

Analizadores descendentes

- LL(1) recursivos:
 - Comportamiento secuencial del analizador
 - Generación del reconocedor:
 - Una **función** por cada **no terminal**
 - Una función que:
 - Controle el símbolo de anticipación: **ant**
 - compare un **terminal** con el token de la cadena de entrada:
cmp(token)
 - Si hay varios consecuentes se usan DIR(símbolo) para identificar el consecuente



Analizadores descendentes

- LL(1) recursivos:

$$\begin{array}{lcl} S & ::= & i L a T z \\ L & ::= & c i L \mid \lambda \\ T & ::= & e \mid r \end{array}$$
$$\text{DIR}(S ::= iLaTz) = \{i\}$$
$$\text{DIR}(L ::= ciL) = \{c\}$$
$$\text{DIR}(L ::= \lambda) = \{a\}$$
$$\text{DIR}(T ::= e) = \{e\}$$
$$\text{DIR}(T ::= r) = \{r\}$$


Analizadores descendentes

- LL(1) recursivos:

```
void S (void){  
    cmp(i);  
    L();  
    cmp(a);  
    T();  
    cmp(z);  
}
```

$S ::= i L a T z$
 $L ::= c i L \mid \lambda$
 $T ::= e \mid r$

$DIR(S ::= iLaTz) = \{i\}$

$DIR(L ::= ciL) = \{c\}$

$DIR(L ::= \lambda) = \{a\}$

$DIR(T ::= e) = \{e\}$

$DIR(T ::= r) = \{r\}$



Analizadores descendentes

- LL(1) recursivos:

```
void S (void){  
    cmp(i); L(); cmp(a); T(); cmp(z);  
}
```

```
void L (void){  
    if (ant == c) {  
        cmp(c);  
        cmp(i);  
        L();  
    }  
}
```

$S ::= i L a T z$
 $L ::= c i L \mid \lambda$
 $T ::= e \mid r$

$DIR(S ::= iLaTz) = \{i\}$

$DIR(L ::= ciL) = \{c\}$

$DIR(L ::= \lambda) = \{a\}$

$DIR(T ::= e) = \{e\}$

$DIR(T ::= r) = \{r\}$



Analizadores descendentes

- LL(1) recursivos:

```
void S (void){  
    cmp(i); L(); cmp(a); T(); cmp(z);  
}
```

```
void L (void){  
    if (ant == c) {  
        cmp(c); cmp(i); L();  
    }  
}
```

```
void T(void){  
    if (ant == e) cmp(e);  
    else if (ant == r) cmp(r);  
    else Error();  
}
```

$S ::= i L a T z$
 $L ::= c i L \mid \lambda$
 $T ::= e \mid r$

¿pila del
autómata?

$DIR(S ::= iLaTz) = \{i\}$

$DIR(L ::= ciL) = \{c\}$

$DIR(L ::= \lambda) = \{a\}$

$DIR(T ::= e) = \{e\}$

$DIR(T ::= r) = \{r\}$



Analizadores descendentes

- Ejercicio:

Codificar el analizador LL(1) recursivo para la siguiente gramática:

$$E ::= \text{cte } E'$$
$$E' ::= + T E' \mid - T E' \mid \lambda$$
$$T ::= \text{cte } T'$$
$$T' ::= * \text{cte } T' \mid / \text{cte } T' \mid \lambda$$
