“Pass the Buck” on a Complete Binary Tree

Kenneth Levasseur
Department of Mathematical Sciences
University of Massachusetts Lowell
Lowell, Massachusetts, USA
kenneth_levasseur@uml.edu

June 18, 2019

Abstract
The Stochastic Abacus is employed to compute winning probabilities at each level of the game “Pass the Buck” on a complete binary tree with the starting vertex being the root of the tree. The derivation is also generalized to play on complete $k$-ary trees.

Introduction. In the 1970’s, Engel [2] devised the Stochastic Abacus as a way to compute probabilities for certain discrete probability problems with minimal numerical computation. More recently, Torrence [6] used the same technique to determine winning probabilities for players in the game “Pass the Buck” for a variety of families of graphs. The Stochastic Abacus has found more widespread exposure due to a recent article by Propp [4] in Math Horizons. In this note, the game is analyzed for compete binary trees where the start vertex is the root, and we derive winning probabilities for nodes at different levels. A limiting probability of $\sqrt{2} - 1$ for the root to win as the number of levels goes to infinity is derived. We also observe that the derivation easily generalizes to complete $k$-ary trees.

1 Pass the Buck
The game “Pass the Buck” is played on a connected undirected graph, with a distinguished “start vertex.” The game proceeds in steps starting with the start vertex holding a prize (the “buck”). At every stage in the game, the current vertex that holds the buck and its neighboring vertices are selected randomly and uniformly. If the the current vertex is selected, the game ends with that vertex winning. If a neighboring vertex is selected the buck is passed there and the process is repeated. More precisely, if the degree of the vertex that holds the buck is $k$, then the buck moves to any of the neighbors with probability $\frac{1}{k+1}$ and ends with the player at the current vertex winning with probability $\frac{1}{k+1}$.

2 The Stochastic Abacus
For different graphs, the derivation of the probabilities of any vertex winning can be derived in a variety ways. For example, we can develop a system of equations that determine the probabilities. The game can also be modeled as a Markov Chain and our desired probabilities can be computed using well-know
techniques. See Kemeny and Snell [3] for a general introduction to Markov chains and Snell [5] for a discussion of the connection between Markov chains and the Stochastic Abacus. Alternatively, the Stochastic Abacus method (also known as Engel’s Algorithm) uses only elementary transition rules to compute winning probabilities. We will illustrate all three methods for the case of a complete binary tree up to level 1, Figure 1, which will serve as a basis for computing the probabilities in larger trees.

**Figure 2.1:** A complete binary tree to one level

In this form of the game, we have three players (root, L, and R) and the buck starts at the root. We can easily derive the probabilities for each vertex by observing that $p_{\text{root}} = \frac{1}{3} + 2 \left( \frac{1}{3}p_{\text{root}} \right) \Rightarrow p_{\text{root}} = \frac{1}{2}$. By symmetry, $p_L = p_R = \frac{1}{4}$.

Although this is the shortest of our derivations, it doesn’t scale so easily.

Next, we will derive the probabilities using Markov chain theory. The states $v_x$ corresponding each of the three vertices are states for which the game is in progress, and are non-absorbing states for the process. We add three absorbing states $w_x$ to represent winning outcomes for each player. An absorbing state is one for which the process never leaves, representing the end of the game in our case. The transition matrix for the process with ordering of states $v_L, v_{\text{root}}, v_R, w_L, w_{\text{root}}, w_R$ is

$$
T = \begin{pmatrix}
0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\
\frac{1}{3} & 0 & \frac{1}{3} & 0 & \frac{1}{3} & 0 \\
0 & \frac{1}{2} & 0 & 0 & 0 & \frac{1}{2} \\
0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 1
\end{pmatrix}
$$

By listing absorbing states last, the form of the transition matrix of an absorbing Markov chain with $k$ absorbing states is

$$
\begin{pmatrix}
Q & R \\
0 & I_k
\end{pmatrix}
$$

In our case $k = 3$. If there are a total of $m$ states, $Q$ is an $(m - k) \times (m - k)$ matrix of transition probabilities between the non-absorbing states. The transition probabilities from non-absorbing states to absorbing states is contained within $R$. The matrix $NR$, where $N = (I - Q)^{-1}$ is the matrix of probabilities into the different absorbing states. The $j^{th}$ row of $NR$ contains the probabilities of ending in the absorbing states assuming the process starts in state $j$.

