From a Ball Game to Incompleteness

Arindama Singh
Department of Mathematics
IIT Madras, Chennai-600036

Abstract

We present a ball game that can be continued as long as we wish. It looks as though the game would never end. By applying a result on trees, we show that the game nonetheless ends in some finite number of moves. We then point out some deep results on the natural number system connected with the game.

Introduction

We have a box containing a finite number of balls each labeled with a natural number $1, 2, 3, \ldots$. Outside the box there are again balls labeled with natural numbers. The ball game requires only one player; let us say, you are playing the game. First, you take a ball out of the box; and read its label. If the label is 1, you keep the ball outside. If the label is $m > 1$, then you choose two natural numbers, say, ℓ and n. The number ℓ must be less than m and the number n can be any natural number of your choice. Next, you keep that ball labeled m outside, and put n number of balls each labeled ℓ into the box. This constitutes one move of the game. In how many moves the box will be empty?

1 The game leads to a tree

Initially the box contains some number of balls with some labels. If there is only one ball labeled 1 in the box initially, then in a single move the box becomes empty. If the box contains 1000 balls each labeled 1, then the box becomes empty after 1000 moves. To make the game interesting, let us assume that the box contains at least one ball whose label is more than 1. And, outside the box, there are infinite supply of balls of each label.

If there are 1000 balls in the box with exactly one ball labeled 2 and the remaining balls labeled 1, then within the first 1000 moves, you will come across that ball labeled 2. In that instance, you will (have to) choose ℓ as 1, and of course, n can be any natural number. If you wish to empty the box sooner than later, then you choose your n as 1 also. And in 1001 moves, the box becomes empty. However, if you want to prolong the game, you may choose your n as a larger number, say, 10^{10}. Then the box becomes empty in $10^{10} + 1000$ moves. So, you may prolong the game as long as you wish.

The game can become very complicated. Suppose, initially, there are 1000 balls in the box, labeled with $1,\ 10,\ 10^2,\ \ldots, 10^{1000}$. At the start, let us say, the ball that is taken out has label 10^{10}. In the first move, you replace that ball with 1000 balls labeled 10^9. You continue the game. In some move of the game one of these 10^9 balls will be picked up; and suppose you replace that with 100 balls each labeled with 10^8. Once more , in some later move, one of these 10^8 balls will be picked; and at this instant, suppose you decide to replace that with 500 balls each labeled with 10^7. If you continue the game in this fashion, and look at which balls give rise to which ones and so on, then schematically, we my represent it as follows:

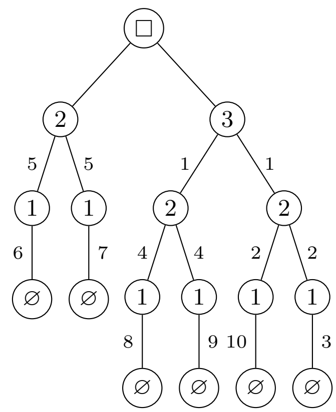
$$10^{10} \to 10^9 \to 10^8 \to 10^7 \to \cdots \to 10^1 \to 1.$$

And then the ball labeled 1 is finally removed. Notice that such a path records only one of your 1000 or 100 or 500 $\cdots$ replaced balls, in each level.

To have grip, let us think of a small game. Suppose initially, there are only two balls in the box, one labeled 2 and the other labeled 3. Instead of 'a ball labeled m', let us say m for simplicity. While playing suppose, first you picked up (randomly) 3. You replaced this with two 2s. So, there are now three 2s in the box. Next, you picked up a ball; it is 2. You then replaced it with two 1s. The box now contains two 2s and two 1s. Then you picked up a 1. It is just kept outside. Next, you picked up a 2. It is replaced with two 1s. Now the box contains one 2 and three 1s. Next you picked the 2. It is replaced with two 1s. Now, there are five 1s in the box. In the next five moves, the box becomes empty. Taking stock at each move, this particular way of playing the game can be seen as follows:

