This is an open-book, open-notes, open-laptop, closed-network exam. Please do not post anything on Piazza until the solutions are put up on the class website.

You have 3 hours to work on this exam. The examination has 7 problems worth 165 points. Please budget your time accordingly. Write your answers in the space provided on the exam. If you use additional scratch paper, please turn that in as well.

Your Name: ____________________ SUNet ID: ____________________

The following is a statement of the Stanford University Honor Code:

1. The Honor Code is an undertaking of the students, individually and collectively:

   (a) that they will not give or receive aid in examinations; that they will not give or receive unpermitted aid in class work, in the preparation of reports, or in any other work that is to be used by the instructor as the basis of grading;

   (b) that they will do their share and take an active part in seeing to it that others as well as themselves uphold the spirit and letter of the Honor Code.

2. The faculty on its part manifests its confidence in the honor of its students by refraining from proctoring examinations and from taking unusual and unreasonable precautions to prevent the forms of dishonesty mentioned above. The faculty will also avoid, as far as practicable, academic procedures that create temptations to violate the Honor Code.

3. While the faculty alone has the right and obligation to set academic requirements, the students and faculty will work together to establish optimal conditions for honorable academic work.

Signature: ____________________

<table>
<thead>
<tr>
<th>Problem</th>
<th>#1</th>
<th>#2</th>
<th>#3</th>
<th>#4</th>
<th>#5</th>
<th>#6</th>
<th>#7</th>
<th>Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>Score</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>30</td>
<td>20</td>
<td>20</td>
<td>20</td>
<td>25</td>
<td>25</td>
<td>25</td>
<td>165</td>
</tr>
</tbody>
</table>
Problem 1. True or false. Specify clearly whether the following statements are true or false. You must justify your answers in 1 to 5 lines [30 Points].

1. Global scheduling, as discussed in class, can potentially slow a program down.

2. Data-flow in Datalog
   (a) We can formulate the reaching definitions problem using Datalog.

   (b) If the answer is true above, then if we use a BDD implementation of Datalog to find reaching definitions, it will run faster than the data flow algorithm described in class that iterates the graph in reverse postorder.

3. For each of the metric below, is it true or false that generational garbage collection improves over the Baker’s algorithm:
   (a) Space reclaimed

   (b) Overall execution time
(c) Space usage

(d) Pause time

(e) Data locality

4. Different depth-first traversals of a reducible flow graph may yield different retreating edges.

5. For the expression \((\neg x_1 \ OR \ x_2) \ AND \ (x_1 \ XOR \ x_2)\), it is possible to create a BDD with just 4 nodes (including the 0 and 1 nodes).
**Problem 2.** Pointer Analysis [20 points]

Perform a context-insensitive, flow-insensitive pointer analysis on the code below. List the hpts tuples that are produced by the analysis.

```java
public class A {
    public A f;
    public void foo(A p, A q) {
        p.f = q;
    }
    public void bar() {
        this.f = new A(); // h1
    }
}

public class Main {
    public static void main(String[] args) {
        A a = new A(); // h2
        A b = new A(); // h3
        b.bar();
        a.foo(a, b);
        b = b.f;
    }
}
```
Problem 3. Partial Redundancy Elimination [20 points]
Extra space for PRE answer:
Problem 4. Software Pipelining [20 points]

Consider the following data dependency graph. Next to each instruction are indicated the resources used by that instruction.

1. What is the minimum initiation interval from the precedence constraints?

2. What is the minimum initiation interval from resource usage?
3. Find the optimal software pipelined schedule for this loop. Show both the modulo reservation table and the generated code (with nops if necessary) for a single iteration of the loop.

4. Can this schedule be generated using the heuristic algorithm described in class? Why or why not?
Problem 5. Move Semantics [25 points]

Some programming languages offer special “move” instructions to support transferring of resource ownership between objects or pointers. An example is the `unique_ptr` (unique pointer) in C++. We say that a `unique_ptr` is valid if it points to a resource, and invalid otherwise. At any given time, there is one and only one (valid) `unique_ptr` that refers to a resource. All `unique_ptr` pointers are invalid at the beginning of the program. A make instruction creates and returns a resource that can be assigned to a pointer. A move instruction transfers the ownership from the source to the destination pointer; the source `unique_ptr` can no longer be used, as the move renders it invalid. When the pointer is overwritten or goes out of scope, if it holds a valid resource, the resource is automatically destroyed by the run-time system.

