

Tema 3. Análisis Sintáctico

Objetivo.

El alumno explicará a detalle la etapa del análisis sintáctico en el proceso de compilación, así como las gramáticas idóneas para la definición de la estructura de los lenguajes de programación.

Contenido.

- 3.1 Gramáticas idóneas para análisis sintáctico.
- 3.2 Representación de sintaxis.
- 3.3 Clasificación de los analizadores sintácticos.

3.1 Gramáticas idóneas para el análisis sintáctico.

Recordemos que el Analizador Sintáctico, como su nombre lo dice, es el encargado de revisar la estructura del programa fuente, así como de observar que las instrucciones estén sintácticamente correctas.

Para realizar dicha función, el Analizador Sintáctico agrupa los componentes léxicos, recibidos del analizador léxico, en frases gramaticales. Dichas frases gramaticales son el resultado de aplicar las reglas o producciones de una gramática.

Las gramáticas que definen la sintaxis del lenguaje y la estructura de los programas fuente, son las gramáticas libres de contexto (GLC). Estas gramáticas deben cumplir con diferentes características las cuales seguramente las revisaste en la asignatura Lenguajes formales y autómatas:

- no deben ser ambiguas,
- ser gramáticas libres de ciclos,
- no tener no-terminales muertos y
- no tener elementos inaccesibles.

Actividad 3.1.1 Revisar a detalle las características y en su caso, los algoritmos que se pueden aplicar a las gramáticas libres de contexto para que cumplan con los requisitos arriba mencionados.

Ahora de manera esquemática, observemos qué elementos maneja el Analizador Sintáctico.



Componentes en el proceso de Análisis Sintáctico.

Observemos del esquema anterior:

- Las tablas sólo son consultadas por el analizador sintáctico; esto es, no realiza actualizaciones sobre ellas. Por esto es que veremos más adelante que será suficiente con tener como entrada una **cadena de átomos** que se puede obtener de la secuencia de tokens.
- Que el analizador sintáctico es en sí un **Autómata Push Down o de Pila (APD)**.
- Y que los tokens deben ser una frase gramatical obtenida de la aplicación de producciones de una **gramática libre de contexto**.

Entonces veamos que la frase gramatical es una cadena que pertenece al lenguaje definido por la gramática libre de contexto. Y que el lenguaje que es definido por dicha gramática lo reconoce el autómata push down. Es así como se relacionan los términos: lenguaje, gramática y autómata.

3.2 Representación de sintaxis.

Gramáticas libres de contexto.

Si bien la sintaxis de las instrucciones se da a través de las gramáticas libres de contexto, podemos usar dos notaciones para su representación:

Notación BNF (Backus Naur Form) y
Diagramas de tren.

Ahora hagamos ejercicios para definir gramáticas libres de contexto que describan la sintaxis de varias instrucciones de un lenguaje x, usando dichas notaciones.

Ejercicio 3.2.1 Elaborar una gramática libre de contexto que defina las sentencias declarativas de un lenguaje x, las cuales tienen la siguiente sintaxis:

<tipo de variable> <lista de variables>;

El tipo de variable puede ser 'real' o 'int'; la lista de variables es el conjunto no vacío de variables; cuando se declara más de una variable, deben ir separadas por coma. Ejemplos:

```
int a;  
real x0, x1, z;
```

Respuesta

a) Primeramente representaremos la sintaxis de la sentencia declarativa usando la notación BNF (Backus Naur Form).

Una vez entendida la sintaxis de la sentencia declarativa, definamos los componentes de la gramática libre de contexto:

Conjunto T (elementos terminales): { real int var , ; }

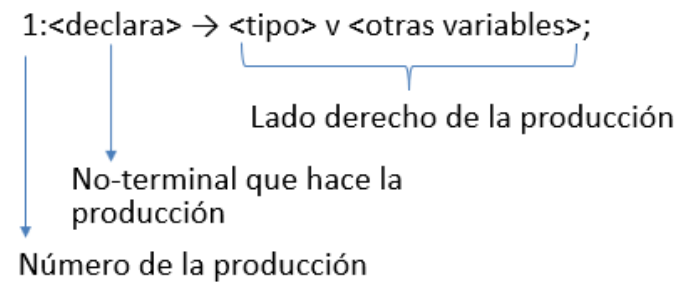
Las buenas prácticas nos dicen que los elementos terminales los representemos con una sola letra minúscula, por lo que la palabra 'real' la definiremos como el terminal 'r' y a la palabra 'int' la definiremos como el terminal 'i'. A las variables las definiremos como el terminal 'v'. Por último, a los símbolos ',' y ';' los mantendremos así.