$$\{2,\underline{3}\} \stackrel{1}{\to} \{2,2,\underline{2}\} \stackrel{2}{\to} \{2,2,1,\underline{1}\} \stackrel{3}{\to} \{2,\underline{2},1\} \stackrel{4}{\to} \{\underline{2},1,1,1\} \stackrel{5}{\to} \{1,1,1,1,\underline{1}\} \stackrel{6}{\to} \{1,1,1,\underline{1}\}$$
$$\stackrel{7}{\to} \{1,1,\underline{1}\} \stackrel{8}{\to} \{1,\underline{1}\} \stackrel{9}{\to} \{\underline{1}\} \stackrel{10}{\to} \varnothing.$$

The underlines show which ball has been replaced, and the superscripts on the arrows show the number of moves taken so far. This particular game gives rise to the following (directed) tree:



The initial box is shown as □; and when a 1 is kept aside, it is indicated by ∅. The numbers on the edges show the number of moves taken so far. The tree has all the details of how the game has been played.

In general, any particular way of playing the game gives rise to such a tree. Further, notice that choosing a particular label at a certain move does not matter; it gives rise to the same tree with possibly different label of the edges. Thus in order to determine whether such a game ever ends, it is enough to consider such trees without edge labeling.

In the sequel, we will be using only directed trees. If there is an edge from a node x to a node y, then in a diagram, we will show y below the node x; and say that y is a child of x. For instance, in the above tree, the node that contains □ has two children containing 2 and 3. These instances of 2 and 3 are the labels of the balls that the box contains initially. This node containing 2 has two children containing 1 and 1; and the node that contains 3 has two children containing 2 and 2; and so on. Further, we will use the following terminology.

If the number of nodes in a (directed) tree is finite, we call the tree as *finite*; else the tree is called *infinite*. If a tree has a designated node which is not a child of any other node, then such a node is called a *root*. A tree with a root is called a *rooted tree*. A *leaf* is a node which has no children. A *finitely generated* tree is a tree in which each node has a finite number of children. In a rooted tree, a sequence of nodes from the root to a leaf, where the $(j+1)$th node is a child of the jth node, is called a *path*.

For instance, in the above tree, the root contains □; and all leafs are nodes that contain ∅. Since any ball is replaced with a finite number of other balls, in any tree depicting a particular game, each node will have a finite number of children. Thus all our trees are finitely generated. Further, the left most path in the above tree has nodes containing □, 2, 1, ∅ in that order. The rightmost path has the nodes containing □, 3, 2, 1, ∅.

2 The Ball Game Ends

Observe that in a move when a ball labeled m is picked, we have to choose balls of smaller label; but we have freedom in choosing the number of such balls. This choice does not affect the height of the tree, but it may either slim down the tree or make it fatter. Again, if the largest label of the balls initially inside the box is k, then any longest path in a tree will have $k+2$ nodes. For instance, in the tree of the last section, the largest label is 3 and a longest path (there are four such) has 5 nodes. Thus corresponding to each game, the tree is finitely generated and each path in such a tree is finite; that is, each path has a finite number of nodes. However, such a tree can become arbitrarily fat, though finite. You may measure the fatness of a tree by the largest number of nodes in a level of the tree. For instance, in the tree of the last section, this number is 6, which happens to be in the third level.

The game will end provided the number of nodes in each level is finite. But how do we guarantee that, when we very well know that we can choose as an arbitrary number n of balls of label $\ell < m$ to replace a ball of label m? Certainly, this number does not exceed the total number of nodes in such a tree. This raises the question whether the total number of nodes in such a tree is finite irrespective of our choices of n at each move. A known result called the *Fan Theorem*, (See [2].) comes of help. It is stated as follows

If each path in a finitely generated rooted tree is finite, then the tree is finite.

