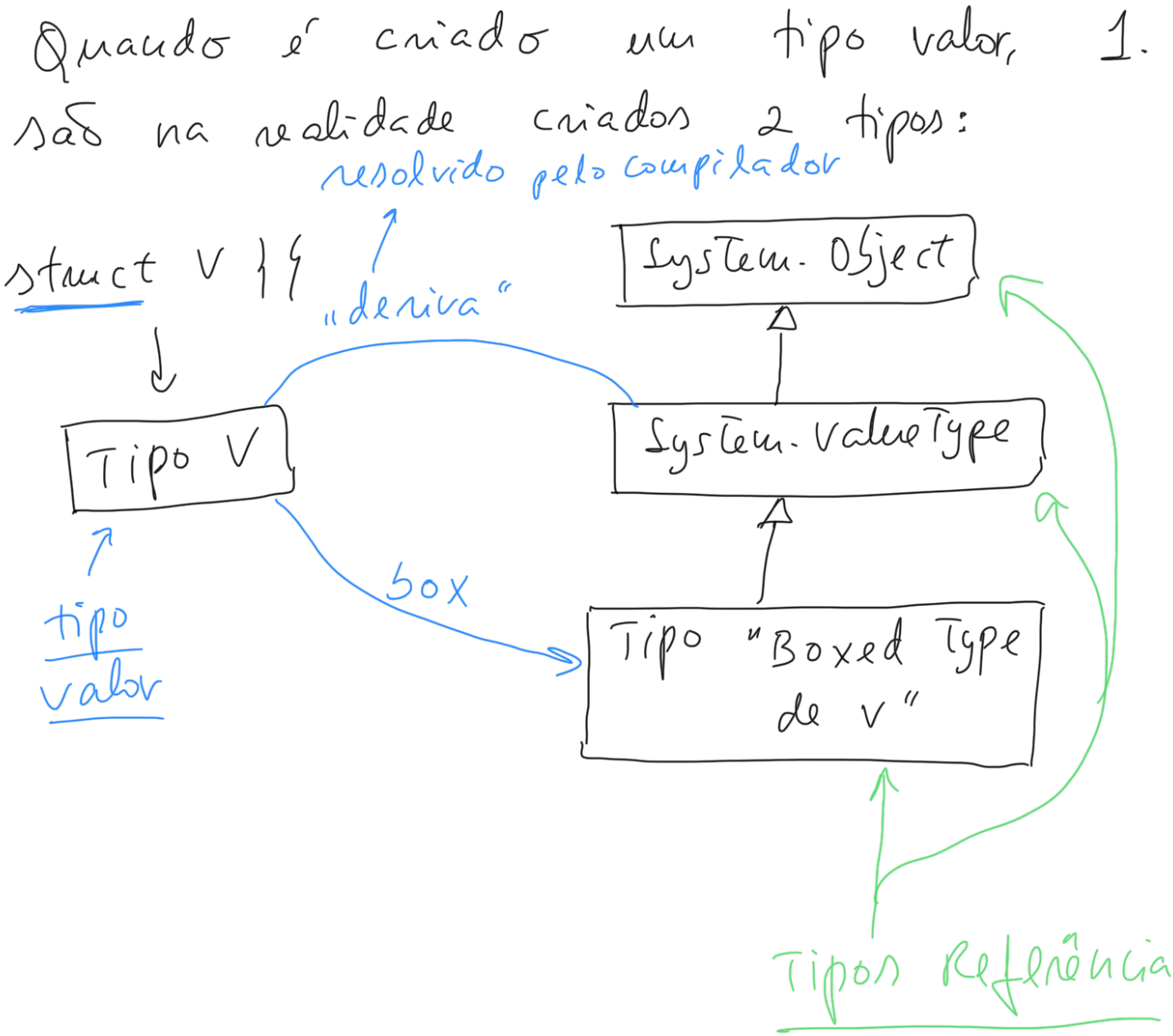
Conversão entre tipos valor e correspondente tipo referência: Box e unbox



Na realidade, os tipos valor não derivam fisicamente de System.ValueType; o seu "boxed type" é que deriva de System.ValueType.

O compilador/AVE associa o tipo valor ao seu "boxed type".

