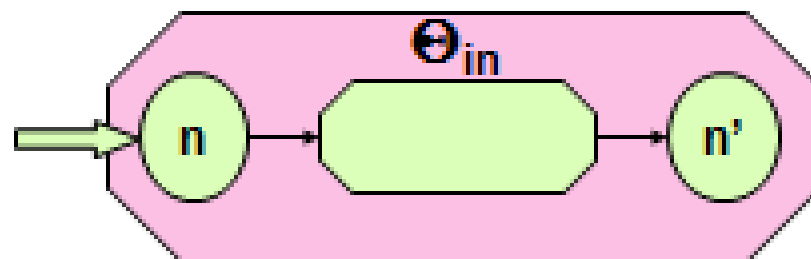
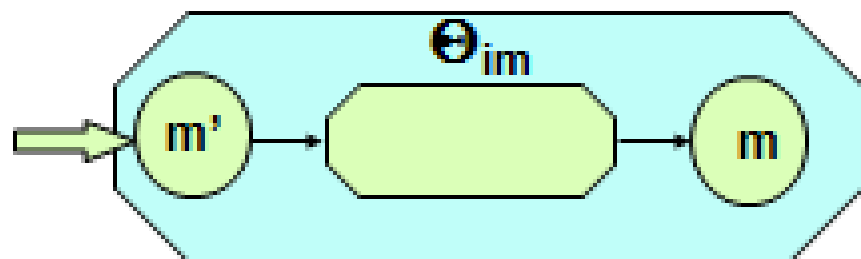
A simplificação neste caso é ligeiramente mais complexa, e por segurança pode ser feita por partes e com cuidado, eliminando primeiro as transições mais simples e depois as menos evidentes.



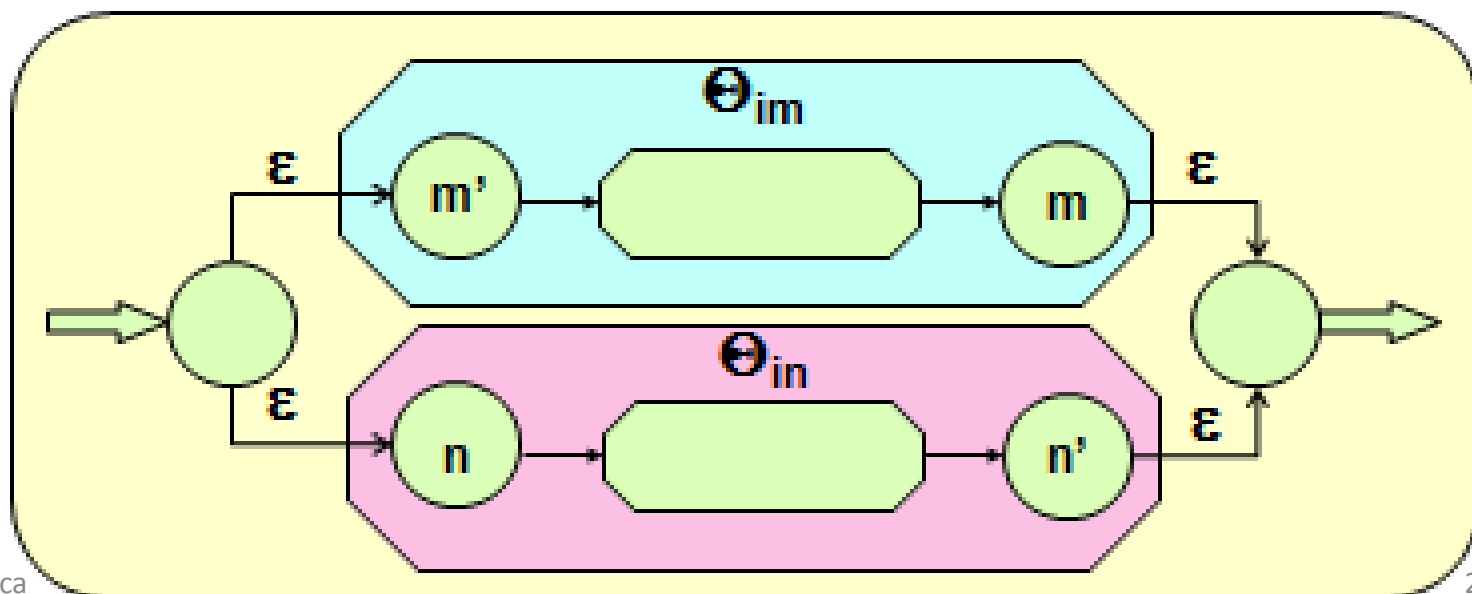
## 7 MÚLTIPLAS OPÇÕES (similar ao agrupamento entre parênteses) (união de linguagens)

$$\Theta_{im} \mid \Theta_{in}$$

Dados os dois diagramas seguintes:

