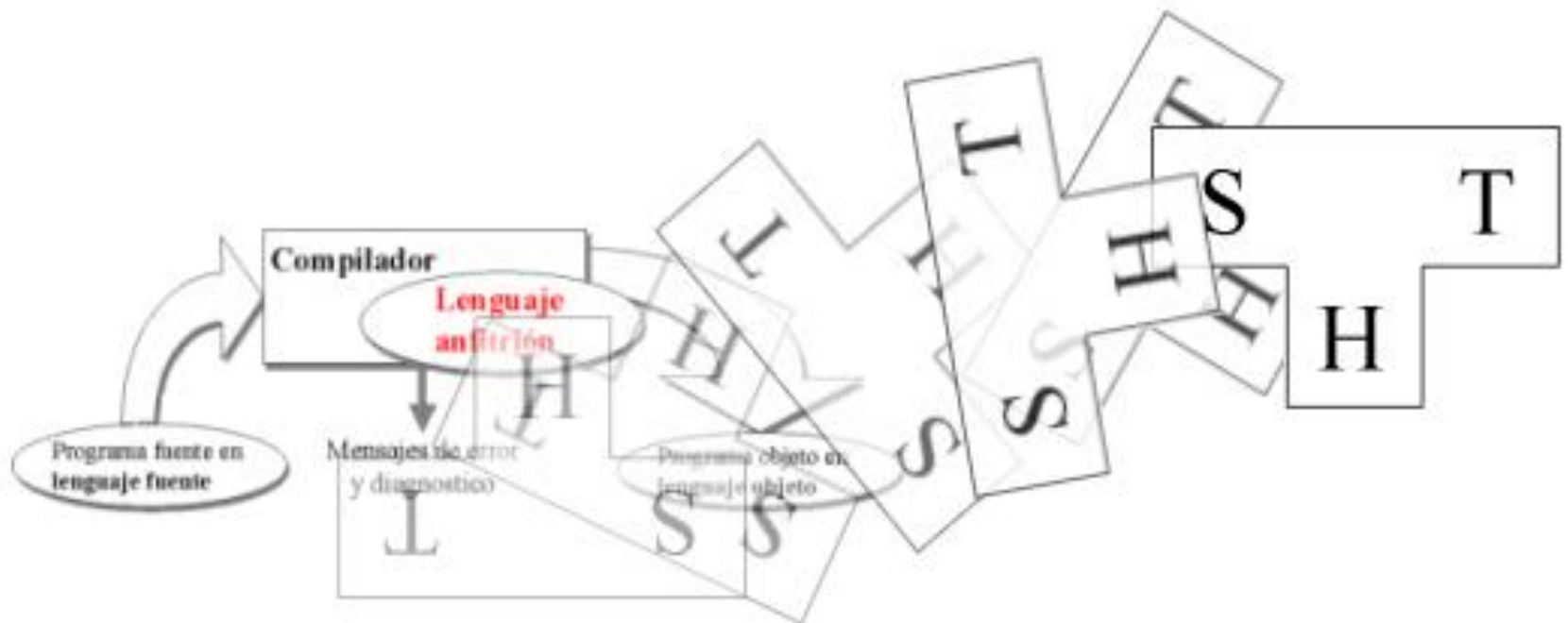
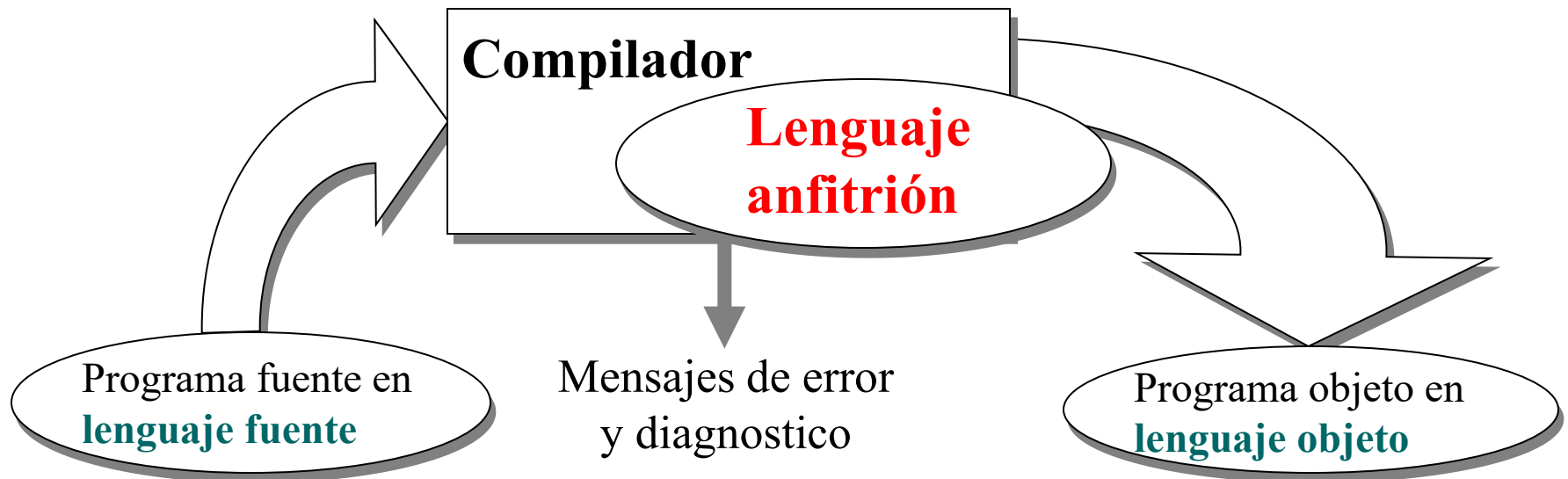


