Nullable, First and Follow

José Romildo Malaquias

1 de novembro de 2017

```
S → if E then S else S
S → if E then S
S → cmd
E → id
```

Esta é a gramática livre de contexto!

S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

Acertando a gramática

Símbolo inicial: S

Novo símbolo inicial: S'

S' → S $

S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

Acertando a gramática

Símbolo inicial: S

Novo símbolo inicial: S'

Nova regra de produção: S' →S $

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow

Tabela de símbolos anuláveis, conjuntos first e conjuntos follow

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'			
S			
E			

Tabela de símbolos anuláveis, conjuntos first e conjuntos follow

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Encontrando os símbolos não terminais anuláveis

Inicialmente nenhum símbolo não terminal é anulável

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Regra com terminal no lado direito é inútil

S' → S $

S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Regra com terminal no lado direito é inútil

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Regra com terminal no lado direito é inútil

```
S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id
```

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Regra com terminal no lado direito é inútil

```
S' → S $
S  → if E then S else S
S  → if E then S
S  → cmd
E  → id
```

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Regra com terminal no lado direito é inútil

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×		
S	×		
E	×		

Encontrando os conjuntos first

Inicialmente são vazios

S' → S $

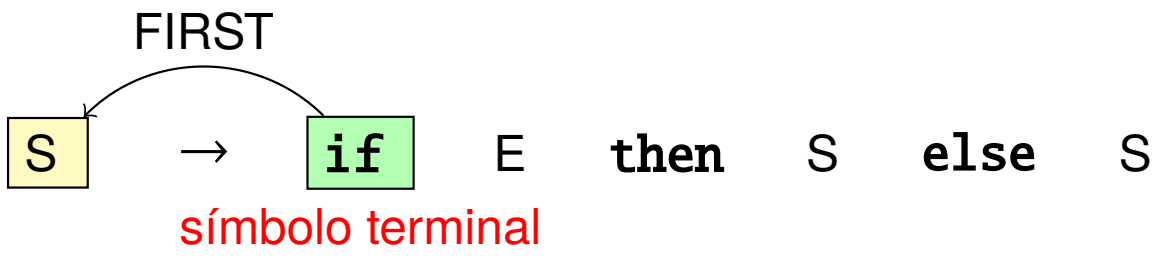
S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

NT	Nullable	First	Follow
S'	×		
S	×	**if**	
E	×		



FIRST(S) = { } ∪ { **if** }

S' → S $

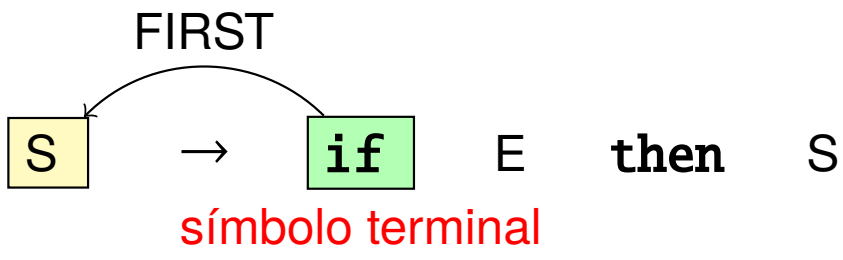
S → if E then S else S

S → if E then S

S → cmd

E → id

NT	Nullable	First	Follow
S'	×		
S	×	if	
E	×		



FIRST(S) = { **if** } ∪ { **if** }

S' → S $

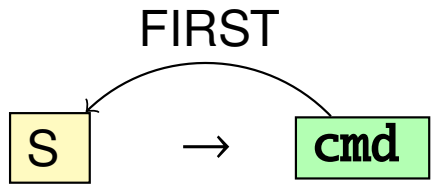
S → if E then S else S

S → if E then S

S → cmd

E → id

NT	Nullable	First	Follow
S'	×		
S	×	if cmd	
E	×		

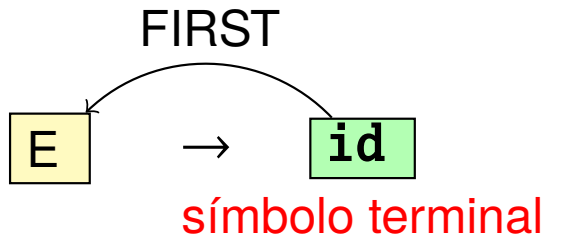


símbolo terminal

FIRST(S) = { if } ∪ { cmd }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

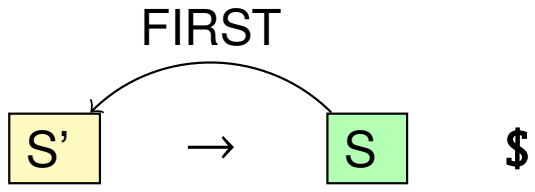
NT	Nullable	First	Follow
S'	×		
S	×	if cmd	
E	×	id	



