Bowls and beans

Suppose to have an array of bowls each containing a number of beans. We have the following *bean rule* for moving beans about:

> If a bowl contains two or more beans, pick any two beans in it and move one of them to the bowl on its left and the other to the bowl on its right.

A bean step according to the bean rule is presented on the left in Figure 1, where we have coloured the moved beans to visualise movement. In fact, we assume that we live in an ideal world: beans are indistinguishable from one another, a bowl can contain an arbitrary number of beans, and the array of bowls extends indefinitely to either side. The problem is to show that if we start with *any* situation in which there is only a *finite* number of beans, only a finite number of successive bean steps is possible. Even stronger, for a given situation all its final situations are in fact the same and reached in the same number of steps!

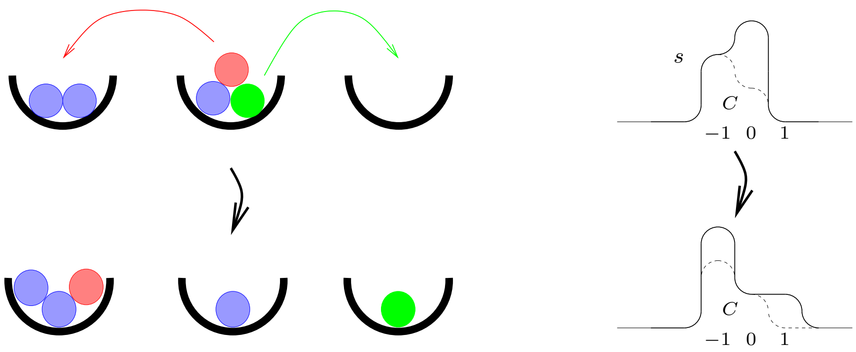

Figure 1: Bean step in reality (left) and in mathematical model (right)

Mathematical modeling

Definition 1 • *A* situation *is a map* $s{:}\mathbb{Z}{\to}\mathbb{N}$.

- *The* bean rule *is the pair* $\langle l, r\rangle$ *of triples, with* $l = \langle 0,2,0\rangle$ *and* $r = \langle 1,0,1\rangle$.
- *A* bean step at i *has shape* $C + l^i \curvearrowright_i C + r^i$*, for some situation* C *and integer* i*. Here,* $+$ *denotes pointwise addition, the* i-shift s^i *of a situation* s *is defined by* $s^i(j) = s(j - i)$*, and a triple* $\langle m, n, k\rangle$ *of natural numbers denotes the situation defined by* $-1 \mapsto m$*,* $0 \mapsto n$*,* $1 \mapsto k$*, and* 0 *elsewhere.*

Example 2 • *The initial situation top left in Figure 1 is modeled as the situation to its right, i.e. as* s *defined by* $s(0) = 3$*,* $s(-1) = 2$*, and* 0 *everywhere else,*

- *The bean step on the left Figure 1 is modeled as the bean to its right, i.e. by a bean step at position* 0*, taking* C *identical to* s*, except that* $C(0) = 1$*.*

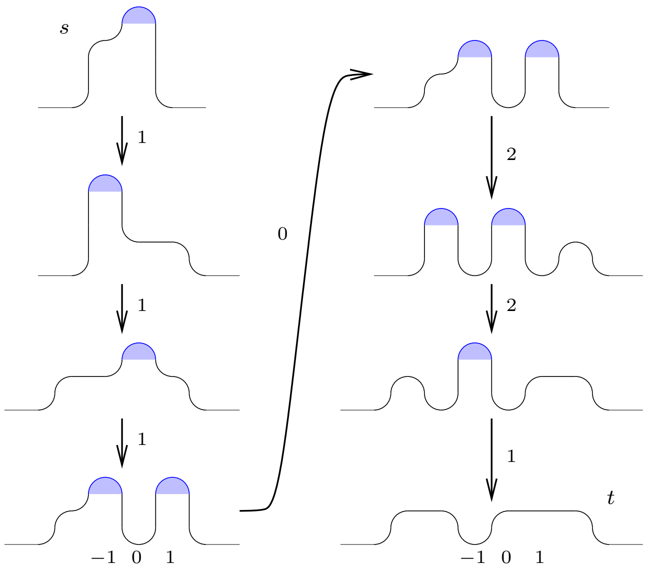

Figure 2: Bean run

- *An exhaustive* bean run *consisting of bean steps, results in the situation* t*, which yields* 1 *at positions* -3*,* -2*,* 0*,* 1 *and* 2*, and* 0 *everywhere else. (see Figure 2). The steps in the figure are in fact* parallel *steps performing a number, indicated alongside the arrows, of bean steps* in parallel*.*

Since these data define an abstract rewrite system (ARS) $\curvearrowright$, we reformulate the puzzle using rewriting terminology:

(P) $\curvearrowright$ is terminating, $\curvearrowright$ has unique normal forms for rewriting, and all $\curvearrowright$-reductions to normal form have the same length.

Solution

Definition 3 *An ARS* $\rightarrow$ *is* linear orthogonal *(LO) if for every* fork $t \leftarrow s \rightarrow u$*, either* $t = u$*, or* $t \rightarrow v \leftarrow u$*, for some* v*.*

Theorem 4 ([Toy92]) *If* $\rightarrow$ *is LO and normalising, then (*P*) holds.*

Lemma 5 $\curvearrowright$ *is LO.*

Proof Consider a fork $t \; {}_i\!\curvearrowleft s \curvearrowright_j u$.

