**Example Efficient Code for Conditionals**

class Activity {

static int activityLevel = 0;

static boolean action(int signals, boolean objects, int smart, int pretty) {

if (smart + 2\*pretty > 10 && !(signals <= 5 && objects)) {

activityLevel++;

return true;

} else {

return false;

}

}

}

How would we compile this using the rules so far?

Compiles using javac into:

static boolean action(int, boolean, int, int);

Code:

0: iload\_2

1: iconst\_2

2: iload\_3

3: imul

4: iadd

5: bipush 10

7: if\_icmple 29

10: iload\_0

11: iconst\_5

12: if\_icmpgt 19

15: iload\_1

16: ifne 29

19: getstatic #2; //Field activityLevel:I

22: iconst\_1

23: iadd

24: putstatic #2; //Field activityLevel:I

27: iconst\_1

28: ireturn

29: iconst\_0

30: ireturn

Observations:

* this code peforms short-circuit evaluation, moreover:
* when ‘smart+2\*pretty > 10’ is false, code immediately ireturns ‘false’
* when ‘signals > 5’ is true, code immediately goes to ‘then’ part
* no intermediate result for if condition - do branches directly
* negation sign is eliminated and pushed through relations
* there is only one iconst\_0 and only one iconst\_1

How can we generate such pretty code?