

Compilers [Spring 2022] Test III

NAME: _____

Instructions:

- 1) This test is 10 pages in length.
- 2) You have 120 minutes to complete and turn in this test.
- 3) Short-answer and essay questions will be graded on how clearly you've communicated the necessary ideas. Respond in complete English sentences. Avoid using bullet points and enumerated lists. Feel free to use abbreviations and notations used in class.
- 4) This test is closed books, notes, papers, friends, phones, neighbors, smartwatches, etc.
- 5) Use the backs of pages in this test packet for scratch work. If you write more than a final answer in the area next to a question, circle your final answer.
- 6) Write and sign the following: "I pledge my Honor that I have not cheated, and will not cheat, on this test."

Signed: _____

1. [6 points]

What are all the kinds of data a complete djc stores on the program stack? [1 sentence]

2. [8 points]

a) How would heap memory normally be partitioned between generations in a generational GC, and why? [Short essay]

b) Besides the partitioning of memory into generations, how else would a generational GC typically be parameterized, and what would be common values for those parameters? [1-2 sentences]

3. [14 points]

G is: $S \rightarrow A\$$ $A \rightarrow 0A0 \mid \epsilon$

a) Considering all the LL sets of grammars we discussed in class, prove which ones (if any) contain G.

G is: $S \rightarrow A\$$ $A \rightarrow 0A0 \mid \epsilon$

b) Show (i) a GLR parse table for **G** and (ii) a GLR parse trace of input $00\$$ according to **G**. If you arrive at a point in the parse trace where the trace cannot continue, indicate the problem and stop the trace at that point.

4. [10 points]

Prove that the halting problem is undecidable, at the level of detail presented in class.

5. [62 points] [Essay] [Your response may continue on to the next 4 pages.]

Let's add 3 new features to DJ:

1. Multi-argument methods, e.g., $f(a,b,c)$ or $f()$.
2. Instanceof expressions, which are used to dynamically test whether an object is an instance of a given class. The format is
 $e \text{ instanceof } C$
where e is an expression and C a class name; the whole instanceof expression evaluates to true (i.e., a nonzero nat) iff e evaluates to an object whose type is a subtype of C .
3. Switch expressions. The format is
 $\text{switch}(e) \{ \text{case } n_1:e_1 \text{ case } n_2:e_2 \dots \text{case } n_i:e_i \text{ default}:e_d \}$
where e is an expression, all the n 's are distinct natural numbers, all the e_i 's are (nonempty) expression lists, and i is a natural number.

How should djc be modified, to implement these new features? Assume DISMs have arbitrarily much code memory available. Use pseudocode when convenient but be specific. More efficient implementations, and more efficient target code, will receive more points.

(This page provides extra space for Problem 5.)

(This page provides extra space for Problem 5.)

(This page provides extra space for Problem 5.)

(This page provides extra space for Problem 5.)

Undergraduates stop here. The remaining problem is for graduate students.

6. [7 points]

Prove that all left-recursive grammars are outside of LL(1).