FIRST(E) = { } ∪ { id }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	
E	×	id	

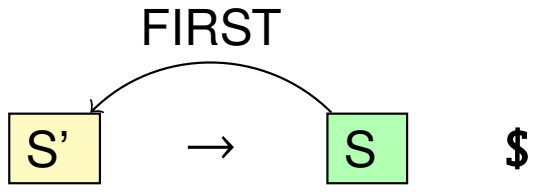


símbolo não terminal

FIRST(S') = { } ∪ { if ,cmd }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	
E	×	id	



símbolo não terminal

FIRST(S') = { if , cmd } ∪ { if ,cmd }

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

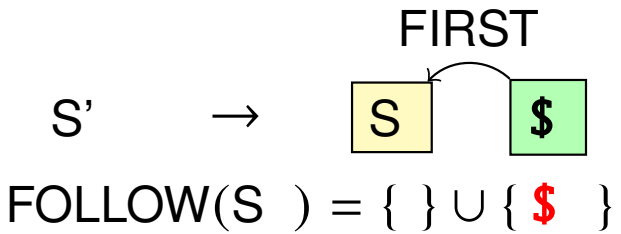
NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	
E	×	**id**	

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	
E	×	**id**	

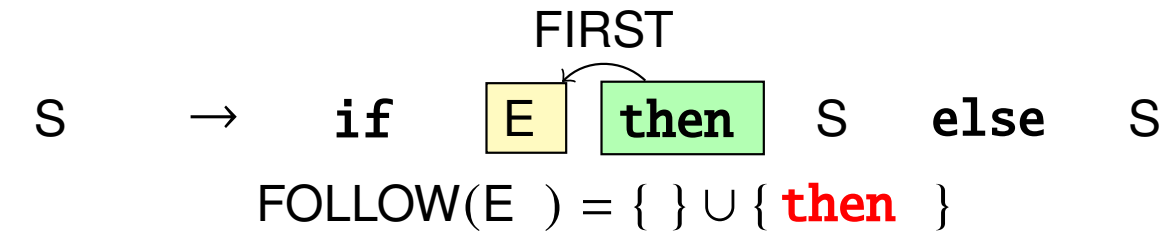
S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$
E	×	id	