Entonces el conjunto T es: { r i v , ; }

Ahora definamos las producciones de la gramática:

- 1:<declara> → <tipo> v <otras variables>;
- 2:<tipo> → r
- 3:<tipo> → i
- 4:<otras variables> → , v <otras variables>
- 5:<otras variables> → ξ

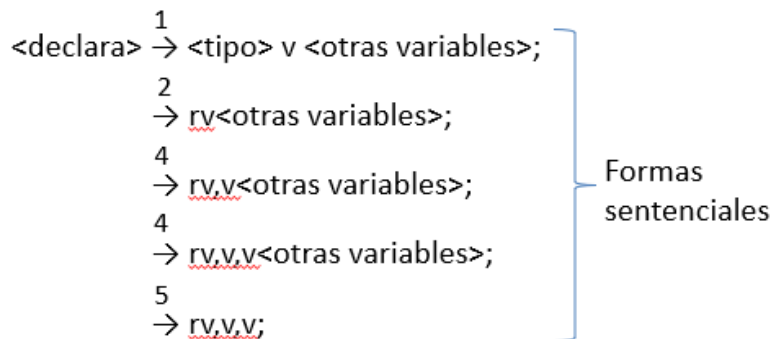
La notación BNF nos indica que las producciones o reglas de derivación deben tener el siguiente formato:



A la sentencia declarativa de ejemplo: real x0, x1, z;

Le corresponde la cadena de terminales: rv,v,v;

Para probar que dicha cadena es una frase gramatical de la gramática definida, aplicaremos las reglas de derivación por la izquierda para obtenerla:



El resultado de aplicar una o más reglas de derivación, se le llama *forma sentencial*; y a la forma sentencial que está formada por puros terminales, se le llama, además, *sentencia* o *frase gramatical*.

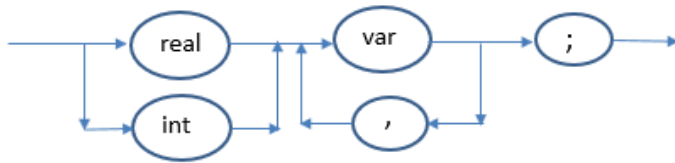
Nótese que arriba del símbolo → se coloca el número de la producción que se aplicó.

Actividad 3.2.1 Realiza la derivación por la izquierda para obtener la frase gramatical correspondiente a la sentencia declarativa de ejemplo: int a, b;

b) Ahora representaremos la sintaxis de la sentencia declarativa del ejercicio, usando diagramas de tren.

Un diagrama de tren está formado por *óvalos* (estaciones) que representan los elementos terminales de la gramática, y por *vías* representadas por líneas flechadas que indican la dirección por la que hay que seguir para visitar las estaciones e ir formando las frases gramaticales.

Entonces el diagrama de tren que define a las sentencias declarativas del ejercicio es:



Es importante notar que sólo hay una línea de entrada y una línea de salida por cada estación.

Ejercicio 3.2.2 Elaborar una gramática libre de contexto que defina la sentencia *for* de un lenguaje *x*, el cual tiene la siguiente sintaxis:

for *n* times [<sentencias>]

Donde:

- '*n*' es una constante entera o una variable de tipo entera.
- '*for*' y '*times*' son palabras reservadas.
- <sentencias> pueden ser 0 o más sentencias; para simplificar, considera que una sentencia cualquiera se representa por lo pronto con la letra '*s*' y es un terminal.

Respuesta.

a) Obtendremos la gramática en notación BNF.