Compilación cruzada y arranque automático por transferencia



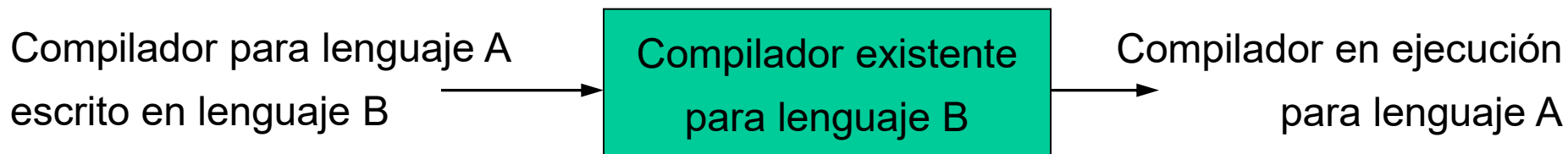
Dr. César García Osorio

- Los factores determinantes de la estructura de un compilador son el **lenguaje fuente** y el **lenguaje objetivo**.
- Existe un tercer lenguaje involucrado en el proceso de construcción de compiladores: el lenguaje en el que el compilador mismo está escrito, el **lenguaje anfitrión**.



- El lenguaje anfitrión para el compilador ejecutable tiene que ser lenguaje máquina. Para los primeros compiladores esta era la única opción.
- En la actualidad un enfoque más razonable es escribir el compilador en otro lenguaje para el cual ya exista un compilador.
- Si el compilador existente ya se ejecuta en la máquina objetivo, entonces solamente necesitamos compilar el nuevo compilador utilizando el existente para obtener un programa ejecutable:

