Population protocols

1 Basics

Consider population comprising two species of agents X, Z. Recall that we assume the following fairness condition.

Definition 1 (Global fairness). *An execution E is fair if and only if for every configuration C occurring infinitely often in E, for every configuration C' reachable from C, C' also occurs infinitely often in E.*

Intuitively, it means that if something is reachable infinitely often, then it is actually reached infinitely often.

Question 1. Design a protocol such that for any population size n composed of x agents of species X, z agents of species Z, eventually exactly x agents out of the z agents of type Z are "killed", i.e., set to some special null state $\perp$. If $x \geq z$, then all the agents of type Z must be killed.

Question 2. Same question, but eventually exactly $2 \cdot x$ agents of type Z are killed. Generalize so that eventually $k \cdot x$ agents of type Z are eventually killed, where k is any predefined constant. How the state space depends on k ?

Question 3. We consider another species Y. Design a protocol such that for any population size n composed of x agents of species X, y agents of species Y and z agents of species Z, eventually exactly $x + y$ agents out of the z agents of types Z are killed.

Question 4. Can you design a protocol such that $x \cdot y$ agents out of the z agents of type Z are eventually killed ? (answer informally)

2 Presburger arithmetics

Consider an arbitrary (finite) set of species Σ. An input assignment is then represented by a vector $v \in \mathbb{N}^{\Sigma}$ specifying the number of agents of each species. Using this representation, there is a zero assignment (representing a configuration with zero agents), and assignments can be additioned component-wise. A predicate ψ is then represented by a function on input assignments taking values in $\{0, 1\}$: the function takes value 1 iff the input assignment satisfies the predicate. Recall that a predicate ψ is stably computable if there exists a population protocol such that fr any population size, for any input assignment v, eventually all the agents permanently output $\psi(v)$.

Question 5. Show that predicate of the form

$$a_1 \cdot \#X_1 + \cdots + a_k \cdot \#X_k = c \mod m$$

where the a_i's , c, and m are integer coefficients, is stably computable

Question 6. Show that boolean combinations of stably computable predicates are also stably computable.

Now, we assume the following lemma proven in [1].

Lemma 1 (Pumping lemma for population protocols). *Let ψ be a predicate over $\mathbb{N}^{\Sigma}$. If ψ is stably computable, then there exists a finite number of couples $(v_1, M_1), \ldots, (v_s, M_s)$ such that*

$$\psi(v) = 1 \Leftrightarrow v \in \bigcup_{i=1}^{s} (v_i + M_i)$$

where each $v_i \in \mathbb{N}^{\Sigma}$ is an input assignment, and each M_i is a set of input assignments containing the zero assignment, and closed under addition.

Question 7. Use the pumping lemma to show that the predicate

$$(\#X)^2 = \#Y$$

is not stably computable.

This last example confirms that the population protocol model is not strong enough to compute multiplication of variables. The following example insists on the fact that the coefficients must be integers. Indeed consider the predicate

$$P:\ \#X \leq \sqrt{2} \cdot \#Y$$

and assume that it is stably computable. Our goal is to derive a contradiction. By the pumping lemma, we thus have

$$M \underset{def}{=} \{(a,b) \in \mathbb{N}^2,\ b\sqrt{2} - a \geq 0\} = \bigcup_{i=1}^{s} (x_i + M_i)$$

For any $u = (a,b), v = (c,d)$ in $\mathbb{N}^2$, we write $u \cdot v = ac + bd$. Let $z = (-1, \sqrt{2})$.

Question 8. – Show that $x \in M$ if and only if $z \cdot x > 0$.
– Let $\epsilon = \min\{z \cdot x_1, \ldots, z \cdot x_s\}$. Show that there exists $y \in M$ such that $0 < z \cdot y < \epsilon$.
– Show that for some i, $(y - x_i) \cdot z < 0$.
– Considering the quantity $(x_i + n(y - x_i)) \cdot z$ for arbitrary large $n \in \mathbb{N}$, derive a contradiction.

References

1. D. Angluin, J. Aspnes, D. Eisenstat, and E. Ruppert. The computational power of population protocols. *Distributed Computing*, 20(4):279–304, 2007.