String Pattern Recognition

Project 2

Linear-Time String Matching Algorithms

Paco Gómez Martín

DEADLINE: November 27st 2009, 23:59 p.m.

1 Introduction

This project has two goals. The first one is to study from an experimental point of view how the worst-case for the Karp-Rabin algorithm works. As seen in theory, the worst case depends on the number of false matches. The second goal is to test out the Z algorithm on several data test chosen from different contexts.

More specifically, goals of this project include:

- Reinforcement of the knowledge acquired by students about the Karp-Rabin and Z algorithms.
- Deep understanding of the parameters associated with the Karp-Rabin algorithm, in particular, the choice of parameter q.
- Testing a string matching algorithm with data close to those found in practice.
- Actual programming of string matching algorithms.
- Conducting a computational experiment.

- Comparison of theoretical and experimental results.
- Interpretation of results and presentation of conclusions.
- Writing an academic paper.

2 The Karp-Rabin Algorithm

The Karp-Rabin algorithm must be implemented. That includes choosing all the parameters of the algorithm as well as taking into account on which computer or with which compiler is used. All the chosen parameters must be justified according to the theory. Students will have to perform two experiments in this section.

Experiment 1 - Directions:

1. Set $\Sigma = \{0, 1, \ldots, 9\}$ as the alphabet for the whole experiment. Assume that the word of computer is $w = 32$.
2. Generate a text T_1 of length $n = 5,000$.
3. Generate patterns of length $m = 50$ by randomly selecting substrings of T_1. Generate a hundred patterns in total.
4. Solve then the SMC problem, that is, search for all the occurrences of each pattern in T_1 with the Karp-Rabin algorithm. Keep track of the number of false matches.
5. Consider the variable number of false matches per search. Perform a statistical analysis of the data. This analysis should include at least mean, variance, median, mode. Estimate the probability of a false match out of your statistical analysis. You should use *Statgraphics* to make the analysis.
6. Plot the data as a histogram. If necessary, use other graphical representations of data.
7. Draw conclusions from the data gathered.

Experiment 2 - Directions:

1. Repeat Experiment 1 this time setting q as the prime 4099.
2. This includes repeating the statistical analysis.
3. Draw conclusions from the data gathered. In particular, compare both data set and explain the influence of the choice of q on the algorithm's performance. Be penetrating here. I will appreciate it.

3 The Z Algorithm

The second part of this project consists of programming the Z algorithm and check it out over large files and patterns with wild cards. Below you have a table with a list of patterns and texts.

Pattern	Wild Card	Text	Alphabet
Pattern1.txt	No	Text1.txt	Binary
Pattern2.txt	Yes, char=X	Text2.txt	Binary
Pattern3.txt	No	Text3.txt	Binary
Pattern4.txt	No	Text4.txt	DNA
Pattern5.txt	Yes, char=0	Text5.txt	DNA

All the files can be found at:

http://www.eui.upm.es/˜fmartin/webpgomez/Docencia/Patter-Recognition-09-10/Project-2-Files/

Solve the SMC problem and after that answer the following questions:

1. Is pattern P in T?
2. How many occurrences of P in T are found? Think carefully before answering this question for the string matching problem with wild cards.
3. For each pair pattern-text the program should output the percentage of cases (Cases 1, 2a and 2b) it went through. Draw conclusions about those percentages and relate them to theoretical complexity.

4 Programming

Implementation of algorithms may be done in any language of student's choice. However, the language and its compiler should support certain features in order to be able to run the experiments properly. The choice of C, C++, Maple or the like should be enough. Source code and a .exe file have to be handed over.

5 Written Paper

A paper describing the following points must be handed over.

- Brief explanation of the algorithms.
- Brief explanation of the implementations. It can be done by including sufficiently detailed comments in the code.
- Brief description of the experiment.
- Interpretation of experimental data.
- Conclusions. Draw your own conclusions (be creative, but not extravagant or too showy).

The paper has to be written in correct Spanish or English; it also has to possess clarity of thought. Show me what you know; do not force to search for it through a poorly written paper.

6 Grading

The whole project counts 25% (2.5 points out of 10) of your final grade. I will take points off when:

- There is spelling mistakes (either in Spanish or in English, but I will be tougher if they occur in Spanish).
- It is plenty of irrelevant material. Down with the irrelevant!
- It lacks clarity of thought.
- It is lengthy, long-winded or poor in content.
- Code is not properly commented.
- Code is not properly structured.
- Variables have absurd names.
- There are run-time errors.

7 Questions and Office Hours

I am willing to answer your questions about algorithms, complexity or the experiment. I will not answer questions about coding errors as it is my feeling that, at this point, writing error-free code is your responsibility.