Si definimos '*f*' para *for*, '*t*' para *times*, '*v*' para *var*, '*c*' para *cte*, entonces en conjunto *T* es: { *f t [v c s* }

1:<sentFor> → *f N t* [<sents>]

2:*N* → *v*

3:*N* → *c*

4:<sents> → *s* <sents>

5:<sents> → ξ

Ahora para la siguiente sentencia *for*, obtener su cadena de terminales y probar que dicha cadena es una frase gramatical de la gramática definida.

for 5 times [*s*₁ *s*₂ *s*₃]

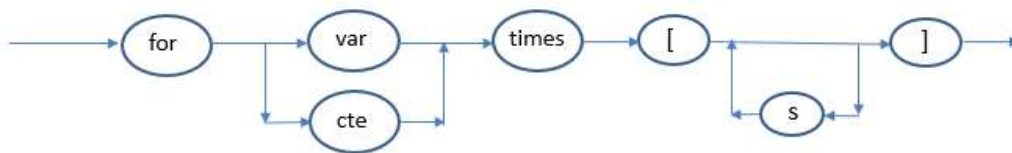
La correspondiente cadena de terminales es: fct[sss]

$\langle \text{sentFor} \rangle \xrightarrow{1} f N t [\langle \text{sents} \rangle]$
 $\xrightarrow{3} fct[\langle \text{sents} \rangle]$
 $\xrightarrow{4} fct[s \langle \text{sents} \rangle]$
 $\xrightarrow{4} fct[ss \langle \text{sents} \rangle]$
 $\xrightarrow{4} fct[sss \langle \text{sents} \rangle]$
 $\xrightarrow{5} fct[sss] \implies$ es la frase gramatical de la sentencia de ejemplo

Actividad 3.2.2 Realiza la derivación por la izquierda para obtener la frase gramatical correspondiente a la siguiente sentencia for: for x times []

b) Ahora representaremos la sintaxis de la sentencia *for* del ejercicio, usando diagramas de tren.

El diagrama de tren que define a la sentencia *for* del ejercicio es:



Actividad 3.2.3 Elaborar la gramática libre de contexto tanto en notación BNF como en diagrama de tren de la sentencia *case* de un cierto lenguaje, la cual tiene la siguiente sintaxis:

```

case of var {
    constCarac: <sentencias>
    constCarac: <sentencias>

    otherwise: <sentencias>
}
    
```

Donde:

- *case*, *of* y *otherwise* son palabras reservadas. Usar *c*, *o* y *w* como terminales.
- 'var' es un identificador. Usa *v* como terminal
- { } : son símbolos especiales. Usa el mismo símbolo como terminal.
- constCarac es una constante carácter. Usa *n* como terminal
- <sentencias> pueden ser 0 o más sentencias; para simplificar, considera que una sentencia cualquiera se representa por lo pronto con la letra 's' y es un terminal.

Ejemplo de una sentencia case:

```
case of valor {
  5: s1
    s2
  10: s3
    otherwise: s4
              s5
}
```

3.3 Clasificación de los analizadores sintácticos.

Los Analizadores sintácticos, también llamados *Parsers*, se clasifican de acuerdo a su método de análisis en dos grandes grupos: descendentes y ascendentes; y éstos a su vez, se clasifican de acuerdo a su implementación. Observemos el siguiente cuadro sinóptico:



En este curso estudiaremos los dos tipos de analizadores sintácticos descendentes, así como el ascendente por corrimiento-reducción.

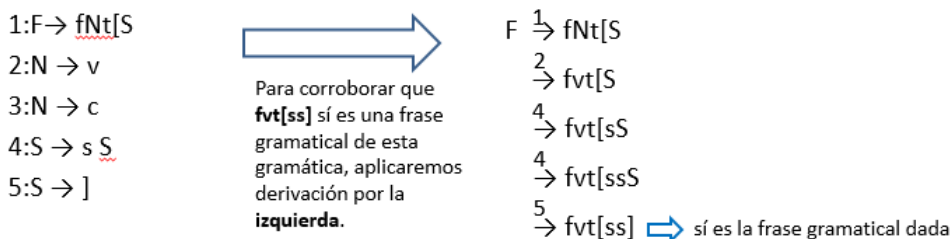
A continuación, veremos las generalidades de los analizadores sintácticos descendentes y ascendentes.

