Which of the following statcments is true?

a) L is recursive
b) L is recursively enumerable but not recursive
c) L is not recursively enumerable
d) Whether L is recursive or not will be known after we find out if P=NP

5. Consider two languages L_1 and L_2, each on the alphabet $\sum$ Let f:$\sum \rightarrow \sum$ be a polynomial time computable bijection such that ($\forall$x) [x$\in L_1$iff f(x)$\in L_2$].Further, let f^{-1} be also polynomial time computable.Which of the following cannot be true?

a) $L_1 \in$ P and L_2 is finite
b) $L_1 \in$ NP and $L_2 \in$ P
c) L_1 is undecidable and $L_2$is decidable
d) L_1 is recursively enumerable and L_2 is recursive

6. The problems 3-SAT and 2-SAT are

a) Both in b) Both NP complecte
c) NP complete and P d) Undecidable and NP complete, respecively

7. Consier the following two problems on undependent graphs:
i) Given G(V,E), does G have an independnt set of size V−4?
ii) Given G(V,E), does G have an independnt set of size 5?
Wich one of the following is TRUE?

a) (i) is in P and (ii) is NP complecte b) (i) is NP complecte and (ii) is in P
c) Both (i) and (ii) are NP complecte d) Both (i) and (ii) are in P

8. Let s be an NP-complete problem and Q and R be two other problem not known to be in NP.
Q is polynomial time reducible to S and is polynomial time reducible to R.
Which one of the following ststements is true?

a) R is NP complete b) R is NP hard
c) Q is NP complete d) Q is NP hard

9. Let SHAM_3 be the problem of finding a Hamiltonian cycle in a graph G=(V,E) with V divisible by 3 and DHAM_3 be the problem of determining if a Hamiltonian cycle exists in such grapghs.
Which one of the folloWing is true?

a) Both DHAM_3 and SAHAM_3 are NP hard
b) SAHAM_3 is NP hard, but DHAM_3 is not
c) DHAM_3 is NP hard, but SAHAM_3 is not
d) Neither DHAM_3 nor SAHAM_3 is NP hard

10. Let $\prod_A$ be a problem that belongs to class NP. Then wich one of the following is true?

a) Ther is no polynomial time algorithm for $\prod_A$
b) If $\prod_A$ can be solved deterministically in polynomial time, then P=NP
c) If $\prod_A$ is NP hard, then it is NP complete
d) $\prod_A$ may be undecidable.

11. The Assuming P $\neq$ NP, wich of the following is true?

a) NP complete = NP b) NP complete $\cap$ P = ϕ
c) NP hard = NP d) P = NP complete

12. The recurrence relation capturing the optimal execution time of the tower of Hanoi problem withn discs is

a) $T(n) = 2T(n-2) + 2$ b) $T(n) = 2T(n-1) + n$
c) $T(n) = 2T(n/2) + 1$ d) $T(n) = 2T(n-1) + 1$

582 | to Automata Theory, Formal Languages and Computation

13. A list of n strings, each of lengh n, is sorted into lexicographic order using the merge-sort algorithm. The worst case running time of this computation is

a) $O(nlogn$ b) $O(n^2logn$ c) $n^2 + logn$ d) $O(n^2$

Answers:

1. b 2. c 3. c 4. a 5. b 6. c 7. c 8. c
9. a 10. a 11. b 12. d 13. a

Hints:

4. Ther are two options $(0+1)^*or\varphi$. Both are regular hence both are recursive.

Exercise

1. Determine $O(n), \omega(n), and\theta(n)$of the following

a) $f(n) = 3n^2 + 2logn + 1$
b) $f(n) = 5n^3 - 3n^2 + 5$
c) $f(n) = 5n^3 - 3n^2 - 5$
d) $f(n) = 3n - 3logn + 2$

2. Find the O or the following recurrene relations

a) $T(n) = 8T(\frac{n}{2}) + cn^2$.

b) $T(n) = \begin{cases} c & \text{for } n = 1 \\ aT(\frac{n}{b} + cn & \text{for } n > 1 \end{cases}$

wher n is a power of b

c) $T(n) = 4T(\frac{n}{2} + cn^2$

d) $T(n) = \begin{cases} 0 & \text{for } n = 1 \\ 3T(\frac{n}{2} + n - 1 & \text{for } n > 1 \end{cases}$

3.Find the time complexity of the following algorithms

a)
```
bool TsFirstElementNull( Stringl[ ] strings )

if(strings[0] == null)

return true;

return false;
```
b)
```
for (int i = 0; i ¡ N; i++)
for (int j = i + 1; j ¡ N; j++)
if (A[i] ¿ A[j] )
swap(A[i], A[j])
```

3. Given a sorted array A, determine whether it contains two elements with the difference D. Find the time complexity of the algorithm.

4. Find the time complexity of heap sort operation.
Find the time complexity of the Lagrange interpolation polynoial.

5.Find the time complexity of the Lagrange interpolation polynomial.

6. Prove that the following problem is in NP.
Given integers n and ther a factor f with 1¡f¡k and f dividing n?

7. prove that the subset sum problem is NP complete.

A set of vertices inside a graph G is an independent set if ther are no edges between any two of these vertices. Prove that for a given graph G and an integer k, "is G' $\subseteq G$ an independent set of size k? " an NP complete problem. (Reduce it to 3 CNF or vertex conver.)

9. Prove that the vertex conver is NP complete by reducing the independent set problem to it.

10. Prove that the graph colouringmproblem is NP complete.

Basics of Compiler Design 13

Introduction When we have learnt different programming languages like C, C++ or Java we got familiar with the term compiler. As an example, for C and C++ we have found Turbo C compiler, for Java we have seen Java compiler and so on. Most of the students have a wrong idea that compilation is required only to check the errors. But this is not the only job for a compiler, its job is much wider than error checking. Compiler is basically built on the knowledge of Automata. To learn different phases of compiler we need the knowledge of Finite Automata, Regular Expression, Context Free Grammar etc. Compiler needs a vast discussion, but in this section we shall discuss only those parts which are directly related to Automata.

13.1 Definition

Compiler is a program that takes a program written in a source language and translates it into an equivalent program in a target language. In relation to compiler the source language is human readable language called High Level Language and target language is machine readable code called machine level language. In the operational process of a compiler it reports the programmer the presence of errors in the source program. The block diagram of compiler is shown in Fig. 13.1.

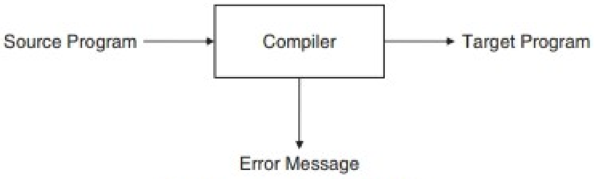

Fig. 13.1 *Block Diagram of Compiler*

13.2 Types of Compiler

Compilers are classified mainly into three types according to their operational process. 1. Single pass: In computer programming, a single pass or one-pass compiler is a compiler that scans through the source code of each compilation unit only once. In other words, a singlepass compiler does not look back the code it has previously processed. Sometimes single pass