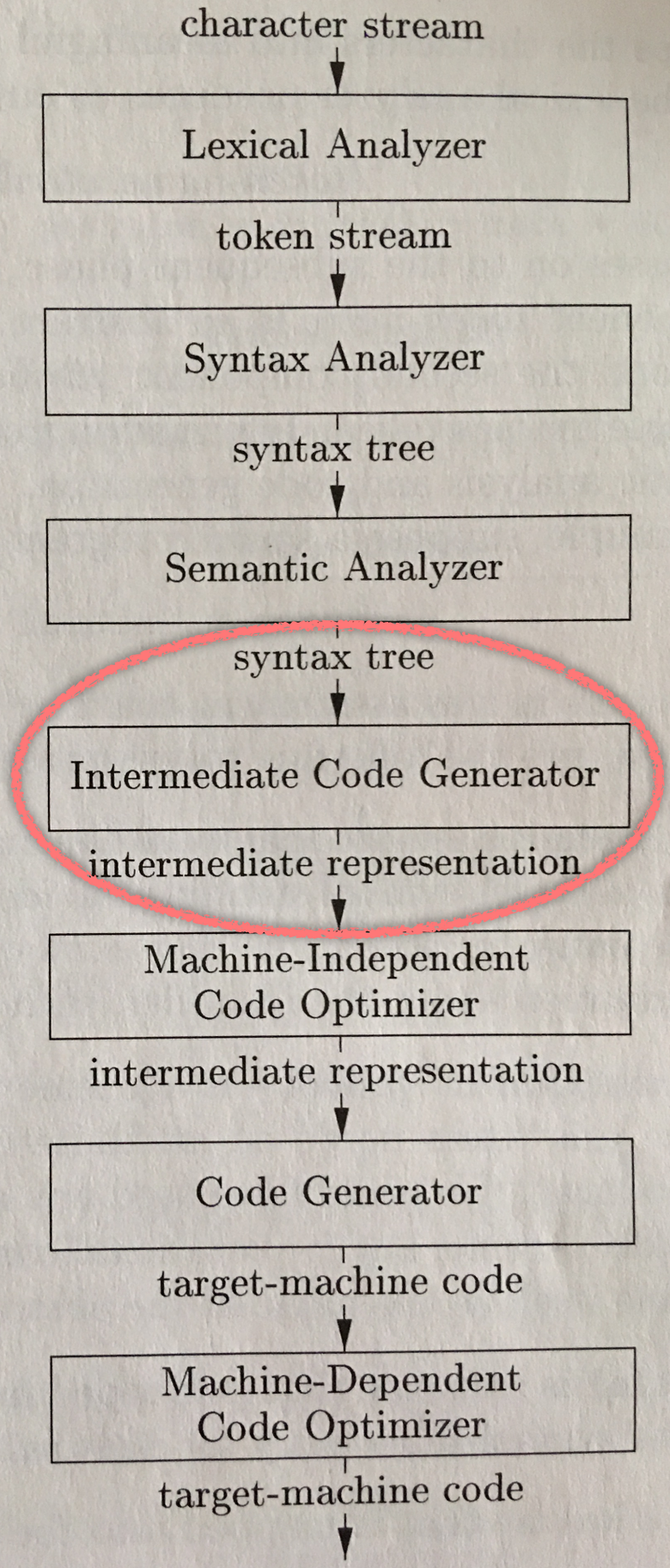
CSE443
Compilers

Dr. Carl Alphonce
alphonce@buffalo.edu
343 Davis Hall

Phases of a compiler

Intermediate Representation (IR): specification and generation

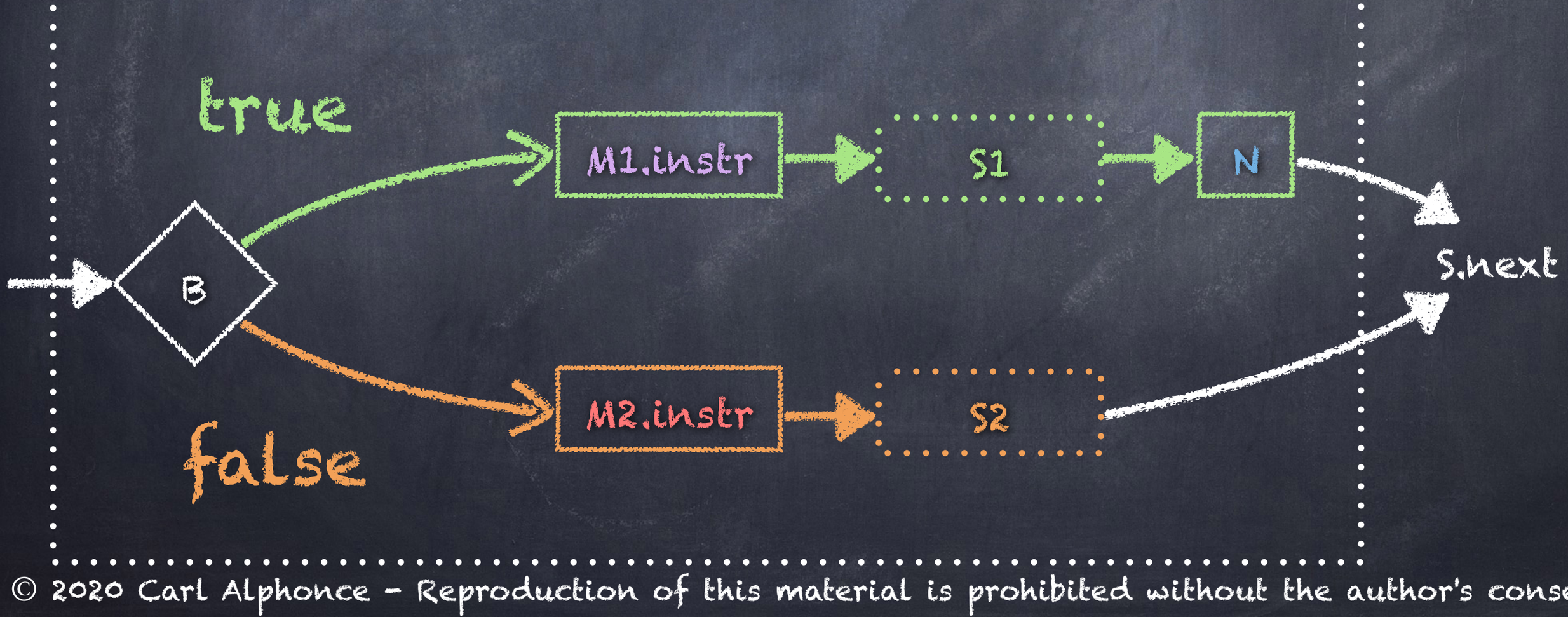
Figure 1.6, page 5 of text



backpatching if

6.7.3 Backpatching Flow-of-Control statements

S -> if (B) M1 S1 N else M2 S2	backpatch(B.truelist, M1.instr) backpatch(B.falselist, M2.instr) temp = merge(S1.nextlist, N.nextlist) S.nextlist = merge(temp, S2.nextlist)
M -> ε	M.instr = nextinstr
N -> ε	N.nextlist = makelist(nextinstr) gen('goto _')



Example 6.24 - extended

if (x < 100 || x > 200 && x != y) S1 else S2

Let's extend the Boolean expression example from part 1 by embedding that expression into an if-then-else statement (using the textbook syntax, not alpha syntax).

Example 6.24 - extended

if (x < 100 || x > 200 && x != y) S1 else S2

```
100: if x < 100 goto ____
101: goto 102
102: if x > 200 goto 104
103: goto ____
104: if x != y goto ____
105: goto ____
```

truelist = {100,104}
falselist = {103,105}

Let's remember where we left off...

Example 6.24 - extended

if (x < 100 || x > 200 && x != y) S1 else S2

```
100: if x < 100 goto ____
101: goto 102
102: if x > 200 goto 104
103: goto ____
104: if x != y goto ____
105: goto ____
106: instruction for S1
107: instruction for S1
108: instruction for S1
109: instruction for S1
110: instruction for S1
111: goto ____
