

Instructions

- Problems 1-4 are each worth 25 points.
- You may discuss the questions with the TAs and others in the class but your answers must be in your own words. You must not copy someone else's solutions. If you consult others or use external sources, please cite the people or sources used in your answers.
- Solutions to these problems will be posted on Courseworks on April 17, 2013.
- This assignment may be submitted electronically on Courseworks by 2:40pm, April 17, 2013 for 50% credit.
- Pdf files are preferred.

Problems

1. Consider the syntax-directed definitions in Figs. 6.19, 6.36 and 6.37 in ALSU for expressions, if-statements and booleans. Assume the boolean operators `&&`, `||`, and `!=` have the customary associativities and precedences.
 - a. Construct a parse tree for the C-like if-statement

```
if( x < 10 && x > 20 || x != y ) x = 30;
```
 - b. Show the values of all the attributes computed at each node in the parse tree by these SDDs.
 - c. Show the three-address code produced for this if-statement.
 - d. Can you see any ways in which the three-address code can be optimized?
2. Write pseudocode for a function `sequiv(exp1, exp2)` that will test the structural equivalence of two type expressions `exp1` and `exp2`. Show how your function computes `sequiv(array(2, array(2, int)), array(2, array(3, int)))`.

3. Let `fib(n)` be the function

```
int fib(n) {  
    if (n == 0)  
        return 0;  
    else if (n == 1)  
        return 1;  
    else  
        return fib(n-1) + fib(n-2);  
}
```

- a. Show the activation tree for `fib(3)`.
 - b. Show the activation records that are on the run-time stack when `fib(1)` is invoked for the first time. Just show the return value, actual parameters, and the caller's frame pointer in each activation record.
4. Give an example from some common programming languages to illustrate the difference between
 - a. Normal-order evaluation and applicative-order evaluation.
 - b. Call by reference and call by value.
 - c. Lexical scope and dynamic scope.
 - d. Stack allocation and heap allocation.
 - e. A static type and a dynamic type.You can use different programming languages for each part.
-