There are two core instructions that the programmer writes:

- \( x = \text{make}(\text{Resource}()) \);
  creates a resource object and gives ownership to the `unique_ptr` \( x \).

- \( y = \text{move}(x) \);
  moves the underlying resource out of the `unique_ptr` \( x \) and into the `unique_ptr` \( y \).

The compiler is responsible for inserting instructions to destroy resources. There are two versions of such instructions:

- \( \text{destroy}_\text{if_valid}(x) \);
  destroys the underlying resource if \( x \) is valid.

- \( \text{destroy}(x) \);
  destroys the underlying resource unconditionally, expecting \( x \) to be valid.

Automatic destruction is implemented with a straightforward compiler pass that inserts a `destroy_if_valid(x)` instruction:

1. right before any \( x = \text{make}(\text{Resource}()) \) instruction
2. right before any \( x = \text{move}(y) \) instruction, where \( x \neq y \)
3. right before variable \( x \) goes out of scope.

You have two tasks:

1. Every \( x = \text{move}(y) \) instruction requires \( y \) to be valid. Warn if it is possible that \( y \) may be invalid.
2. Optimize the destruction of the resources by removing unnecessary `destroy_if_valid(x)` instructions and by replacing `destroy_if_valid(x)` instructions with the faster version `destroy(x)`, whenever possible.
For example:

```cpp
destroy_if_valid(a); // REMOVE, as a is not valid
a = make(Resource());
destroy_if_valid(b); // REMOVE, as b is not valid
b = make(Resource());
destroy_if_valid(b); // CHANGE to destroy(b), as b is valid
b = move(a);
destroy_if_valid(c); // REMOVE, as c is not valid
c = move(a); // WARN HERE, as a is invalid
```

Describe your data-flow analysis by specifying the following:

- Direction of your data-flow analysis (forward/backward)
- Elements in the semi-lattice and their meaning
- Meet operator or lattice diagram
- Top and bottom elements if they exist
- Transfer function
- Boundary condition initialization
- Interior points initialization

Make sure you specify how you warn on uses of invalid objects and perform the `destroy_if_valid(x)` replacements using your analysis results.

Write your data-flow analysis answer here (may use the next page as well):
Continue your data-flow analysis answer here (if needed):
Problem 6. Parallelization [25 points]

Consider the following program.

```c
for (int i = 1; i < N; i++) {
    A[i, i] = B[i, i];  \hspace{1cm} (1)
    for (int j = i + 1; j < N; j++) {
        B[i, j] = (B[i-1, j] + B[i, 0]) * A[i, j-1]; \hspace{1cm} (2)
    }
}
```

1. Draw the iteration space for each statement in the program. Use arrows to mark data-dependencies between iterations.

2. Is there any communication-free parallelism that can be obtained using affine transformations? If yes, what is the affine partitioning for statement (1), and for statement (2)?
3. Assuming $p$ is the processor ID, what is the range of $p$? What is the SPMD code for each processor? You do not need to optimize the code generated; simply wrap statements with the right conditional statements.
Problem 7. Path sensitive analysis with SMT [25 points]

Consider the following function.

```c
1 double foo( double a, double b ) {
2     double c, d;
3     if ( b > 0 ){
4         c = 2 * a - 3 * b;
5         if ( c > b ){
6             d = c;
7         } else {
8             d = 0;
9         }
10         return c - d;
11     }
12     return 0;
13 }
```

1. Rewrite the program in SSA form.
2. Consider the following property $P$: the return value in line 10 must be less than the first input to the function (double $a$). Write an SMT formula to check whether this function has this property. You do not need to solve the formula or check whether the function actually has this property.

For simplicity, use Real to model double. First introduce the variables using declare-const (e.g., (declare-const $a_0$ Real)). Then write all the assertions for checking the property (e.g., (assert (= phi0 (> b0 0))))

You will not be deducted points for minor syntax errors.

3. Suppose CVC4 returns $R$ on this formula, where $R \in \{\text{SAT, UNSAT}\}$. Describe how you can use $R$ to determine whether the function has property $P$?