Análisis sintáctico descendente.

El método del análisis sintáctico **descendente** consiste en ir identificando qué reglas de derivación se aplicaron por la **izquierda** para obtener la sentencia o frase gramatical dada, partiendo del símbolo inicial de la gramática.

Hagamos un ejercicio para entender este método.

Ejercicio 3.3.1 Para la frase gramatical **fvt[ss]**, de la gramática dada, muestra cómo el analizador sintáctico descendente va reconociendo las reglas de derivación, aplicadas por la izquierda, partiendo del símbolo inicial de la gramática.



Una vez que aplicamos la derivación por la izquierda para obtener la frase gramatical dada, veamos el procedimiento que haría el analizador sintáctico descendente para reconocer dicha frase gramatical:

Paso 1:

Sabemos que el analizador sintáctico es un autómata de pila, por lo que voy a ir indicando la situación de la pila y de la cadena de entrada (la frase gramatical). Para un análisis descendente. La situación inicial de la pila es con el símbolo inicial de la gramática finalizada con el símbolo Λ que indica pila vacía; agregaremos a la cadena de entrada el símbolo \dagger como indicador de fin de cadena. Entonces, para iniciar el reconocimiento, ésta es la situación inicial:

Cadena	Pila	
fvt[ss] \dagger	F Λ	F es el símbolo inicial de la gramática

Paso 2:

El analizador revisa qué producción de F (que está en el tope de la pila), inicia con 'f' que es el que inicia la cadena, y observa que es la producción 1: $F \rightarrow fNt[S$, entonces sustituye el símbolo que está en el tope de la pila por todos los elementos del lado derecho de la producción que le siguen a la 'f', o sea por $Nt[S$, y lee el siguiente elemento de la cadena.

Entonces así queda la situación:

Cadena	Pila	Producción reconocida
vt[ss] \dagger	Nt[S Λ	1: $F \rightarrow fNt[S$

Paso 3:

Ahora el analizador revisa qué producción de N (que está en el tope de la pila), inicia con 'v' que es el que leyó de la cadena, y observa que es la producción 2: $N \rightarrow v$, por lo que, como no hay elementos del lado derecho de 'v' en dicha producción, se hace un *pop* en la pila, y se lee el siguiente elemento de la cadena:

Cadena	Pila	Producción reconocida
t[ss] \dagger	t[S Λ	2: $N \rightarrow v$

Paso 4:

Como en el tope de la pila está el terminal 't', debe haber leído ese mismo símbolo de la cadena de entrada. Como sí es así, entonces hace un *pop* en la pila y lee el siguiente elemento:

Cadena	Pila	Producción reconocida
[ss] \dagger	[S Λ	no aplica

Paso 5:

Como en el tope de la pila está el terminal '[', debe haber leído ese mismo símbolo de la cadena de entrada. Como sí es así, entonces hace un *pop* en la pila y lee el siguiente elemento:

Cadena	Pila	Producción reconocida
ss] \dagger	S Λ	no aplica

Paso 6:

Ahora el analizador revisa qué producción de S (que está en el tope de la pila), inicia con ' s ' que es el que leyó de la cadena, y observa que es la producción $4:S \rightarrow s S$, por lo que, como está el elemento S del lado derecho de ' s ' y coincide con el mismo elemento en el tope de la pila, se hace sólo la lectura del siguiente elemento de la cadena de entrada:

Cadena	Pila	Producción reconocida
$s] \uparrow$	$S \wedge$	$4:S \rightarrow s S$

Paso 7:

Se presenta la misma situación del paso 6:

Cadena	Pila	Producción reconocida
$] \uparrow$	$S \wedge$	$4:S \rightarrow s S$

Paso 8:

Ahora el analizador revisa qué producción de S (que está en el tope de la pila), inicia con ' $]$ ' que es el que leyó de la cadena, y observa que es la producción $5:S \rightarrow]$, por lo que, como no hay elementos del lado derecho de ' $]$ ' en dicha producción, se hace un *pop* en la pila, y se lee el siguiente elemento de la cadena:

Cadena	Pila	Producción reconocida
\uparrow	\wedge	$5:S \rightarrow]$

Paso 9:

Debido a que se leyó el indicador de fin de cadena y en la pila sólo está el indicador de pila vacía, entonces se reconoce la cadena de entrada. Es decir, sí es una frase gramatical perteneciente a la gramática dada.