S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$
E	×	id	then



S' → S $

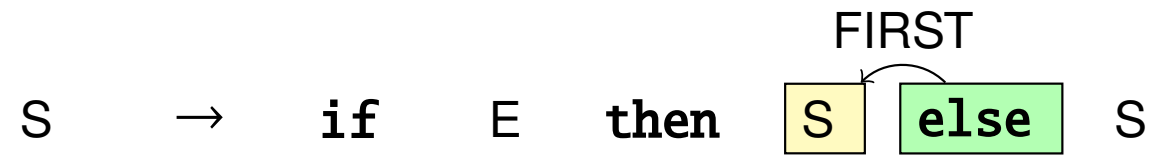
S → if E then S else S

S → if E then S

S → cmd

E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then



S' → S $

S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

S → **if** E **then** S **else** S

Náo há nenhum símbolo após o último não terminal

S' → S $

S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

FIRST

S → **if** E **then** S

FOLLOW(E) = { **then** } ∪ { **then** }

S' → S $

S → **if** E **then** S **else** S

S → **if** E **then** S

S → **cmd**

E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

S → **if** E **then** S

Náo há nenhum símbolo após o último não terminal

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

S →**cmd**

Nenhum símbolo não terminal no lado direito da regra

Regra inútil

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

E →**id**

Nenhum símbolo não terminal no lado direito da regra

Regra inútil

S' → S $

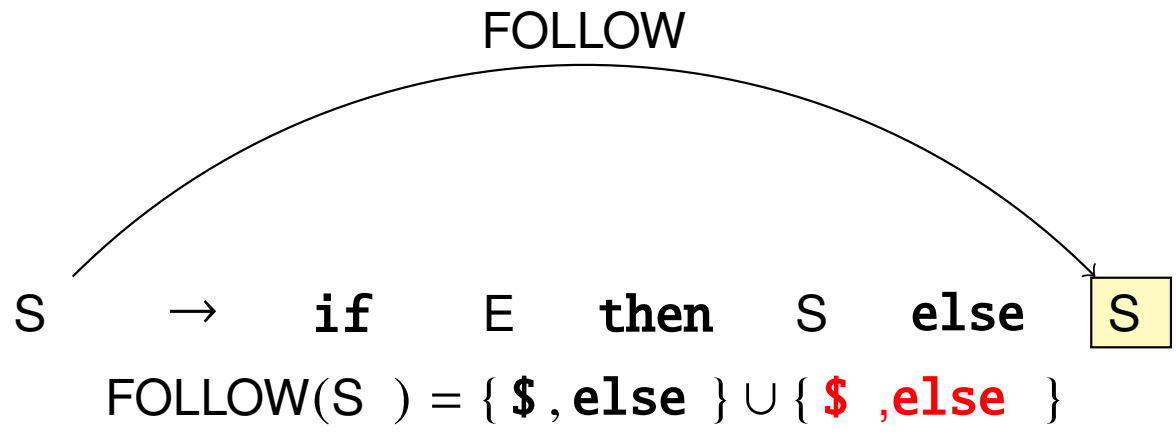
S → if E then S else S

S → if E then S

S → cmd

E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then



S' → S $

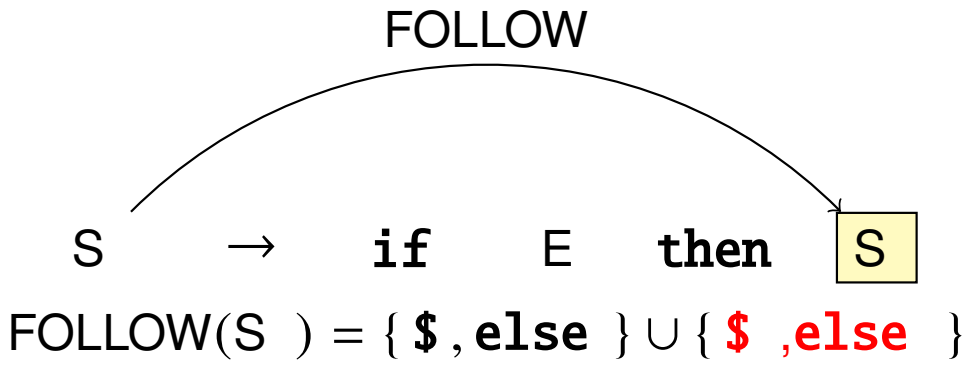
S → if E then S else S

S → if E then S

S → cmd

E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then



S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'						
S						
E						

Construção da tabela LL(1)

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'						
S						
E						

FIRST(S $) = { } ∪ { if cmd }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S						
E						

S' → S $
if cmd

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S						
E						

FIRST(if E then S else S) = { } ∪ { if }

S' → S $
S → **if** E **then** S **else** S
S → **if** E **then** S
S → **cmd**
E → **id**

NT	Nullable	First	Follow
S'	×	**if** **cmd**	
S	×	**if** **cmd**	**$** **else**
E	×	**id**	**then**

	$	**cmd**	**else**	**id**	**if**	**then**
S'		S **$**			S **$**	
S					**if** E **then** S **else** S	
E						

S → **if** E **then** S **else** S
if

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S					if E then S else S	
E						

FIRST(if E then S) = { } ∪ { if }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S					if E then S else S if E then S	
E						

S → if E then S
if

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S					if E then S else S if E then S	
E						

FIRST(cmd) = { } ∪ { cmd }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S		cmd			if E then S else S if E then S	
E						

S → cmd
cmd

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S		cmd			if E then S else S if E then S	
E						

FIRST(id) = { } ∪ { id }

S' → S $
S → if E then S else S
S → if E then S
S → cmd
E → id

NT	Nullable	First	Follow
S'	×	if cmd	
S	×	if cmd	$ else
E	×	id	then

	$	cmd	else	id	if	then
S'		S $			S $	
S		cmd			if E then S else S if E then S	
E				id		

E → id
id