I

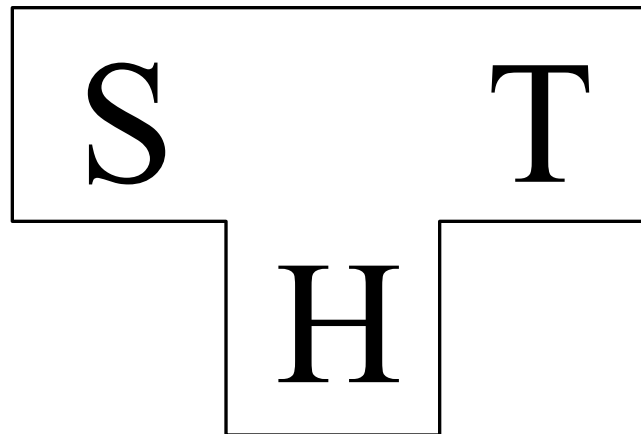


Diagramas T

(1)

II

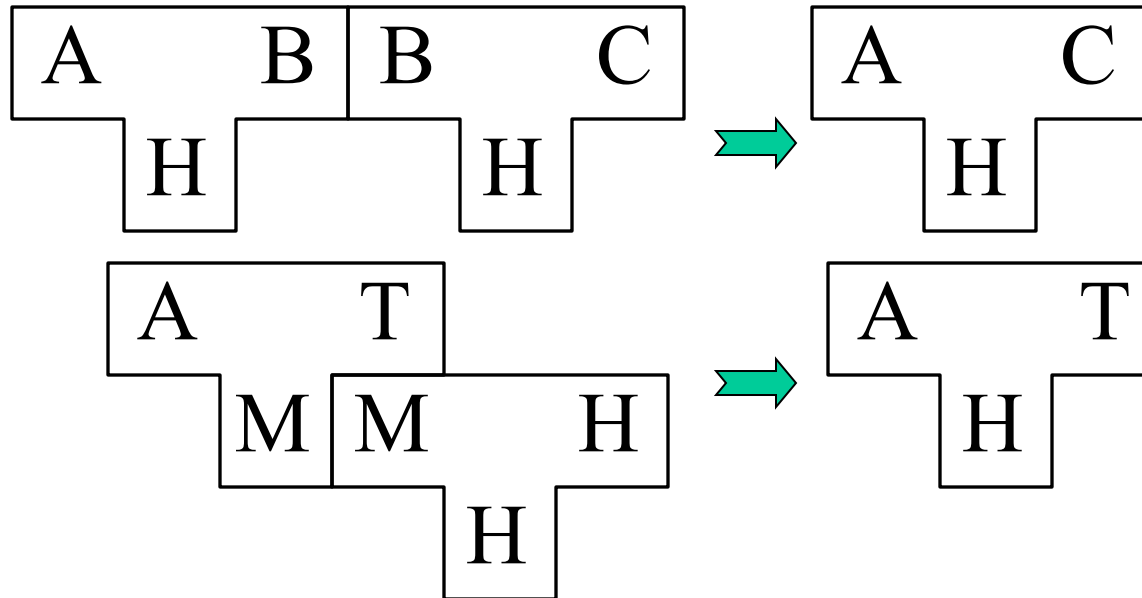
- Si el compilador existente para el lenguaje B se ejecuta en una máquina diferente de la máquina objetivo la compilación produce entonces un **compilador cruzado**, es decir, un compilador que genera código objetivo para una máquina diferente de aquella en la que puede ejecutarse.
- Ésta y otras situaciones más complejas se describen mejor al esquematizar un compilador como un diagrama T (llamado así debido a su forma).
- Un compilador escrito en el lenguaje H (*host*) que traduce de lenguaje S (*source*) a lenguaje T (*target*) se dibuja como:



