Illinois Institute of Technology
Department of Computer Science

Solutions to Second Examination

CS 330 Discrete Structures
Spring Semester, 2008

1. **Coin Tossing**

   We have a coin for which the probability of tossing heads is p, and the probability of tossing tails is $1 - p$. We toss the coin for $n > 2$ times.

   (a) What is the probability that there were no heads?

   This means that every toss must be tails; each toss has probability $1 - p$ of tails, and all are independent events, so by the rule of product, the answer is $(1 - p)^n$

   (b) What is the probability that there were at least one head?

   This means that we have the complimentary case of part (a), *not all tails*, so the answer is $1 - (1 - p)^{n-1}$.

   (c) What is the expected number of heads we will get?

   The expected number of heads in n tosses is the sum of the expected number of heads on each toss; that is, it is n times the expected number of heads on each toss, or np.

2. **Recurrences**

   Solve the following recurrences using annihilators, finding the general form of each solution. You do not need to solve the simultaneous linear equations that give the coefficients in the general form, but you must give those equations.

   (a) $S(0) = 3, S(n) = 2S(n-1) + 2$, $n \geq 1$.

   The annihilator is $(E - 2)(E - 1)$, so the solution is $\alpha a^i + \beta b^i$. Using the initial conditions, $S(0) = 3 = \alpha 2^0 + \beta = \alpha + \beta$ and $S(1) = 8 = \alpha 2^1 + \beta = \alpha 2 + \beta$, which gives $\alpha = 5$, and $\beta = -2$. The solution is thus $S(n) = 5 \cdot 2^n - 2$.

   (b) $F(0) = 3, F(1) = 1$, $F(n) = 6F(n-1) - 9F(n-2)$, $n \geq 2$.

   The annihilator is $(E^2 - 6E + 9) = (E - 3)^2$, so the solution is $(\alpha i + \beta)3^i$. Using the initial conditions, $F(0) = 3 = \beta$ and $F(1) = 1 = (\alpha 1 + \beta)3^1 = (\alpha + \beta)3$, so $\beta = 3$, and $\alpha = -\frac{8}{3}$. The solution is thus $F(n) = (-8n/3 + 3)3^n$.

   (c) $a_0 = 1, a_1 = 1$, $a_n = 2a_{n-2} - a_{n-1} + 1$, $n \geq 2$. The annihilator is $(E^2 - 2E + 1)(E - 1) = (E - 1)^3$, so the solution is $a_n = (\alpha_0 + \alpha_1 i + \alpha_2 i^2)1^i$. Using the initial conditions, $a_0 = 1 = \alpha_0$, $a_1 = \alpha_0 + \alpha_1 + \alpha_2$, and $a_2 = 2 \cdot 1 - 1 + 1 = 2 = \alpha_0 + \alpha_1 2 + \alpha_2 4$, we get $\alpha_0 = 1$, $\alpha_1 = -1/2$, and $\alpha_2 = 1/2$. The solution is thus $a_n = 1 - n/2 + n^2/2$

3. **Rolling a Die.**

   You roll an ordinary six-sided die $n \geq 2$ times, recording the sequence of rolls.

   (a) Find a recurrence relation for the number of possible sequences in which there are no two consecutive sixes.

   Call such a sequence “valid”. If the first roll is not a 6 (that is, is 1, …, 5), the sequence can be continued with any valid sequence of lenth $n - 1$; there are a_{n-1} such ways to continue the sequence and 5 choices for the first roll, a total of $5a_{n-1}$ by the rule of product. On the other hand, if the first roll *is* a 6, the next roll must be 1, …, 5 and the reamaing $n-2$ rolls can be any valid sequence; there are $5a_{n-2}$ was to continue after the first throw by the rule of product. By the rule of sum then, $a_n = 5a_{n-1} + 5a_{n-2}, n \geq 2$. Obviously, $a_0 = 1$ and $a_1 = 6$.

(b) Solve the recurrence using annihilators; you need not solve the simultaneous equations from the initial conditions.

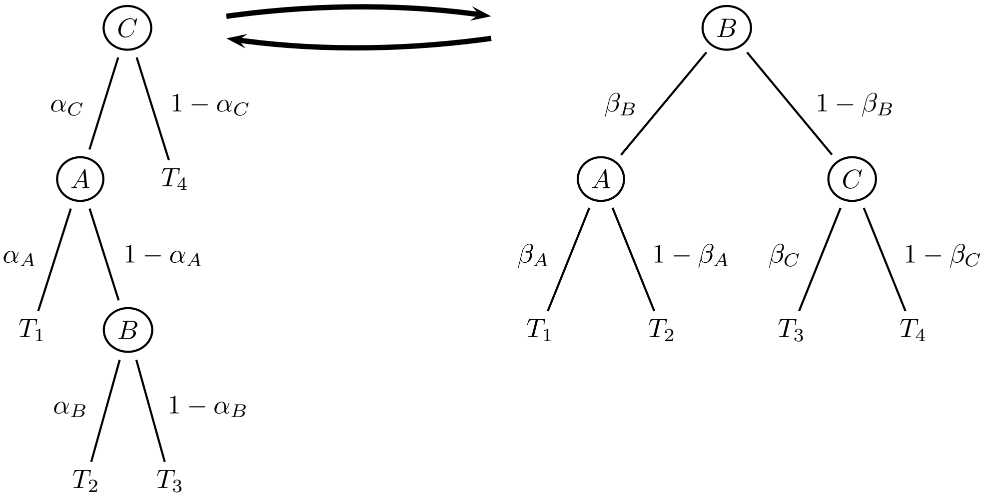
The annihilator is $(E^2 - 5E - 5) = (E - r_1)(E - r_2)$ where $r_1 = (5 + \sqrt{5})/2$ and $r_2 = (5 - \sqrt{5})/2$, so the solution is $a_n = \alpha_1 r_1^n + \alpha_2 r_2^n$. The initial conditions give us $a_0 = 1 = \alpha_1 + \alpha_2$ and $a_1 = 6 = \alpha_1 r_1 + \alpha_2 r_2$.