112: instruction for S2
113: instruction for S2
114: instruction for S2
```

truelist = {100,104}
falselist = {103,105}

In the example above we have not spelled out what S1 and S2 are.

Let's assume S1 requires 5 instructions and S2 requires 3 instructions.

Example 6.24 - extended

if (x < 100 || x > 200 && x != y) S1 else S2

```
100: if x < 100 goto 106
101: goto 102
102: if x > 200 goto 104
103: goto 112
104: if x != y goto 106
105: goto 112
106: instruction for S1
107: instruction for S1
108: instruction for S1
109: instruction for S1
110: instruction for S1
111: goto ____
112: instruction for S2
113: instruction for S2
114: instruction for S2
115:
```

truelist = {100,104}

falselist = {103,105}

nextlist = {111}

Embedded in the context of this if-then-else statement we can backpatch truelist and falselist from the Boolean expression, and we introduce nextlist.

backpatching while

6.7.3 Backpatching Flow-of-Control statements

The end-of-rule actions for a **while statement** are shown on the next slide.

Exercise:
Extend example 6.24 as a while statement where the body of the while requires 5 instructions.

```
while (x < 100 || x > 200 && x != y) S1
```

Show how backpatching works in the instruction array.

6.7.3 Backpatching Flow-of-Control statements

S -> while M1 (B) M2 S1	backpatch(S1.nextlist, M1.instr) backpatch(B.truelist, M2.instr) S.nextlist = B.falselist gen('goto' M1.instr)
M -> ε	M.instr = nextinstr