Revisemos qué producciones fue reconociendo:

1: $F \rightarrow fNt[S$	}	Son las producciones que si las aplicamos, obtenemos la frase gramatical a reconocer.
2: $N \rightarrow v$		
4: $S \rightarrow s S$		
4: $S \rightarrow s S$		
5: $S \rightarrow]$		

Ahora que ya sabemos la forma en que hace el reconocimiento el analizador sintáctico **descendente**, resolvamos otro ejercicio.

Ejercicio 3.3.2 Muestra cómo hace el analizador sintáctico **descendente** el reconocimiento de la frase gramatical $fct[s]$ de la misma gramática del ejercicio anterior. Después aplica las producciones reconocidas conservando el orden para obtener dicha cadena.

1: $F \rightarrow fNt[S$
2: $N \rightarrow v$
3: $N \rightarrow c$
4: $S \rightarrow sS$
5: $S \rightarrow]$

Reconocimiento

Cadena	Pila	Producción reconocida
<u>f</u> ct[s]↑	F Λ	No aplica
c <u>t</u> [s]↑	Nt[SΛ	1: $F \rightarrow fNt[S$
t[s]↑	t[SΛ	3: $N \rightarrow c$
[s]↑	[SΛ	No aplica
s]↑	SΛ	No aplica
]↑	SΛ	4: $S \rightarrow sS$
↑	Λ	5: $S \rightarrow]$
		Se reconoce la frase gramatical

Como la situación final es que leyó el indicador de fin de cadena y la pila está vacía, entonces la cadena de entrada es reconocida como frase gramatical válida.

Ahora apliquemos las producciones en el orden en que fueron reconocidas: 1,3,4 y 5

1: $F \rightarrow fNt[S$
2: $N \rightarrow v$
3: $N \rightarrow c$
4: $S \rightarrow sS$
5: $S \rightarrow]$



$F \xrightarrow{1} fNt[S$
 $\xrightarrow{3} fcvt[S$
 $\xrightarrow{4} fct[sS$
 $\xrightarrow{5} fct[s] \Rightarrow$ sí es la frase gramatical dada.

Actividad 3.3.1 Muestra cómo el analizador sintáctico **descendente** revisa la siguiente frase gramatical dssswv; que pertenece a la gramática dada (define la instrucción do-while de un lenguaje x). Después aplica las producciones reconocidas conservando el orden para obtener dicha cadena.

1: $D \rightarrow dsSv;$
2: $S \rightarrow sS$
3: $S \rightarrow w$

Análisis sintáctico ascendente.

El método del análisis sintáctico **ascendente** consiste en ir identificando qué reglas de derivación por la **derecha** se aplicaron para obtener la sentencia o frase gramatical dada, partiendo de la última regla aplicada hasta llegar símbolo inicial de la gramática.

Hagamos un ejercicio para entender este método.

Ejercicio 3.3.3 Para la frase gramatical **fvt[ss]**, de la gramática dada, muestra cómo el analizador sintáctico ascendente va reconociendo las reglas de derivación, aplicadas por la derecha, se llega al símbolo inicial de la gramática.

1: $F \rightarrow fNt[S$
2: $N \rightarrow v$
3: $N \rightarrow c$
4: $S \rightarrow s S$
5: $S \rightarrow]$



Para corroborar que **fvt[ss]** sí es una frase gramatical de esta gramática, aplicaremos derivación por la **derecha**.