(c) How does the probability that a sequence of n rolls of the die with no two consecutive sixes grow as $n \to \infty$?

There are 6^n possible sequences of rolls of the die, of which $\alpha_1 \left(\frac{5+\sqrt{5}}{2}\right)^n + O(1)$ have no two consecutive sixes. The probability thus grows (shrinks, actually) like $\left(\frac{5+\sqrt{5}}{12}\right)^n$.

4. **Double Rotations.**

Consider the double rotation of a binary search tree shown below.



(a) Explain why $\beta_B = \alpha_C\alpha_A + \alpha_C(1-\alpha_A)\alpha_B$.

$\beta_B = \mathbf{Pr}\{x < B\}$ where x is what we are searching for. We need to go to either T_1 or T_2. To go to T_1 we have to go through both C and A. To go to T_2 in the lefthand tree we have to go left through C and left through A, or left through C, right through A, and left through B.

(b) Express β_A and β_C in terms of α_A, α_B, and α_C.

Use Bayes' Theorem:

$$\begin{aligned}\beta_C &= \mathbf{Pr}\{x < C | x > B\} \\ &= \frac{\mathbf{Pr}\{x < C\}\mathbf{Pr}\{x > B | x < C\}}{\mathbf{Pr}\{x > B\}} \\ &= \frac{\alpha_C(1-\alpha_A)(1-\alpha_B)}{\mathbf{Pr}\{x > B\}}\end{aligned}$$

$$\begin{aligned}
&= \frac{\alpha_C(1-\alpha_A)(1-\alpha_B)}{1-\mathbf{Pr}\{x<B\}} \\
&= \frac{\alpha_C(1-\alpha_A)(1-\alpha_B)}{1-\alpha_C\alpha_A-\alpha_C(1-\alpha_A)\alpha_B}
\end{aligned}$$

and

$$\begin{aligned}
\beta_A &= \mathbf{Pr}\{x<A|x<B\} \\
&= \frac{\mathbf{Pr}\{x<A\}\mathbf{Pr}\{x<B|x<A\}}{\mathbf{Pr}\{x<B\}} \\
&= \frac{\alpha_A\alpha_C 1}{\mathbf{Pr}\{x<B\}} \\
&= \frac{\alpha_A\alpha_C}{\alpha_C\alpha_A+\alpha_C(1-\alpha_A)\alpha_B} \\
&= \frac{\alpha_A}{\alpha_A+(1-\alpha_A)\alpha_B}
\end{aligned}$$

5. **Divide-and-Conquer**

Having seen the power of recursion and divide-and-conquer, CS 330 student decided to write a program to compute x^n,

(a) His first attempt was

```
function Power($x$, $n$)
if $n = 0$ then
    return 1
else if $n$ is odd then
    return $x$ * Power($x$, $\lfloor n/2 \rfloor$) * Power($x$, $\lfloor n/2 \rfloor$)
else
    return Power($x$, $\lfloor n/2 \rfloor$) * Power($x$, $\lfloor n/2 \rfloor$)
end if
```

Analyze the time required by this algorithm.

$T(0) = c$, $T(n) = 2T(n/2) + k$, where the constants c and k are, respectively, the small constant amounts of time used to return 1, or test the parity of n and to multiply the two recursively computed values and return the result. Using the secondary recurrences and annihilators, $t_i = T(n_i)$, where $n_i = 2n_{i-1}$, $n_0 = 1$ so that (as in class), $n_i = 2^i$, the annihilator for $t_i = 2t_{i-1} + k$ is $(E-2)(E-1)$, and so $t_i = T(n_i) = 2^i = n_i$; that is, $T(n) = \Theta(n)$.

(b) His second attempt was

```
function Power($x$, $n$)
if $n = 0$ then
    return 1
else
    integer $t \leftarrow$ Power($x$, $\lfloor n/2 \rfloor$)
    if $n$ is odd then
        return $x$ * $t$ * $t$
    else
        return $t$ * $t$
    end if
```

```
end if
```

Analyze the time required by this algorithm.

$T(0) = c$, $T(n) = T(n/2) + k$; the same secondary recurrence as in part (a) leads to $t_i = t_{i-1} + k$ which has annihilator $(E - 1)^2$, so $t_i = T(n_i) = \Theta(i)$; that is $T(n) = \Theta(\log n)$.