

CLAIMS:

1. ~~—A computing device comprising a processor configured to:
—receive source code at a compiler, wherein the source code includes:
 at least one compound conditional having a plurality of conditions; and
——— for each condition of the plurality of conditions, a respective code block including one or more instructions to evaluate the condition;
—— for each ordering of a plurality of orderings of the plurality of conditions, determine that the ordering satisfies one or more legality constraints;
—— for each ordering of the plurality of orderings that satisfy the one or more legality constraints, determine a respective estimated computational cost for that ordering; and
—— reorder the plurality of conditions to have an ordering that has a lowest estimated computational cost of the plurality of orderings that satisfy the one or more legality constraints.~~
2. ~~—The computing device of claim 1, wherein the source code further includes:
 a first branch code block including one or more first branch evaluation instructions configured to be executed when the plurality of conditions are true; and
 a second branch code block including one or more second branch evaluation instructions configured to be executed when at least one condition of the plurality of conditions is false.~~
3. ~~—The computing device of claim 2, wherein the one or more legality constraints include a constraint that the source code includes one first branch code block and one second branch code block.~~
4. ~~—The computing device of claim 1, wherein the one or more legality constraints include a constraint that the respective code block of each condition includes an instruction to proceed to a common target code block.~~
5. ~~—The computing device of claim 4, wherein the one or more legality constraints include a constraint that a portion of the source code following a last code block of the respective code blocks of the plurality of conditions and preceding the common target code block has one entry point and one exit point.~~
6. ~~—The computing device of claim 1, wherein the one or more legality constraints include a constraint that no instruction calls an undefined variable.~~
7. ~~—The computing device of claim 1, wherein each code block is a basic block.~~
8. ~~—The computing device of claim 1, wherein at least one code block includes a plurality of basic blocks.~~

9. ~~The computing device of claim 1, wherein the processor is configured to determine the respective estimated computational cost for each ordering at least in part by:~~

~~— assigning an estimated cost score to each instruction included in the ordering;~~

~~and~~

~~— summing the respective estimated cost scores assigned to each instruction.~~

10. ~~The computing device of claim 1, wherein the compound conditional includes three or more conditions.~~

11. ~~A method performed by a processor of a computing device, the method comprising:~~

~~— receiving source code at a compiler, wherein the source code includes:~~

~~at least one compound conditional having a plurality of conditions; and~~

~~— for each condition of the plurality of conditions, a respective code block including one or more instructions to evaluate the condition;~~

~~— for each ordering of a plurality of orderings of the plurality of conditions, determining that the ordering satisfies one or more legality constraints;~~

~~— for each ordering of the plurality of orderings that satisfy the one or more legality constraints, determining a respective estimated computational cost for that ordering; and~~

~~— reordering the plurality of conditions to have an ordering that has a lowest estimated computational cost of the plurality of orderings that satisfy the one or more legality constraints.~~

12. ~~The method of claim 11, wherein the source code further includes:~~

~~a first branch code block including one or more first branch evaluation instructions configured to be executed when the plurality of conditions are true; and~~

~~a second branch code block including one or more second branch evaluation instructions configured to be executed when at least one condition of the plurality of conditions is false.~~

13. ~~The method of claim 11, wherein the one or more legality constraints include a constraint that the respective code block of each condition includes an instruction to proceed to a common target code block.~~

14. ~~The method of claim 11, wherein the one or more legality constraints include a constraint that no instruction calls an undefined variable.~~

15. ~~The method of claim 11, wherein determining the respective estimated computational cost for each ordering includes:~~

~~— assigning an estimated cost score to each instruction included in the ordering;~~
and
~~— summing the respective estimated cost scores assigned to each instruction.~~