Ex:

```
int i = 5;
```

⇕

System.Int32

```
object o = i;   // Int32 → object ⇒ box
```

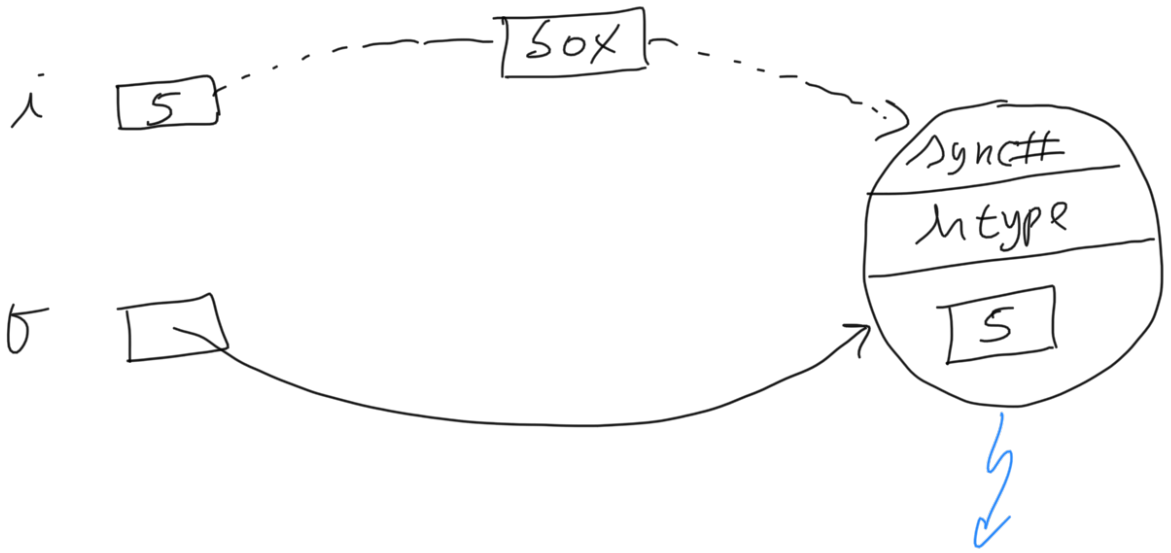
(VT: Int32; RT: object)

⇕

System.Object

STACK

HEAP



Instruções IL:

ldloc.0

box Int32

Tipo exato:

Boxed Type

de Int32

Box: Conversão de uma instância de VT para instância de RT

unbox: contrário; em C#, o unbox é conseguido fazendo no código um cast explícito para o tipo valor original daquela instância.

- Box é um operador de casting implícito; o compilador só aceita tipos compatíveis com o tipo valor:
  - object
  - System.ValueType
  - Qualquer interface implementada

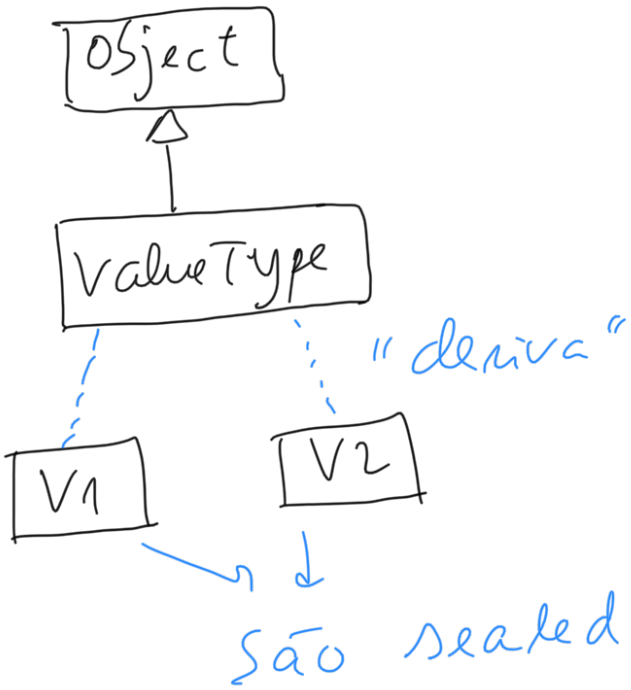
=> são todos RT

Ex:

```
interface IPrintable { ... }
struct V1 : IPrintable { int x; }
struct V2 : IPrintable { int y; }

class Prog {
    static void Print(V1 v) { ... }
    static void Print2(IPrintable p) { ... }

    main:
        V1 v1 = new V1(); // initobj
        V2 v2 = v1;       // ✗ Erro: V1 e V2 não são compatíveis entre si
        V2 v2 = new V2(); // initobj
        Prog.Print(v2);   // ✗
        Prog.Print(v1);   // ✓
        Prog.Print2(v2);  // ✓ → box
        Prog.Print2(v1);  // ✓ → box
```



Operação box

1. Recebe a instância (ldloc.0) de tipo valor a converter + token que identifica o tipo valor

2. Calcula e reserva espaço no heap para alojar o objeto = sizeof (campos) + header

3. Copia os valores dos campos do stack para o heap

4. retorna a referência para o objeto.

