$A \to S\ Opt\ T\ Opt_1$

$Opt \to a \mid \varepsilon$

$Opt_1 \to b \mid \varepsilon$

$A \to S\ T$

$A \to S\ T\ b$

$A \to S\ a\ T$

$A \to S\ a\ T\ b$

00
01
10
11

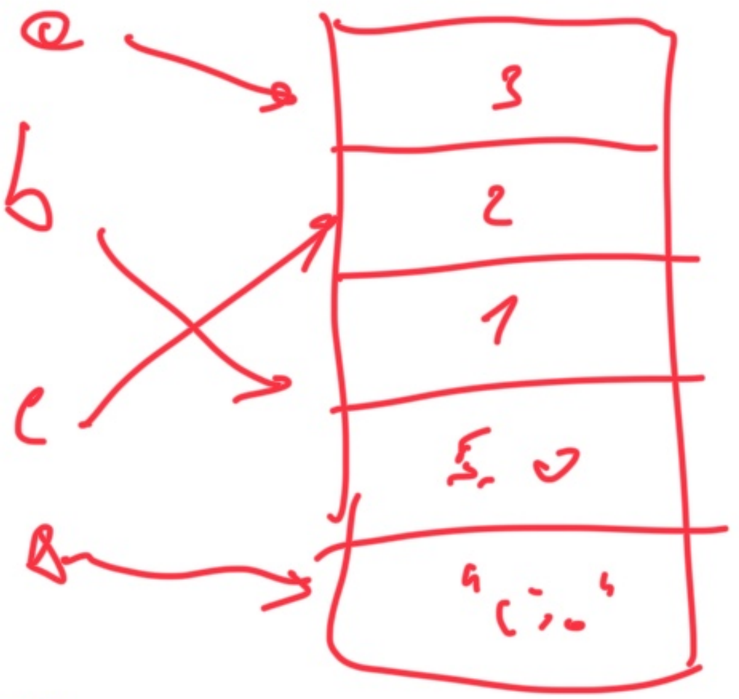
RISOLUZIONE DELLE PRODUZIONI VUOTE

