The Pebble Game as a Computational Model

(Abstract)

Ing. Jakub Budiský, ibudisky@fit.vutbr.cz

Theoretical computer science provides several models for studying programming languages and programs themselves. Development of these models was mainly driven by the need for formal verification and efficiency of programs.

One of the byproducts of the work on the formal languages and their semantics is The Pebble Game [1]. This game can be used to study and demonstrate possible trade-offs between time and space complexity of algorithms on a specific task.

In The Pebble Game, direct acyclic graphs (DAGs) are used to represent a problem that we are trying to solve. Vertices represent data needed to perform a computational step and edges are used to model dependencies between them.

Pebbles, placed on the vertices, represent the current execution progress. Each pebble is associated with a space needed to store the data from the vertex, and can be seen as a CPU register. Total number of pebbles needed throughout the game then corresponds to the space complexity of a chosen algorithm.

With a set of simple rules on how to place pebbles, the goal of the game is to pebble all of the output vertices. Time complexity can be then derived from the number of steps it took to achieve this goal.

In this presentation, The Pebble Game is demonstrated using a few examples and some aspects of this game are observed. Furthermore, limitations of this approach are discussed along with its modification, The Red-Blue Pebble Game.

References:

[1] J. E. Savage, “Models of computation, exploring the power of computing”, p. 698, 2008.