1. A computing device comprising a processor configured to:

during an execution of a compiler;

receive source code at the compiler, wherein the source code includes:

at least one compound conditional having a plurality of conditions
configured to be evaluated as part of execution of assembly code compiled
from the source code to determine which of a plurality of branch code
blocks of the assembly code are to be executed as part of the execution of
the assembly code, wherein each condition of the plurality of conditions
includes a code block that includes one or more instructions to evaluate the
condition as part of execution of the assembly code;

determine a plurality of orderings of the conditions included in the
compound conditional;

determine that two or more orderings of the plurality of orderings of
the plurality of conditions satisfy one or more legality constraints on a
syntactic structure of the one or more instructions of each of the plurality
of conditions, and that at least one ordering of the plurality of orderings
does not satisfy the one or more legality constraints;

only for each ordering of the plurality of orderings that satisfy the one
or more legality constraints, during the execution of the compiler and
subsequently to determining that the ordering satisfies the one or more
legality constraints, determine a respective estimated computational cost
for that ordering without determining at least one respective computational
cost for the at least one ordering that does not satisfy the one or more
legality constraints;

wherein the at least one respective computational cost for each
ordering of the plurality of orderings that satisfy the one or more legality
constraints is determined at least in part by:

assigning an estimated cost score to each instruction included in the
ordering as specified by an estimated computational cost table that maps a
plurality of instruction types to a respective plurality of estimated cost
scores, and

summing the respective estimated cost scores assigned to each instruction;

reorder the plurality of conditions to have an ordering that has a lowest estimated computational cost of the plurality of orderings that satisfy the one or more legality constraints; and

compile the source code with the reordered plurality of conditions into the assembly code.

2. The computing device of claim 1, wherein the source code further includes: a first branch code block including one or more first branch evaluation instructions configured to be executed when the plurality of conditions are true; and a second branch code block including one or more second branch evaluation instructions configured to be executed when at least one condition of the plurality of conditions is false.

3. The computing device of claim 2, wherein the one or more legality constraints include a constraint that the source code includes one first branch code block and one second branch code block.

4. The computing device of claim 1, wherein the one or more legality constraints include a constraint that the respective code block of each condition includes an instruction to proceed to a common target code block.

5. The computing device of claim 4, wherein the one or more legality constraints include a constraint that a portion of the source code that follows a last code block of the respective code blocks of the plurality of conditions and that precedes the common target code block has one entry point and one exit point.

6. The computing device of claim 1, wherein the one or more legality constraints include a constraint that no instruction calls an undefined variable.

7. The computing device of claim 1, wherein each code block is a basic block.

8. The computing device of claim 1, wherein at least one code block includes a plurality of basic blocks.

9. The computing device of claim 1, wherein the compound conditional and the plurality of conditions included in the compound conditional are each configured to be evaluated to respective Boolean values when the assembly code is executed.

10. The computing device of claim 1, wherein each of the plurality of instruction types is a respective type of operation, and where each type of operation has an associated estimated integer cost and an estimated floating-point cost, the estimated integer cost being an estimated cost score of the type of operation when performed using integer computation, and the estimated floating-point cost being an

estimated cost score of the type of operation when performed using floating point computation.

11. The computing device of claim 10, wherein the types of operations include addition, subtraction, comparison, negation, division, remainder, AND, OR, XOR, minimum, maximum, load, and/or store.

12. A method performed by a processor of a computing device, the method comprising:

during an execution of a compiler: receiving source code at the compiler, wherein the source code includes:

at least one compound conditional having a plurality of conditions configured to be evaluated as part of execution of assembly code compiled from the source code to determine which of a plurality of branch code blocks of the assembly code are to be executed as part of the execution of the assembly code,

wherein each condition of the plurality of conditions includes a code block that includes one or more instructions to evaluate the condition as part of execution of the assembly code;

determining a plurality of orderings of the conditions included in the compound conditional;

determining that two or more orderings of the plurality of orderings of the plurality of conditions satisfy one or more legality constraints on a syntactic structure of the one or more instructions of each of the plurality of conditions, and that at least one ordering of the plurality of orderings does not satisfy the one or more legality constraints;