- If $i = j$, then $t = u$, hence LO holds.
- If $i \neq j$, then we claim $t \curvearrowright_j v \; {}_i\!\curvearrowleft u$ holds, for some v. In case i and j are far enough apart ($|i - j| > 2$), this is trivial. Otherwise, one uses that if $m \geq 2$, then also $m + 1 \geq 2$ and $(m - 2) + 1 = (m + 1) - 2$.

Lemma 6 $\curvearrowright$ *is normalising.*

Proof We define a normalising *wave* strategy for a situation s, by recursion on the number of beans. In case there are no beans, then nothing can or needs to be done. Otherwise, pick an arbitrary bean, say at position i, remove it and normalise the resulting situation by a recursive call to the wave strategy. This yields some situation in normal form. Now note that if we drop the bean back in at position i resulting in, say, t, then s reduces to t.

- In case $t(i) = 1$, then t is a normal form and we are done.
- Otherwise, $t(i) = 2$ and we perform parallel bean steps from t as visualised in Figure 3. See how the wave front first extends from the position where the bean was dropped to the two borders. When a border is reached it is extended, the front moves back to the center and the waves die off.

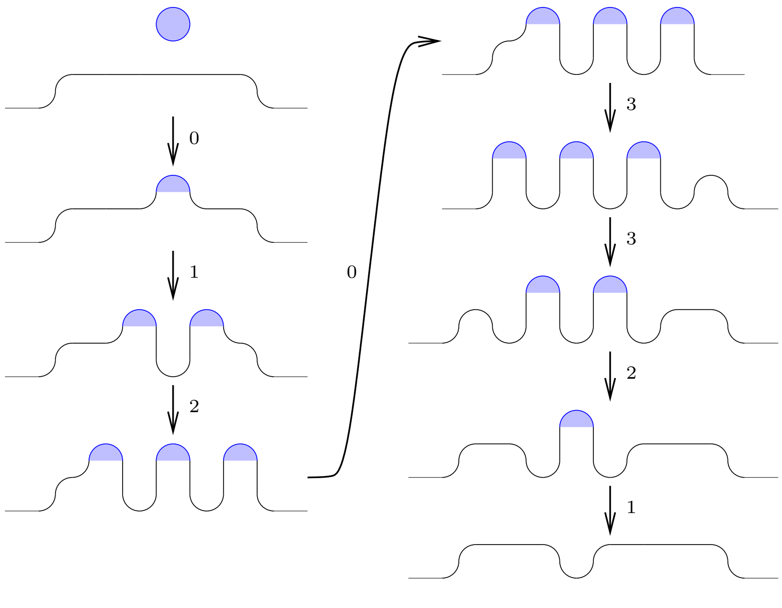

Figure 3: Bean drop

References

[Toy92] Yoshihito Toyama. Strong sequentiality of left-linear overlapping term rewriting systems. In *LICS '92*, pages 274–284, 1992.

A Variations

On termination One can vary in the way termination is established. We present one such variation combining ideas of Joost Joosten and Lev Beklemishev. The idea is to assign a weight function to situations such that the weight increases with every reduction step. When an upperbound on the weights can be established, termination follows.

Let there be m beans. Weigh a situation by a base-$m+1$ number in the obvious way, removing all (infinitely many) heading and trailing zeros. For instance, the situations in Figure 1 are represented in base 6 as 23 and 311, respectively. As one easily checks, by the choice of the base every reachable situation maps to a base-$m+1$-number, and any bean move will increase the weight establishing our first requirement.

To establish the second requirement as well, note that all reachable situations are weakly connected to the original situation in the sense that if in the initial situation some bowl contains a bean, then in any reachable situation, either the bowl itself or both its neighbours contain a(t least one) bean. As an easy consequence, any situation containing m beans cannot extend more than $2m$ bowls beyond its initial width, which gives the number represented by n consecutive ms as a (rough) upper bound on the weight, where n is the initial width plus four times the number of beans m.

However, note that the puzzle is not yet completely solved when termination is established; one still has to prove that all reductions to normal form have the same length. Interestingly, the route via [Toy92] as employed here, allows one to prove this stronger result, at the expense of checking linear orthogonality (two diagrams), but *weakening* the termination assumption to normalisation!

On topology One can vary on the puzzle by varying its topology. For instance, a solution analogous to the above was found (independently) by Hans Zantema in case the topology is a ring instead of a two-sided infinite array, under the condition that the number of beans is less than the length of the ring. The proof above goes through unmodified using that the condition implies the border of any group of beans as in Figure 3 is formed by *distinct* (empty) bowls.

On rules One may vary on the rules. E.g. if we allow for an infinite number of beans, but still require that for every rule the numbers of beans in its left- and right-hand sides are the same, one obtains universal computing power. To see this, view the array of bowls as the tape of a Turing machine. The ith symbol of its signature, say of total size m, can then be represented on this tape by a sequence of three bowls the first of which contains $m+1$ beans (a marker), the second i beans (the symbol), and the third $m-i$ beans (the complement). It is easy to see that by encoding blanks, the position of the head, and the rules of the Turing machine in a similar way, the Turing machine can be faithfully simulated. (The markers are used to enforce that rules can only be applied with an empty, i.e. everywhere 0, context situation C.)