ANALISI SEMANTICA

AMBIENTE

STATO

```
a = 2;
b = 1;
a = b;
```



L-VACUE
VARIABILE

R-VALVE
VALORE

1

NOME → VARIABILE ← VALORE

AMBIENTE

STATO

CALCOLO DEL TIPO DI a in B_4

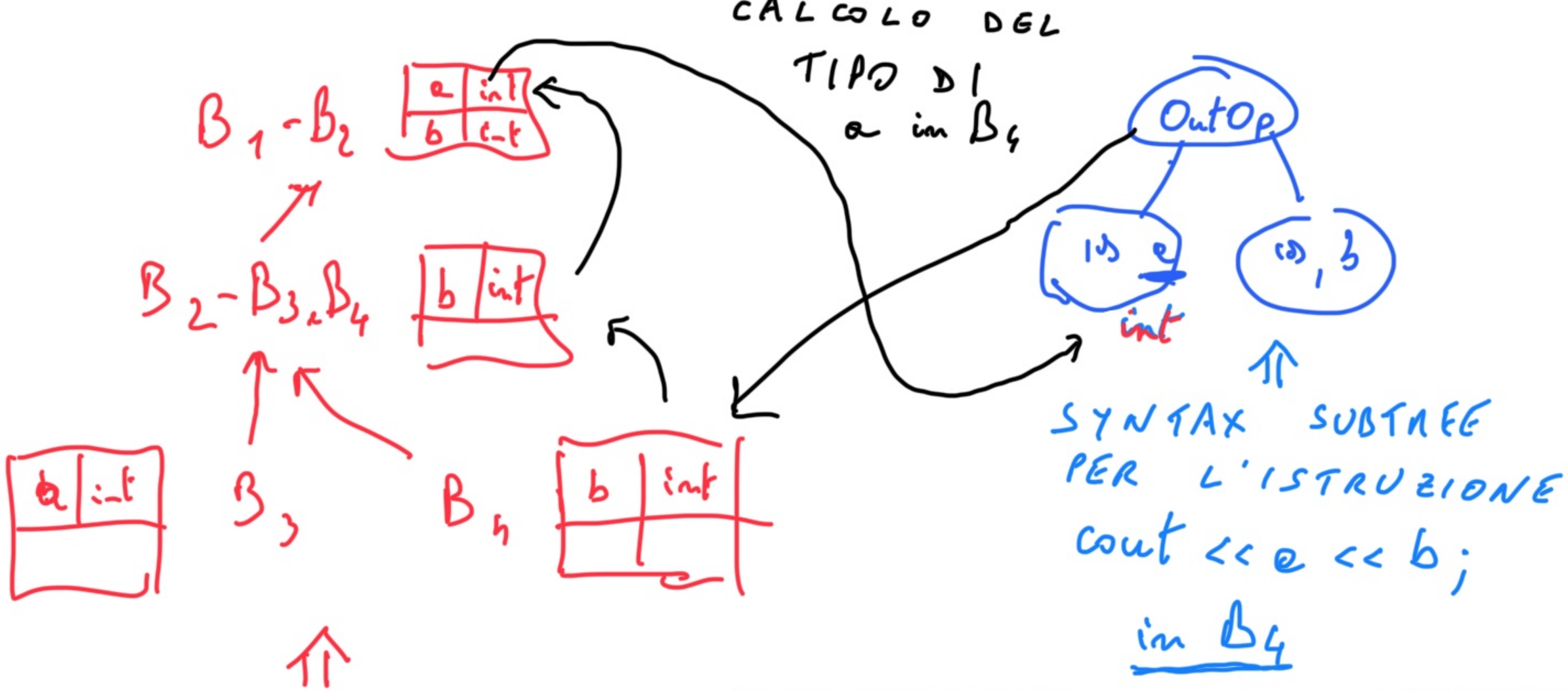
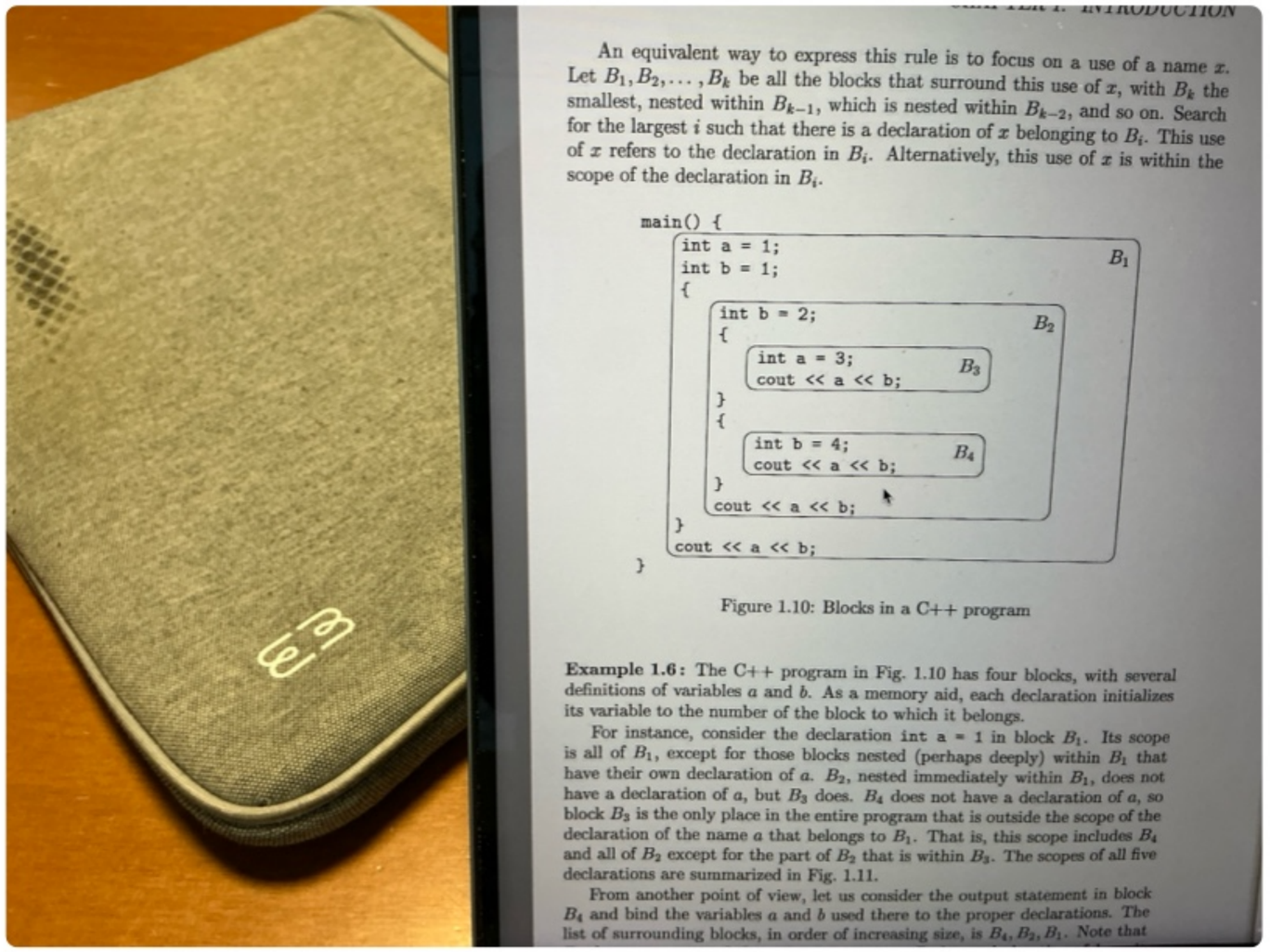


