

Question 1: Lisp text questions

Question 2.2

EXERCISES

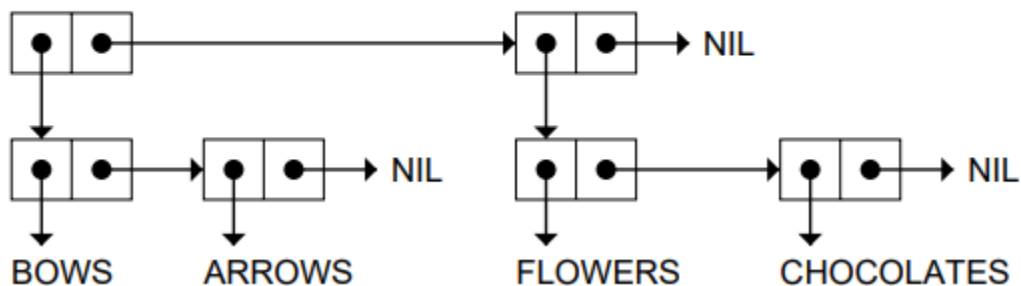
2.2. Which of these are well-formed lists? That is, which ones have properly balanced parentheses?

1. (A B (C))
2. ((A) (B))
3. A B)(C D)
4. (A (B (C)))
5. (A (B (C)))
6. (((A) (B)) (C))

The second, fifth, and sixth are well-formed lists.

Question 2.4

2.4. What is the parenthesis notation for this cons cell structure?

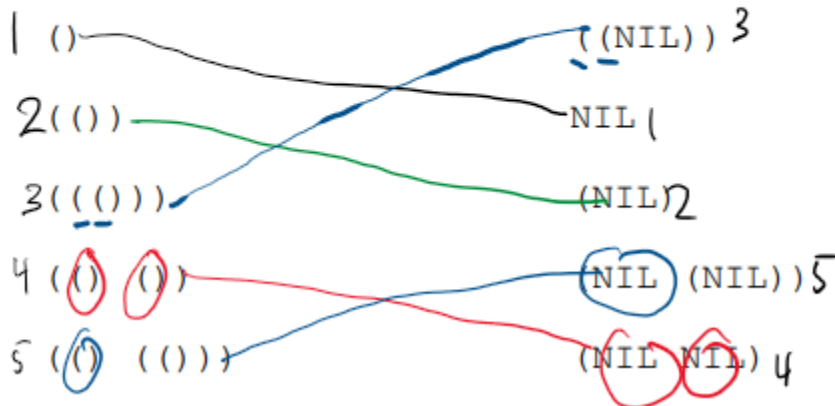


((BOWS ARROWS) (FLOWERS CHOCOLATES))

Question 2.6

EXERCISE

2.6. Match each list on the left with a corresponding list on the right by substituting NIL for () wherever possible. Pay careful attention to levels of parenthesization.



() - NIL | (()) - (NIL) | (((())) - ((NIL)) | ((() ())) - (NIL NIL) | ((() ())) - (NIL (NIL))

Question 2.13

Let's build a table of the steps to follow to get to PYRAMID:

<u>Step</u>	<u>Result</u>
start	((BLUE CUBE) (RED PYRAMID))
C...DR	((RED PYRAMID))
C..ADR	(RED PYRAMID)
C.DADR	(PYRAMID)
CADADR	PYRAMID

EXERCISES

2.13. Write down tables similar to the one above to illustrate how to get to each word in the list (((FUN)) (IN THE) (SUN)).

First target: **FUN**

Start: (((FUN)) (IN THE) (SUN))

C..AR: ((FUN))

C.AAR: (FUN)

CAAAR: FUN

Next: **IN**

Start: (((FUN)) (IN THE) (SUN))

C..DR: (IN THE)

C.ADR: IN THE

CAADR: IN

Next: **THE**

Start: (((FUN)) (IN THE) (SUN))

C...DR: ((IN THE))

C..ADR: (IN THE)

C.DADR: (THE)

CADADR: THE

Next: **SUN**

Start: (((FUN)) (IN THE) (SUN))

C...DR: ((IN THE))

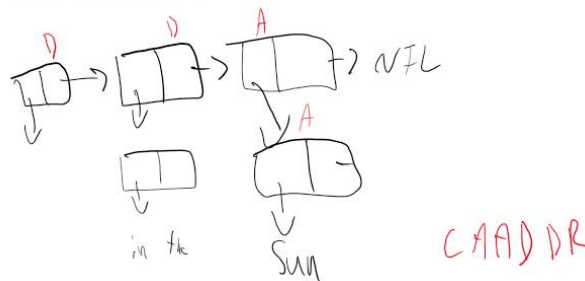
C..DDR: ((SUN))

C.ADDR: (SUN)

CAADDR: SUN

EXERCISES

2.13. Write down tables similar to the one above to illustrate how to get to each word in the list (((FUN)) (IN THE) (SUN)).