Unbox

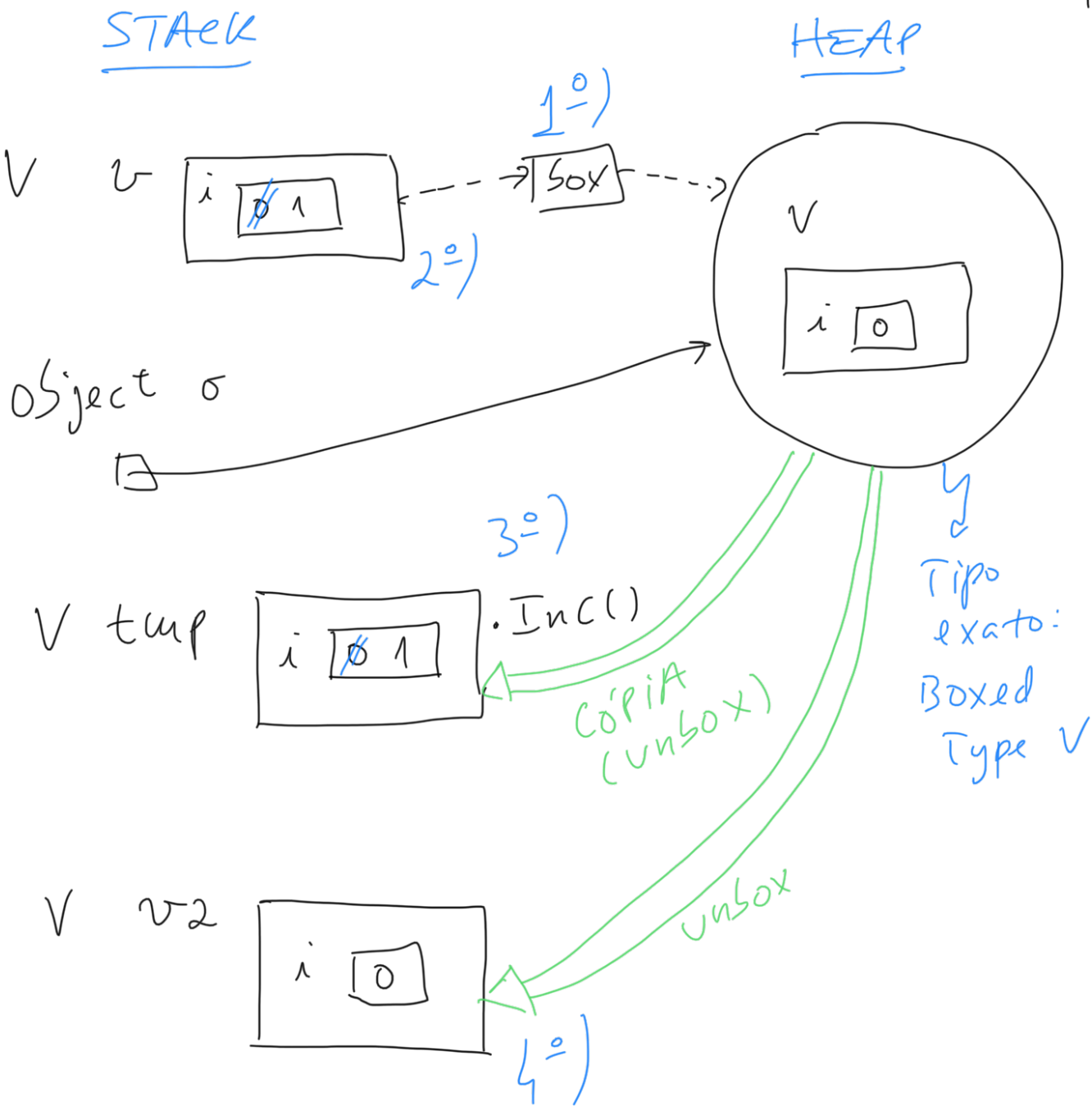
→ instrução IL unbox.any

- Ao contrário do box, não faz alojamento no heap
- obtém o ponteiro para os campos e copia os seus valores para o stack.

Normalmente, de seguida existe uma cópia por valor para uma variável local (stloc)

→ Um objeto (representação boxed) apenas pode ser unboxed para uma variável que tem o mesmo tipo que o valor boxed.
⇒ em caso de falha, lança InvalidCastException

```
EX: struct V {
        int i;
        void Inc() { ++i; }
    }
main:
    V v = new V();
    object o = v;  // box → 1°)
    v.i = 1;       → 2°)
    ((V)o).Inc();  // unbox → 3°)
    V v2 = (V)o;   // unbox → 4°)
    Console.Write(v2.i); // 0
```


STAck
HEAP
1º)
V v
i 0 1
Box
2º)
V
i 0
object o
3º)
V tmp
i 0 1
.InC()
Cópia (unBox)
Tipo exato: Boxed Type V
V v2
i 0
unBox
4º)