Semantic Analysis

1. What determines whether a language rule is a matter of syntax or of static semantics?

2. Why is it impossible to detect certain program errors at compile time, even though they can be detected at run time?

3. What is an attribute grammar?

4. What are programming assertions? What is their purpose?

5. What is the difference between synthesized and inherited attributes?

6. Give two examples of information that is typically passed through inherited attributes.

7. What is attribute flow?

8. What is a one-pass compiler?

9. What does it mean for an trribute grammar to be S-attributed? L-attributed? Noncircular? What is the significance of these grammar classes?

10. What is the difference between a semantic function and an action routine?

11. Why can't action routines be placed at arbirary locations within the right-hand side of productions in an `LR` CFG?

12. What patterns of attribute flow can be captured easily with action routines?

13. Some compilers perform all semantic checks and intermediate code generation in action routines. Others use action routines to build a syntax tree and then perform checks and intermediate code generation in separate traversals of the syntax tree. Discuss the tradeoffs between these two strategies.

14. What sort of information do action routines typically keep in global variables, rather than in attributes?

15. Describe the similarites and differences between context-free grammars and tree grammars.

16. How can a semantic analyzer avoid the generation of cascading errors messages?

17. Explain how to manage space for synthesized attributes in a bottom-up parser.

18. Explain how to manage space for inerited attributes in a bottom-up parser.

19. Define left corner and trailing part.
vskip 1cm
20. Under what circumstances can an action routine be embedded in the right-hand side of a production in a bottom-up parser? Equivalently, under what circumstanes can a marker symbol be embedded in a right-hand side without rendering the grammar non-LR?

21. Summarize the tradeoffs between the ad hoc management of space for attributes in a top-down parser.

22. At any given point in a top-down parse, which symbols wil have attribute records in an automatically managed attribute stack?