



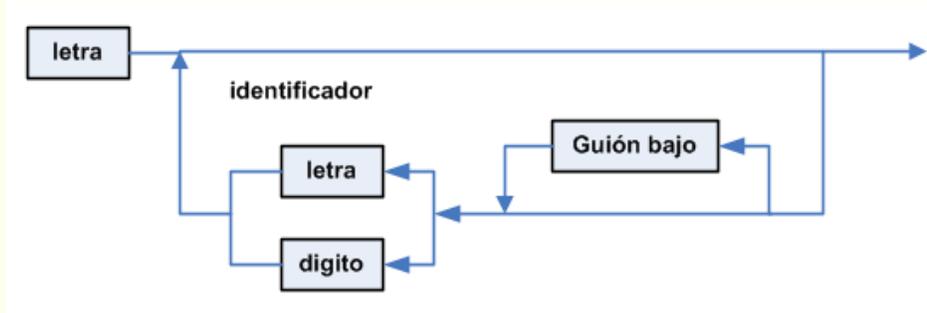
Diagramas de Conway

Es una herramienta gráfica para describir la sintaxis de un lenguaje. En ella se usan símbolos terminales y no terminales, que se representan:

terminales con un círculo o un rectángulo redondeado

no terminales con un rectángulo

Los diagramas se enlazan con *flechas* que indican el orden en que se debe realizar la interpretación. Por ejemplo la sintaxis de *identificador* se hace así



En esta definición vemos que

La *opción* es un camino alternativo que se puede seguir o no, como en el caso del guión bajo

Las *alternativas* están en caminos que bifurcan como en letra y dígito.

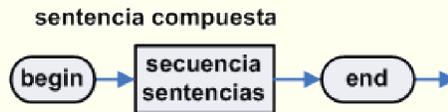
Lo que se puede *repetir* está formando un circuito que se puede recorrer cualquier número de veces.

El elemento no terminal que se está definiendo figura como una etiqueta.

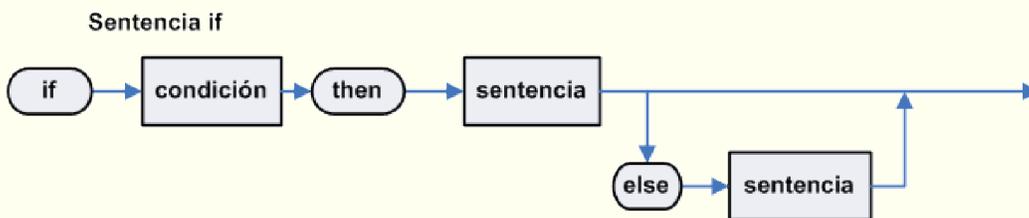
La sintaxis de la *secuencia de sentencias* es



La sintaxis de la *sentencia compuesta* es



La sintaxis de la *sentencia if*

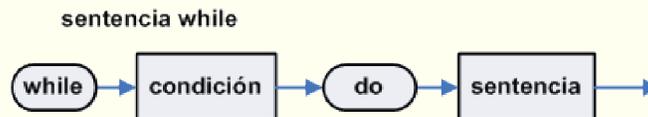


La semántica de la sentencia if dice que

La condición tiene que ser boolean y definida en el momento de ejecutar la sentencia.

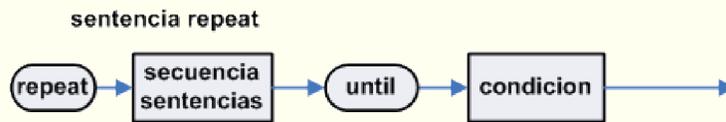
Las sentencias de las partes then o else pueden ser de cualquier tipo.

La *sentencia while* también necesita una condición de tipo boolean y plenamente definida en el momento de ejecutar la sentencia while. El ciclo se repite mientras la condición se evalúe a true y se sale la primera vez que se evalúa a false.

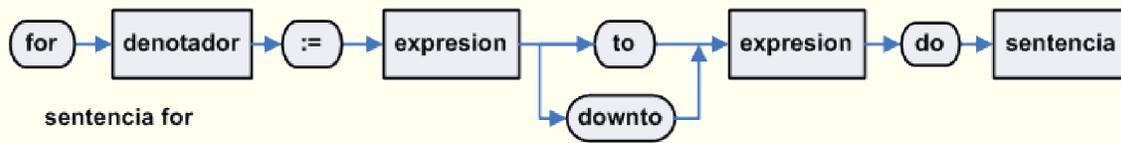


La sentencia que constituye el cuerpo de while puede ser cualquiera.

La *sentencia repeat* ejecuta una secuencia de sentencias hasta que una condición lógica toma el valor true. Por tanto se ejecuta mientras se evalúa a false.



La *sentencia for* tiene esta sintaxis



La *semántica de for* (resumida)

- El denotador debe serlo de una variable de tipo ordinal
- La expresión inicial debe ser definida y de tipo ordinal compatible en asignación con la variable
- La expresión final debe ser definida y de tipo ordinal compatible en asignación con la variable
- Si se sigue el camino to, la variable progresa al sucesor del valor actual y si se sigue el de downto al predecesor.
- La sentencia puede ser cualquiera.
- La variable de control queda indefinida al terminar for.

COMPATIBLE EN ASIGNACION: un valor de cierto tipo tiene esta relación con una variable del mismo o de otro tipo, cuando es posible almacenar el valor en la variable.

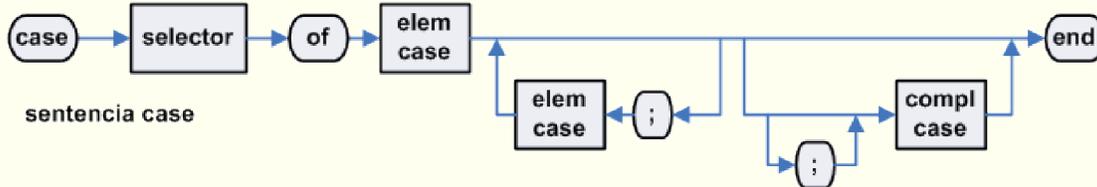
Por ejemplo: en variables de tipo real podemos almacenar números reales o números enteros. Hay compatibilidad de asignación entre los valores enteros o reales y las variables reales.

Sin embargo, no hay compatibilidad de asignación entre los valores reales y las variables enteras: no podemos guardar un valor real en una variable entera.

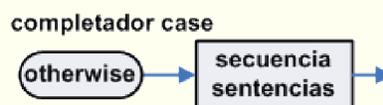
Existen valores incompatibles en asignación con cualquier tipo de variable. Por ejemplo los *valores de tipo archivo son inasignables*.

En general existe compatibilidad de asignación entre los valores de un tipo y las variables del mismo tipo. Salvo para los ficheros.

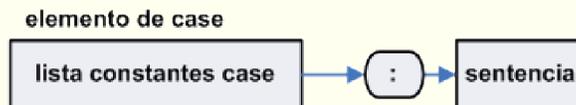
Finalmente, la *sintaxis de case* es



donde el completador case



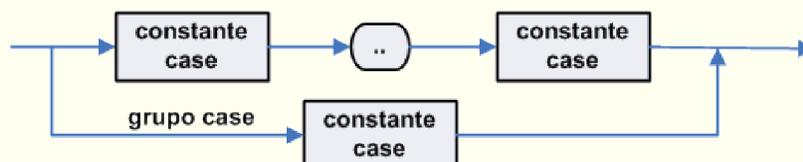
el elemento de case es



la lista de constantes de case es



y un grupo case es



y una constante case es un valor ordinal del mismo tipo que el selector.