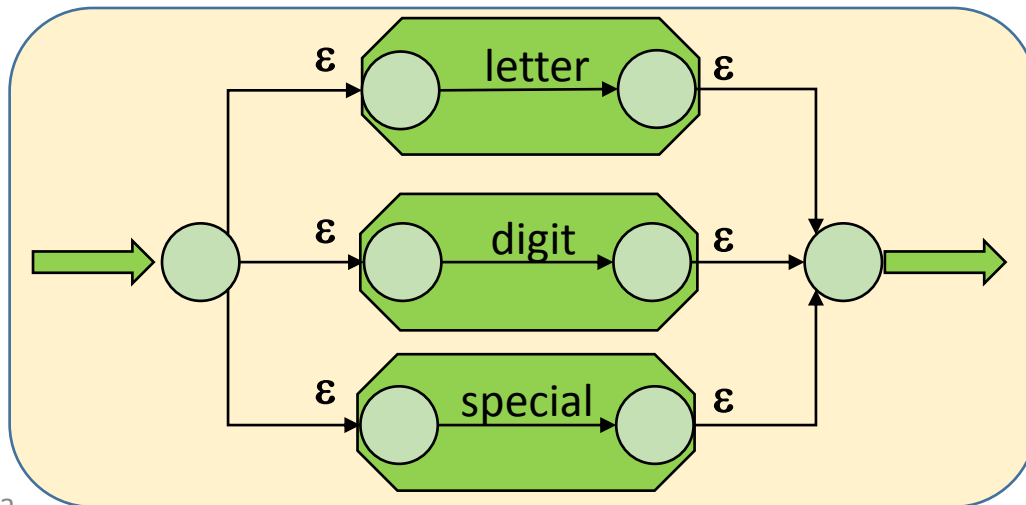
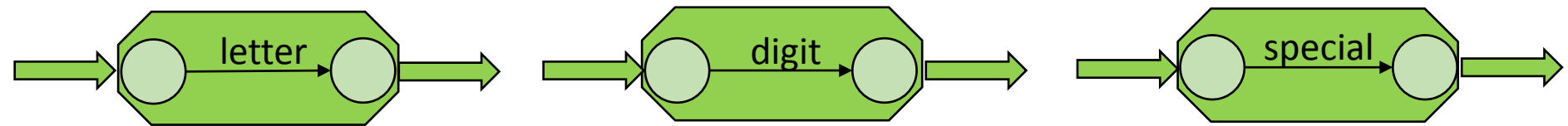


constrói-se



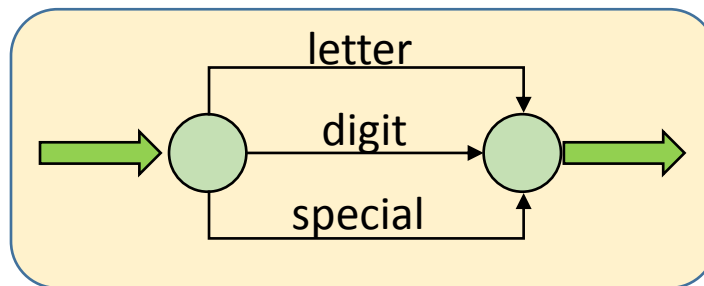
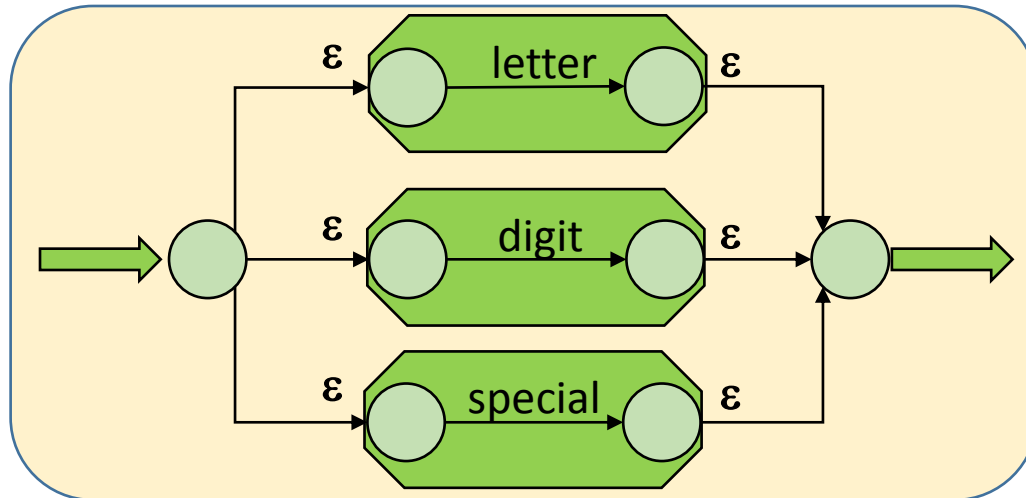
Character = letter | digit | special .