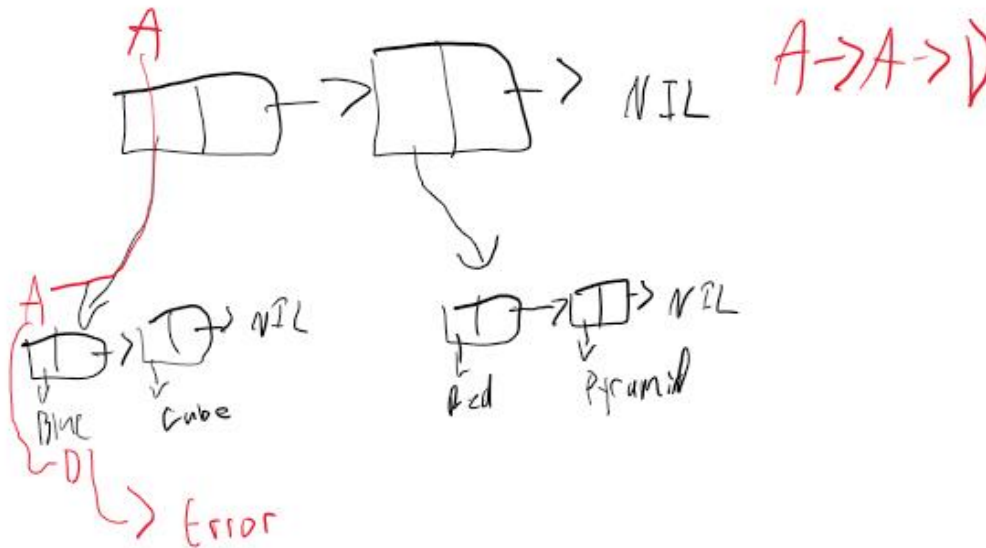
Question 2.15

2.14. What would happen if you tried to explain the operation of the CAADR function on the list ((BLUE CUBE) (RED PYRAMID)) by reading the As and Ds from left to right instead of from right to left?

We'd get an error. We'd get to the "Blue", and try to take the CDR pointer. However, 'CAR' at this point is not part of a list, but a singular element. So we'd get an error for trying to access the next

element of the list.

2.14. What would happen if you tried to explain the operation of the CAADR function on the list ((BLUE CUBE) (RED PYRAMID)) by reading the As and Ds from left to right instead of from right to left?



Question 2.16

2.16. What does CAAR do when given the input (FRED NIL)?

This would give us an error. The CAR of (FRED NIL) is 'FRED', and the CAR of that would then cause an error.

Question 2: Sebesta Ch. 1 Questions

6. In what language is most of UNIX written?
 7. What is the disadvantage of having too many features in a language?
 8. How can user-defined operator overloading harm the readability of a program?
 9. What is one example of a lack of orthogonality in the design of C?
6. Most of UNIX is written in C
 7. The disadvantage of having too many features in a language is the loss of simplicity, that can make the language more difficult to learn and use. Programmers may learn only a subset of the language, leading to readability problems if different programmers use different subsets.

8. User-defined operator overloading can harm readability when not done sensibly. For example, using + for operators that are not addition could cause confusion for both the author and the reader.
9. One example of a lack of orthogonality in the design of C is that pointers are not allowed to point to arrays.
10. What language used orthogonality as a primary design criterion?
11. What primitive control statement is used to build more complicated control statements in languages that lack them?
12. What does it mean for a program to be reliable?
13. Why is type checking the parameters of a subprogram important?
14. What is aliasing?
15. What is exception handling?
16. Why is readability important to writability?
20. What two programming language deficiencies were discovered as a result of the research in software development in the 1970s?
21. What are the three fundamental features of an object-oriented programming language?
22. What language was the first to support the three fundamental features of object-oriented programming?
23. What is an example of two language design criteria that are in direct conflict with each other?
24. What are the three general methods of implementing a programming language?
25. Which produces faster program execution, a compiler or a pure interpreter?

10. ALGOL 68 used orthogonality as a primary design criterion.

11. 'goto' is the primitive control statement used to build more complicated ones.

12. For a program to be reliable, it must perform to its specifications under all conditions.

13. Type checking the parameters of a subprogram is important for language reliability since it allows type errors to be detected (making the program more predictable, and less likely to fail due to inconsistent data).

14. Aliasing is having two or more distinct names in a program that can be used to access the same memory cell.

15. Exception handling is the ability of a program to intercept run-time errors, take corrective measures, and then continue execution.

20. The two programming language deficiencies discovered as a result of the research in software development in the 1970s were incompleteness of type checking, and inadequacy of control statements.

21. The three fundamental features of an object-oriented programming language are data abstraction, inheritance, and dynamic method binding.
 22. Smalltalk was the first to support the 3 fundamental features of OOP.
 23. An example of two conflicting design criteria are reliability and cost of execution.
 24. The three general methods of implementing a programming language are compilation, pure interpretation, and hybrid implementation.
 25. A compiler produces faster program execution.
29. What are the advantages in implementing a language with a pure interpreter?
29. The advantages of implementing in a language with a pure interpreter are easy implementation of source-level debugging operations, and their run-time error messages can refer to source-level units.