$F \xrightarrow{1} fNt[S$
 $\xrightarrow{4} fNt[sS$
 $\xrightarrow{4} fNt[ssS$
 $\xrightarrow{5} fNt[ss]$
 $\xrightarrow{2} fvt[ss]$ \Rightarrow sí es la frase gramatical dada

Como en el método del análisis sintáctico **ascendente** se basa en reconocer que en el tope de la pila ya se tiene el lado derecho de una producción, señalaré en la derivación por la derecha realizada, para cada forma sentencial, la cadena de elementos que son el lado derecho de una producción.

$F \xrightarrow{1} fNt[S$
Prod 1
 $\xrightarrow{4} fNt[sS$
Prod 4
 $\xrightarrow{4} fNt[ssS$
Prod 4
 $\xrightarrow{5} fNt[ss]$
Prod 5
 $\xrightarrow{2} fvt[ss]$
Prod 2

Recordemos que la forma sentencial es toda la cadena resultado de ir aplicando las reglas de derivación.

En cada forma sentencial señalé la subcadena que pertenece al lado derecho de la producción indicada.

Dicha subcadena tiene gran relevancia porque en ella se basa el procedimiento del reconocimiento ascendente.

Procedimiento de reconocimiento ascendente.

- El analizador sintáctico ascendente, cada vez que vaya leyendo un carácter de la cadena de entrada la va a ir colocando en el tope de la pila, esta operación se le llama “corrimiento”.
- Cuando el lado derecho de una producción se encuentre en el tope de la pila del analizador sintáctico ascendente, éste realizará una “reducción”, es decir sacará toda la subcadena de la pila y la sustituirá por el no terminal que se encuentra a la izquierda de la producción correspondiente a la subcadena.

Apliquemos este procedimiento para la frase gramatical del ejercicio **fvt[ss]**↓.

Paso 1:

Voy a ir indicando la situación de la pila y de la cadena de entrada (la frase gramatical), para un análisis descendente. La situación inicial de la pila para un análisis sintáctico **ascendente** es con el símbolo Λ que indica pila vacía y a la cadena se le agrega el símbolo ↓ como indicador de fin de cadena. Entonces, para iniciar el reconocimiento, ésta es la situación inicial:

Cadena	Pila
fvt[ss]↓	Λ

Paso 2:

El analizador hace *corrimiento* de 'f', es decir, lo coloca en el tope de la pila. Entonces la situación se encuentra así:

Cadena	Pila	Operación
vt[ss]↓	Λf	corrimiento de 'f'

Paso 3:

Ahora el analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de 'v', es decir, coloca a 'v' en el tope de la pila:

Cadena	Pila	Operación
t[ss]↓	Λfv	corrimiento de 'v'

Paso 4:

El analizador detecta que en el tope de la pila está el lado derecho de la producción 2, o sea la subcadena "v", por lo que hace la *reducción* por la producción 2, sustituyendo a "v" por el no terminal 'N' ya que es el lado izquierdo de la producción 2: $N \rightarrow v$

Cadena	Pila	Operación
t[ss]↓	ΛfN	reducción por la producción 2

Paso 5:

El analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de 't':

Cadena	Pila	Operación
[ss]↓	ΛfNt	corrimiento de 't'

Paso 6:

Ahora el analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de '[' :

Cadena	Pila	Operación
ss]↓	ΛfNt[corrimiento de '['

Paso 7:

El analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de 's' :

Cadena	Pila	Operación
s]↓	ΛfNt[s	corrimiento de 's'

Paso 8:

El analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de 's':

Cadena	Pila	Operación
]↓	ΛfNt[ss	corrimiento de 's'

Paso 9:

Ahora el analizador revisa si en el tope de la pila tiene el lado derecho completo de una producción; como no es así, realiza el *corrimiento* de ']' :

Cadena	Pila	Operación
↓	ΛfNt[ss]	corrimiento de ']'

Paso 10:

El analizador detecta que en el tope de la pila está el lado derecho de la producción 5, o sea la subcadena "]", por lo que hace la *reducción* por la producción 5, sustituyendo a "]" por el no terminal 'S' ya que es el lado izquierdo de la producción 5: $S \rightarrow]$

Cadena	Pila	Operación
↓	ΛfNt[ssS	reducción por la producción 5

Paso 11:

El analizador detecta que en el tope de la pila está el lado derecho de la producción 4, o sea la subcadena "sS", por lo que hace la *reducción* por la producción 4, sustituyendo a "sS" por el no terminal 'S' ya que es el lado izquierdo de la producción 4: $S \rightarrow sS$

Cadena	Pila	Operación
↓	ΛfNt[sS	reducción por la producción 4

Paso 12:

Se presenta la misma situación del paso 10, por lo que hace la *reducción* por la producción 4, sustituyendo a "sS" por el no terminal 'S'.