letter | digit | special



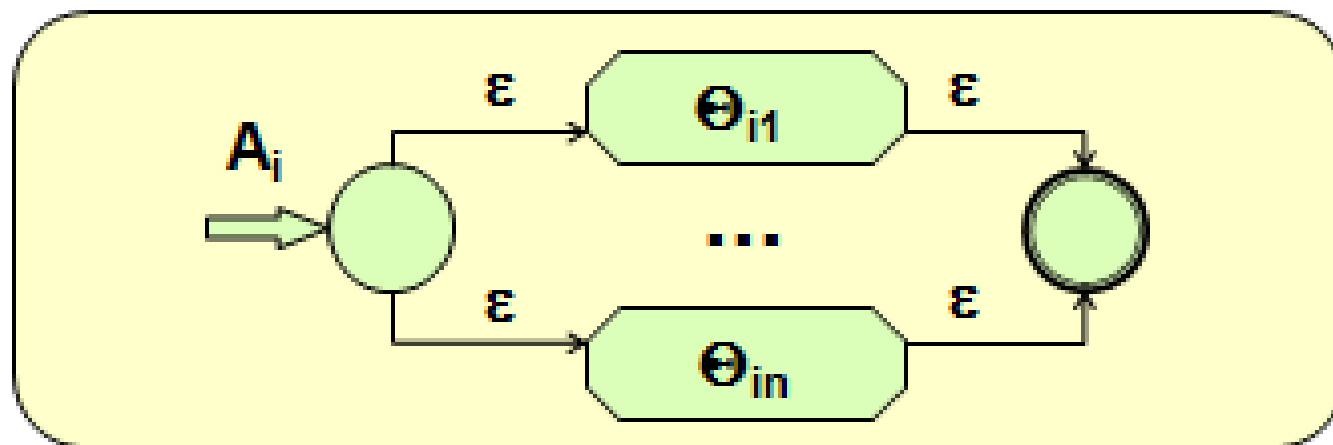
# Simplificação

Este autômato permite simplificações triviais, que podem ser feitas diretamente, eliminando todas as transições em vazio.



**8** DEFINIÇÃO DE UM NÃO-TERMINAL  $A_i$  como uma das opções das formas  $\Theta_{ij}$ , em que  $\Theta_{ij}$  são expressões de Wirth: (também similar ao agrupamento entre parênteses)

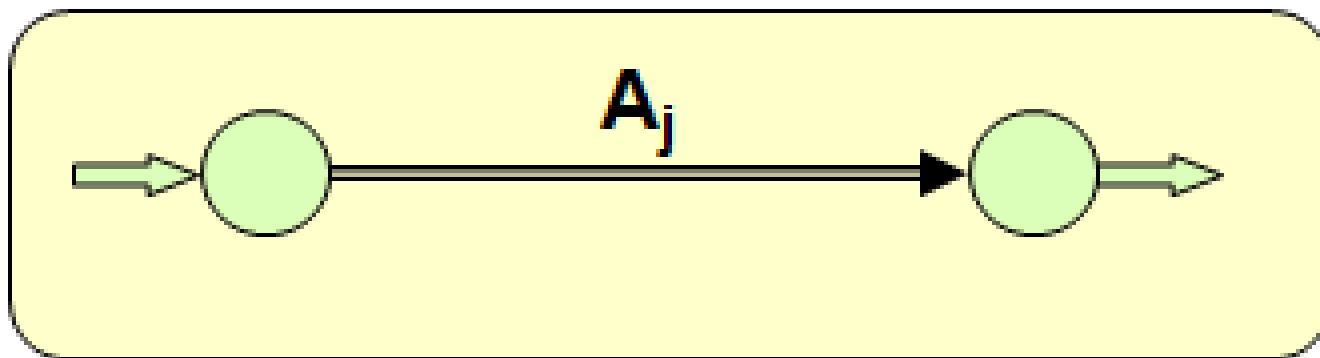
$$A_i = \Theta_{i1} \mid \Theta_{i2} \mid \dots \mid \Theta_{in} .$$



- Cada um dos blocos  $\Theta_{ij}$  denota uma parte do diagrama de estados de  $A_i$ , responsável pelo reconhecimento da forma  $\Theta_{ij}$

## 9 OCORRÊNCIAS DE NÃO-TERMINAIS (chamada de submáquina)

$A_j$





# Exercício

- Aplique a toda a gramática do Dartmouth Basic, inicialmente fornecida, as regras de obtenção de autômatos a partir de regras gramaticais na Notação de Wirth, aqui apresentadas. Você vai ficar com uma coleção de autômatos que efetuam reconhecimentos parciais da linguagem.
- Agrupe os autômatos parciais que se referem a partes regulares da linguagem e procure, através de substituições adequadas, eliminar a necessidade de alguns desses reconhecedores parciais. A meta é eliminá-los todos, deixando o reconhecedor completo expresso em um só autômato finito.
- Analogamente, considerando os não-terminais restantes, procure reduzir o número deles, deixando ao final apenas referências aos não-terminais ditos essenciais: a raiz da gramática e não-terminais que sejam independentes, auto-recursivos centrais, não elimináveis.