Diagramas T

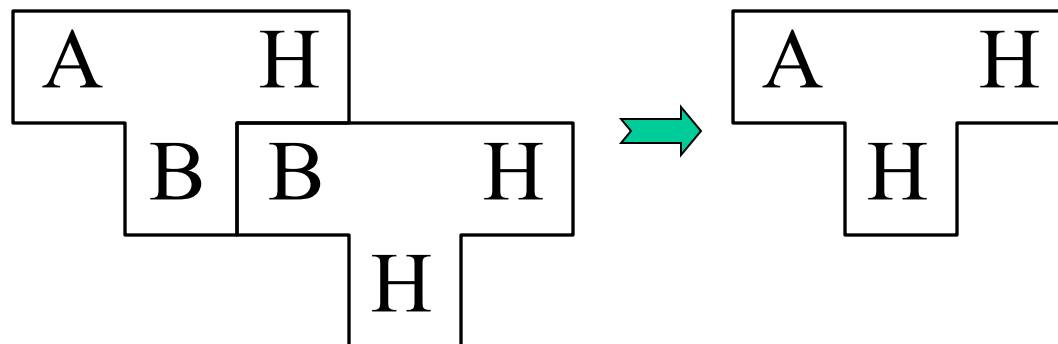
(2)

- Los diagramas T se pueden combinar en dos maneras.



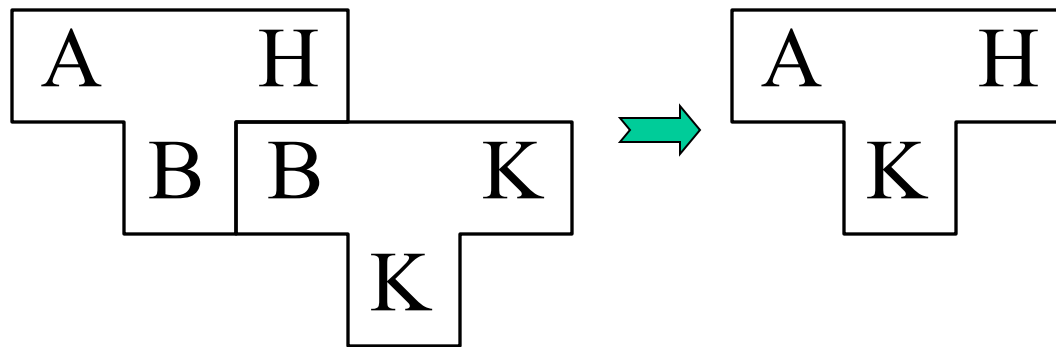
Obtención de un
compilador cruzado

- El uso de un compilador preexistente para compilar un nuevo compilador (I) es un caso especial del diagrama anterior



Compilador cruzado

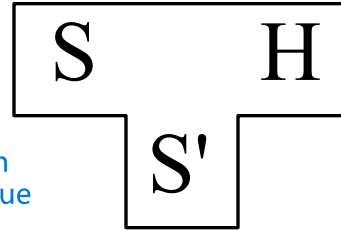
- Cuando el compilador del lenguaje B se ejecuta en una máquina diferente (II), lo cual resulta en un compilador cruzado para A, puede describirse de manera similar como sigue:



Arranque automático por transferencia 1

- Es común escribir un compilador en el mismo lenguaje que está por compilarse:

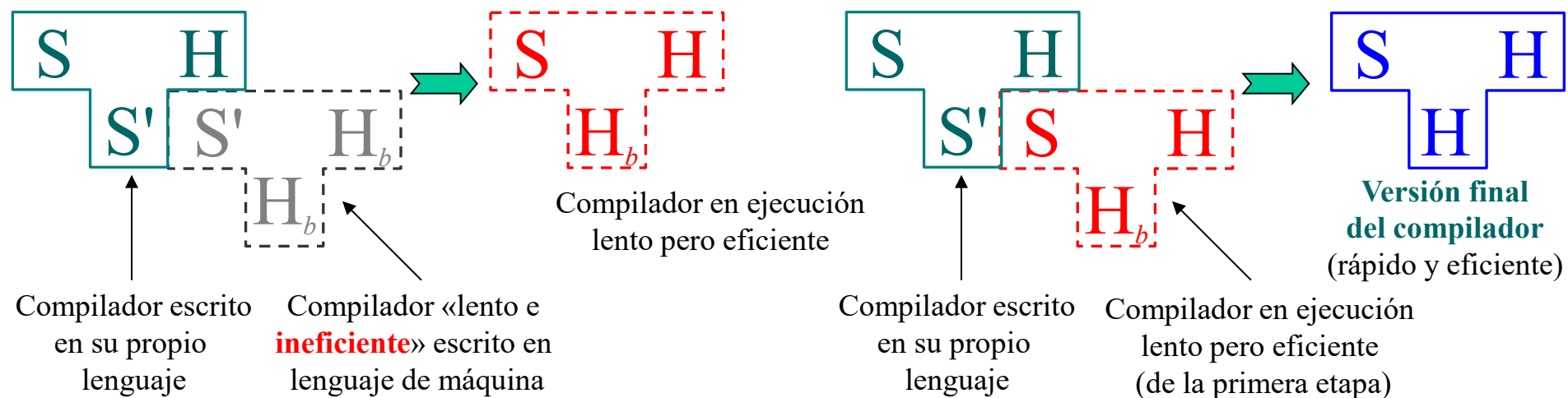
Nosotros codificamos el compilador gris de forma mas o menos sencilla siendo lento e ineficiente para un único uso. Así obtenemos un compilador intermedio que también será de usar y tirar.