To summarize, take the box as the root, and its children as the balls which are there initially. When we replace a ball with finitely many others, these new balls become the children of the removed ball. Each particular game is thus a finitely generated tree. In such a tree, label the root, i.e., the box, with one plus the maximum of labels of all balls the box contains initially. All other nodes (the balls) are already labeled with natural numbers. In this tree, the children have smaller labels than that of their parent nodes. If the root has label i, then its children have labels less than i; the children of those children have labels less than $i - 1$; and so on. Thus, any path in such a tree is finite. Then use the fan theorem to conclude that any such tree is finite. Therefore, given a box of balls initially, whatever way the game continues, it will eventually come to an end; the box will be eventually empty. However, there is no bound to the number of moves for ending the game.

3 Incompleteness Phenomenon

Form the ball game, we conclude that given any natural number k, we can make the game not to end in k or fewer number of moves; yet the game comes to an end. We look at the phrase "the ball game definitely ends in k or fewer moves" as a property of the natural number k. Call this property as $P(k)$. Then "the ball game eventually ends" is translated as $\exists k P(k)$. Thus we obtain the following:

None of $P(1),\ P(2),\ P(3),\ \ldots$ is true but $\exists k P(k)$ is true.

This is paradoxical. Let us look back. We have *proved* (in a sense) that for no k, $P(k)$ can be shown to be true. And also we have proved that $\exists k P(k)$ can be shown to be true. Thus we have proved the following:

$$\text{Neither of } P(1),\ P(2),\ P(3),\ \ldots, \text{ is provable, but} \exists k P(k) \text{ is provable.} \tag{1}$$

If this is still counter-intuitive, we play around with it a little bit. Write $Q(k)$ for "the game does not end in k or fewer moves". Since we can always prolong the game for more than any given k moves, we see that $Q(k)$ is provable for any k. But we have also proved that the game eventually ends. That means, the sentence $\forall k Q(k)$ is not provable. This amounts to the following:

$$\text{Each one of } Q(1),\ Q(2),\ Q(3),\ \ldots \text{ is provable, but} \forall k Q(k) \text{ is not provable.} \tag{2}$$

Observe that this involves no paradox. The reason is proofs are finite in length; thus we may have a proof of $Q(1)$, a proof of $Q(2)$, and so on, but we do not know how to combine these infinite number of proofs to create a proof for $\forall k Q(k)$.

In such a case, when we have proofs for each $Q(k)$, necessarily each such $Q(k)$ is true. And from this we conclude that $\forall k Q(k)$ must also be true. But we cannot prove it. On the other hand, we cannot prove its negation since it is true. It so happens that such a sentence $\forall k Q(k)$ can be formalized in the first order axiomatic theory of natural numbers, called the *Arithmetic*, where the notion of proof is completely formalized. And then it would lead to the following:

> There exists a sentence in the first order theory ofnatural numbers which is true
> but neither it nor its negation is provable. (3)

The result in (3) is the negation incompleteness theorem of K. Gödel; the result in (2) is called the ω-incompleteness of Arithmetic; and the one in (1) is called the $\exists$-incompleteness of Arithmetic.

4 Conclusion

In this article, we have looked afresh at a ball game devised by the American logician R. M. Smullyan for explaining *König's Lemma* on infinite trees. The lemma states that each finitely generated infinite tree has an infinite path. We have used the contraposition of this lemma, which is called the fan theorem. For proofs of these results, see [1, 2]. The ball game brought into fore a possible gap between the truth in natural number system and provability in Arithmetic. This gap is demonstrated (informally) by the three incompleteness results such as negation incompleteness, ω-incompleteness and $\exists$-incompleteness of Arithmetic. To work through formal proofs of these results, you may see the references.

Acknowledgement

The author thanks the referee for very constructive suggestions that improved the presentation of the paper.

References

[1] A.Singh, *Logics for Computer Science, 2nd Ed.* To appear, PHI, 2017.

[2] R. M. Smullyan, *A Beginner's Guide to Mathematical Logic*. Dover, 2014.