only for each ordering of the plurality of orderings that satisfy the one or more legality constraints, during the execution of the compiler and subsequently to determining that the ordering satisfies the one or more legality constraints, determining a respective estimated computational cost for that ordering without determining at least one respective computational cost for the at least one ordering that does not satisfy the one or more legality constraints,

wherein determining the at least one respective computational cost for each ordering of the plurality of orderings that satisfy the one or more legality constraints includes:

assigning an estimated cost score to each instruction included in the ordering as specified by an estimated computational cost table that maps a plurality of instruction types to a respective plurality of estimated cost scores, and

summing the respective estimated cost scores assigned to each instruction;

reordering the plurality of conditions to have an ordering that has a lowest estimated computational cost of the plurality of orderings that satisfy the one or more legality constraints; and

compiling the source code with the reordered plurality of conditions into the assembly code.

13. The method of claim 12, wherein the source code further includes:

a first branch code block including one or more first branch evaluation instructions configured to be executed when the plurality of conditions are true; and

a second branch code block including one or more second branch evaluation instructions configured to be executed when at least one condition of the plurality of conditions is false.

14. The method of claim 13, wherein the one or more legality constraints include a constraint that the source code includes one first branch code block and one second branch code block.

15. The method of claim 12, wherein the one or more legality constraints include a constraint that the respective code block of each condition includes an instruction to proceed to a common target code block.

16. The method of claim 12, wherein the one or more legality constraints include a constraint that a portion of the source code that follows a last code block of the respective code blocks of the plurality of conditions and that precedes the common target code block has one entry point and one exit point.

17. The method of claim 12, wherein the one or more legality constraints include a constraint that no instruction calls an undefined variable.

18. The method of claim 12, wherein each code block is a basic block.

19. The method of claim 12, wherein at least one code block includes a plurality of basic blocks.

20. The method of claim 12, wherein each of the plurality of instruction types is a respective type of operation, and where each type of operation has an associated estimated integer cost and an estimated floating-point cost, the estimated integer cost being an estimated cost score of the type of operation when performed using integer computation, and the estimated floating-point cost being an estimated cost score of the type of operation when performed using floating point computation.

21. The method of claim 20, wherein the types of operations include addition, subtraction, comparison, negation, division, remainder, AND, OR, XOR, minimum, maximum, load, and/or store.

22. A computing device comprising a processor configured to:
during an execution of a compiler:
receive source code at the compiler, wherein the source code includes a
plurality of code blocks that are included in a compound conditional and are
configured to be evaluated as part of execution of assembly code compiled from the
source code to determine which of a plurality of branch code blocks of the assembly
code are to be executed as part of the execution of the assembly code;
determine a plurality of orderings of the conditions included in the compound
conditional;
determine that two or more orderings of the plurality of orderings of the
plurality of code blocks satisfy one or more legality constraints on a syntactic
structure of one or more instructions of each of the plurality of code blocks, and that
at least one ordering of the plurality of orderings does not satisfy the one or more
legality constraints;
only for each ordering of the plurality of orderings that satisfy the one or more
legality constraints, during the execution of the compiler and subsequently to
determining that the ordering satisfies the one or more legality constraints, determine
a respective estimated computational cost for that ordering without determining at
least one respective computational cost for the at least one ordering that does not
satisfy the one or more legality constraints,
wherein the at least one respective computational cost for each ordering of the
plurality of orderings that satisfy the one or more legality constraints is determined at
least in part by:
assigning an estimated cost score to each instruction included in the ordering
as specified by an estimated computational cost table that maps a plurality of
instruction types to a respective plurality of estimated cost scores, and
summing the respective estimated cost scores assigned to each instruction;
reorder the plurality of code blocks to have an ordering that has a lowest
estimated computational cost of the plurality of orderings that satisfy the one or more
legality constraints; and
compile the source code with the reordered plurality of conditions into the
assembly code.