$$S' \subseteq S$$

Para escribir el compilador no necesito todas las características del lenguaje

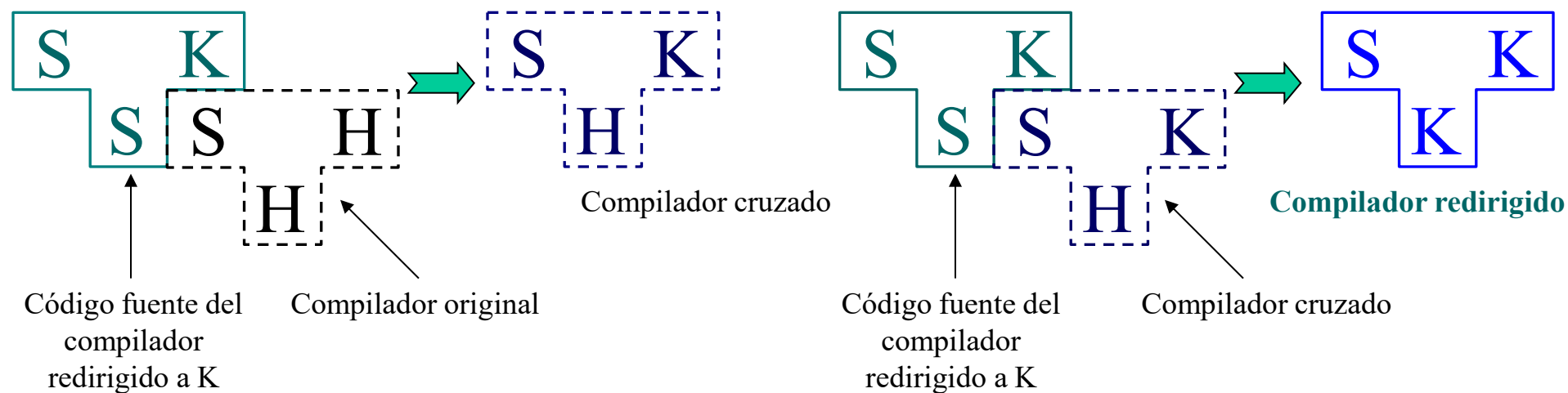
- Los pasos para su compilación son:
 - Escribir un compilador **lento e ineficiente**.
 - Utilizarlo para obtener un compilador ineficiente.
 - Compilar de nuevo usando el compilador ineficiente.



BOOTSTRAPPING

Arranque automático por transferencia 2

- Existe otra ventaja. Transportar ahora el compilador a una nueva computadora anfitrión solamente requiere que la etapa final del código fuente vuelva a escribirse para generar código para la nueva máquina.
- Éste se compila entonces utilizando el compilador antiguo para producir un compilador cruzado, y el compilador es nuevamente recompilado mediante el compilador cruzado para producir una versión de trabajo para la nueva máquina.



PORTING