In our example,

$$
NR = \begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1
\end{pmatrix} \quad - \quad \begin{pmatrix}
\frac{1}{3} & 0 & 0 \\
0 & \frac{1}{3} & 0 \\
0 & 0 & \frac{1}{2}
\end{pmatrix} \quad = \quad \begin{pmatrix}
\frac{5}{6} & \frac{1}{3} & \frac{1}{6} \\
\frac{1}{6} & \frac{1}{3} & \frac{1}{6} \\
\frac{1}{2} & \frac{1}{3} & \frac{1}{2}
\end{pmatrix}
$$
We are mostly concerned with the middle row, which gives probabilities that are consistent with the previous derivation. For this simple case, the graph we have considered is also a path graph and the probabilities were the starting position is either $L$ or $R$ is consistent with the general case of “pass the buck” on a path graph starting at an end vertex as was considered by Torrence [6].

The same probabilities we have observed twice will be arrived at using a Stochastic abacus. The abacus is constructed by first considering the graph to be directed, with each undirected vertex becoming a pair of directed edges. Then we add terminal vertices to the graph, one for each “internal vertex,” and an edge leading into each terminal vertex, as in Figure 2. The root is designated as the starting position. Initially, we deposit chips into the internal vertices, the number chips being one less than the outdegree of each vertex in the augmented graph, as indicated in each vertex of Figure 2. At this point, the system is “critically loaded.” The process then consists of sequentially, adding a chip to the start vertex, $v_{\text{root}}$, and then repeatedly “firing” chips whenever the content of a vertex is greater than or equal to the outdegree of that vertex. This involves distributing a chip along each outgoing edge of the “loaded vertex” to neighboring vertices. This process of adding chips and firing as long as possible continues until the chip content of the nonterminal vertices returns to the critical loading state.

![Figure 2.2: Stochastic Abacus for a Complete Level 1 Binary Tree](image)

The remarkable fact is that after we have returned to the original critical loading of internal vertices, the probability that any vertex wins the game is equal to the number of chips in its corresponding terminal vertex divided by the total number of chips in all terminal vertices. See Snell [5] for a proof.
Here is a step by step account of how the process plays out in our example.

| Step | Comment | root | L | R | \( t_{\text{root}} \) | \( t_L \) | \( t_R \) |
|------|---------|------|---|---|----------------|------|------|
| 1    | Critically loaded | 2    | 1 | 1 | 0 | 0 | 0 |
| 2    | Add 1 to root | 3    | 1 | 1 | 0 | 0 | 0 |
| 3    | root fires | 0    | 2 | 2 | 1 | 1 | 1 |
| 4    | \( L \) fires | 1    | 0 | 2 | 1 | 1 | 0 |
| 5    | \( R \) fires | 2    | 0 | 0 | 1 | 1 | 1 |
| 6    | Add 1 to root | 3    | 0 | 0 | 1 | 1 | 1 |
| 7    | root fires | 0    | 1 | 1 | 2 | 1 | 1 |
| 8    | add 1 to root | 1    | 1 | 2 | 1 | 1 | 1 |
| 9    | add 1 to root | 2    | 1 | 1 | 2 | 1 | 1 |

It may not be obvious, but if two nodes can fire, as in steps 4 and 5, it doesn’t matter in what order they are fired. See Bjöner [1] for a proof. After step 9, the three interior vertices are back to being critically loaded, and the process ends. The total number of chips in the terminal vertices is 4 and the root had 2, so its probability of winning is \( \frac{1}{2} \), consistent with our previous derivation. The other two vertices again have winning probability \( \frac{1}{4} \).

### 3 Pass the Buck on a Complete Binary Tree

Consider the game of Pass the Buck on a complete binary tree with \( n \) full levels, \( n \geq 0 \), where the buck starts at the root of the tree. We derive formulae for the probabilities that any node at level \( k \) of the tree, \( 0 \leq k \leq n \) will win the game. Our derivation is based on an observation that the chips needed at different levels is recursive, with a second order recurrence.

**Theorem 3.1** The number of chips in the terminal vertex of the root at the end of the stochastic abacus process, \( a(n) \), follows the recursion \( a(n) = 4a(n-1) - 2a(n-2), n \geq 2 \).