TABELLE DEI SIMBOLI ASSOCIATE AGLI SCOPING:

B_1-B_2, B_2-B_3,B_4, B_3, B_4



An equivalent way to express this rule is to focus on a use of a name x. Let $B_1, B_2, \ldots, B_k$ be all the blocks that surround this use of x, with B_k the smallest, nested within B_{k-1}, which is nested within B_{k-2}, and so on. Search for the largest i such that there is a declaration of x belonging to B_i. This use of x refers to the declaration in B_i. Alternatively, this use of x is within the scope of the declaration in B_i.

```
main() {
    int a = 1;                          B1
    int b = 1;
    {
        int b = 2;                      B2
        {
            int a = 3;                  B3
            cout << a << b;
        }
        {
            int b = 4;                  B4
            cout << a << b;
        }
        cout << a << b;
    }
    cout << a << b;
}
```

Figure 1.10: Blocks in a C++ program

Example 1.6: The C++ program in Fig. 1.10 has four blocks, with several definitions of variables a and b. As a memory aid, each declaration initializes its variable to the number of the block to which it belongs.

For instance, consider the declaration `int a = 1` in block B_1. Its scope is all of B_1, except for those blocks nested (perhaps deeply) within B_1 that have their own declaration of a. B_2, nested immediately within B_1, does not have a declaration of a, but B_3 does. B_4 does not have a declaration of a, so block B_3 is the only place in the entire program that is outside the scope of the declaration of the name a that belongs to B_1. That is, this scope includes B_4 and all of B_2 except for the part of B_2 that is within B_3. The scopes of all five declarations are summarized in Fig. 1.11.

From another point of view, let us consider the output statement in block B_4 and bind the variables a and b used there to the proper declarations. The list of surrounding blocks, in order of increasing size, is B_4, B_2, B_1. Note that