Cadena	Pila	Operación
↓	ΛfNt[S	reducción por la producción 4

Paso 13:

El analizador detecta que en el tope de la pila está el lado derecho de la producción 1, o sea la subcadena "fNt[S", por lo que hace la *reducción* por la producción 1, sustituyendo a "fNt[S" por el no terminal 'F' ya que es el lado izquierdo de la producción 1: $F \rightarrow fNt[S$

Cadena	Pila	Operación
↓	ΛF	reducción por la producción 1

Paso 14:

Debido a que se leyó el indicador de fin de cadena y en el tope de la pila está el símbolo inicial de la gramática, la frase gramatical es reconocida.

Revisemos qué producciones fue reconociendo:

2: $N \rightarrow v$	}	Son las producciones, en sentido inverso , que aplicamos a la gramática haciendo una derivación por la derecha para obtener la frase gramatical a reconocer.
5: $S \rightarrow]$		
4: $S \rightarrow sS$		
4: $S \rightarrow sS$		
1: $F \rightarrow fNt[S$		

Importante:

Para generar la frase gramatical a reconocer, aplicamos derivación por la **derecha** de la siguiente secuencia de producciones: 1, 4, 4, 5 y 2

Sin embargo, el analizador sintáctico **ascendente**, reconoce las producciones en la siguiente secuencia: 2, 5, 4, 4 y 1. Es decir, de la última a la primera que como se generó.

Ejercicio 3.3.4 Muestra cómo hace el analizador sintáctico **ascendente**, el reconocimiento de la frase gramatical fct[s] de la misma gramática del ejercicio anterior. Después aplica las producciones reconocidas en el sentido inverso para obtener dicha cadena.

1:F → fNt[S

2:N → v

3:N → c

4:S → s S

5:S →]

Reconocimiento

Cadena	Pila	Operación
fct[s]↓	Λ	No aplica
ct[s]↓	Λf	Corrimiento de 'f'
t[s]↓	Λfc	Corrimiento de 's'
t[s]↓	ΛfN	Reducción por producción 3
[s]↓	ΛfNt	Corrimiento de 't'
s]↓	ΛfNt[Corrimiento de '['
]↓	ΛfNt[s	Corrimiento de 's'
↓	ΛfNt[s]	Corrimiento de ']'
↓	ΛfNt[sS	Reducción por producción 5
↓	ΛfNt[S	Reducción por producción 4
↓	ΛF	Reducción por producción 1
		Se reconoce la frase gramatical

Como la situación final es que leyó el indicador de fin de cadena y en el tope de la pila está el símbolo inicial de la gramática, entonces la cadena de entrada es reconocida como frase gramatical válida.

Ahora apliquemos las producciones en el **orden inverso** en el que fueron reconocidas: 1, 4, 5 y 3

1:F → fNt[S

2:N → v

3:N → c

4:S → s S

5:S →]



$F \xrightarrow{1} fNt[S$
 $\xrightarrow{4} fNvt[sS$
 $\xrightarrow{5} fNt[s]$
 $\xrightarrow{3} fct[s] \Rightarrow$ sí es la frase gramatical dada.

Actividad 3.3.2 Muestra cómo el analizador sintáctico **ascendente** revisa la siguiente frase gramatical `dssswv;` que pertenece a la gramática dada (define la instrucción `do-while` de un lenguaje `x`). Después aplica las producciones reconocidas en el orden inverso para obtener dicha cadena.

1: $D \rightarrow dsSv;$

2: $S \rightarrow sS$

3: $S \rightarrow w$