**Proof.** If a complete binary tree to \( n \) levels is critically loaded and augmented with terminal vertices, we observe that in order to return the two subtrees starting at level 1 to critical loading status, each vertex at level 1, which are roots of binary trees of level \( n-1 \), must fire \( a(n-1) \) times. Every time these vertices fire, they need 4 chips. The initial loading of three chips to each of the vertices at level 1 are used in the first firing, but then to return to critical loading, three other chips are needed. Therefore the root must fire \( 4a(n-1) \) times, almost. There is one other source of chips to each vertex at level 1. That is the vertices at level 2. Each time they fire, they return a chip back up to level 1. Therefore, the root must fire \( a(n) = 4a(n-1) - 2a(n-2) \) times to complete the process. This is then the number of chips that are deposited into the terminal vertex of the root. \( \blacksquare \)

We will apply this recursion together with the two base cases, \( a(0) = 1 \) and \( a(1) = 2 \), to compute our probabilities.

**Corollary 3.2** At the end of the stochastic abacus process on a complete binary tree of level \( n \), the number of chips deposited into each terminal vertex at level \( k \), \( 0 \leq k \leq n \) is \( a(n-k) \). The total number of chips that are in all terminal vertices is then \( \sum_{k=0}^{n} 2^k a(n-k) \).

**Corollary 3.3** With the initial conditions \( a(0) = 1 \) and \( a(1) = 2 \), \( a(n) = \frac{1}{2} \left( (2 - \sqrt{2})^n + (2 + \sqrt{2})^n \right) \), and the total number of chips in all terminal vertices at the end of the stochastic abacus process is \( t(n) = \sum_{k=0}^{n} 2^k a(n-k) = \)
\[ \frac{\left(2+\sqrt{2}\right)^{n+1} - \left(2-\sqrt{2}\right)^{n+1}}{2\sqrt{2}}. \]

Let \( p(n, k) \) be the probability that one of the \( 2^k \) vertices at level \( k \) win the level \( n \) game. The probability that the root is the winner of the level \( n \) game is

\[ p(n, 0) = \frac{a(n)}{t(n)} = \frac{\sqrt{2} \left(\left(2 - \sqrt{2}\right)^n + \left(2 + \sqrt{2}\right)^n\right)}{(2 + \sqrt{2})^{n+1} - (2 - \sqrt{2})^{n+1}}. \]

Interestingly, \( \lim_{n \to \infty} p(n, n) = \sqrt{2} - 1 \). More generally, \( p(n, k) = \frac{a(n-k)}{t(n)} \) and

\[ \lim_{n \to \infty} p(n, k) = \frac{\sqrt{2}}{(2 + \sqrt{2})^{k+1}}. \]

\[ 4 \] Pass the buck on \( k \)-ary Trees

We can generalize our argument on binary trees to \( k \)-ary trees, \( k \geq 2 \). A complete \( k \)-ary tree to level \( n, n > 0 \), will have a root and \( k \) subtrees, each a \( k \)-ary tree to level \( n - 1 \). A \( k \)-ary tree up to level \( 0 \) is a single vertex, which is the form of a leaf for larger trees. By the same logic as the binary case, if we critically load the augmented directed graph for a complete \( k \)-ary tree up to level \( n, n \geq 2 \), then the number of firings of the start vertex that are needed to return to the critically loaded state, \( a(k, n) \), satisfies the recursion

\[ a(k, n) = (k + 2)a(k, n - 1) - ka(k, n - 2). \]

The basis is \( a(k, 0) = 1 \) and \( a(k, 1) = 2 \).

References

[1] Bjöner, A., Lovasz, L., Shor, P. (1991), Chip-firing games on graphs, Eur. J. Combin. 12 (4), 283–291, doi.org/10.1016/s0195-6698(13)80111-4.

[2] Arthur Engel (1976), Why does the probabilistic abacus work?, Educational Studies in Mathematics 7, 59–69.

[3] John G. Kemeny and J. Laurie Snell, Finite Markov Chains, Undergraduate Texts in Mathematics, Springer-Verlag, New York, 1976.

[4] Propp, J. (2018), Prof. Engel’s marvelously improbable machines, Math Horizons, 26(2): 5–9. doi.org/10.1080/10724117.2018.1518840.

[5] J. Laurie Snell, The Engel algorithm for absorbing Markov chains, Available at https://arxiv.org/abs/0904.1413v1

[6] Bruce Torrence, Passing the Buck and Firing Fibonacci: Adventures with the Stochastic Abacus, The American Mathematical Monthly, May 2019, 126 no. 5, 387–399, doi.org/10.1080/00029890.2019.1577089.