THE OPEN PROBLEMS FROM

Membrane Computing. An Introduction,
Springer-Verlag, Berlin, 2002
ISBN 3-540-43601-4

What follows is a quick "extraction" from the book, the 39 + 1 open problems identified in the text with Q1 – Q40, with a short formulation, with very few details, just a sort of reminder for the reader who is already familiar with the membrane computing, in particular, with the basic definitions and notations. However, recent information (at the level of September 2002) is added about the status of these problems – some of them were solved after having the "final" form of the manuscript. Needless to say that further solutions, to other problems, are very much welcome, and that the list has its "historical" and subjective limits, hence further problems might prove to be of a higher interest. Thus, contributions to this section of the P page are encouraged, bringing new information about the existing problems/solutions and/or proposing additional problems and research topics. (Gh. Păun)

First, the table itself from page 399 of the book, indicating the pages where the problems are formulated:

Problem	Page	Problem	Page
Q1	63 and 70	Q21	248
Q2	74	Q22	261
Q3	77	Q23	273
Q4	102	Q24	301
Q5	105	Q25	313
Q6	123	Q26	320
Q7	127	Q27	325
Q8	140	Q28	341
Q9	141	Q29	341
Q10	144 and 159	Q30	345
Q11	144	Q31	357
Q12	153	Q32	364
Q13	168 and 207	Q33	364
Q14	202	Q34	365
Q15	208	Q35	365
Q16	218	Q36	365
Q17	220	Q37	366
Q18	226	Q38	371
Q19	232	Q39	375
Q20	232	Q40	399

Q1. Which is the power of P systems with symbol-objects, using only non-cooperative and catalytic rules (and no other feature, such as priorities, membrane permeability control, etc)? Systems with non-cooperative rules only generate the length set of context-free languages (Theorem 3.3.2), those with cooperative rules are computationally universal (Theorem 3.3.3), but, for the intermediate case of catalytic rules, "we *conjecture* that $NOP_*(cat, tar)$ contains non-semilinear sets of numbers, hence it is not equal to NCF – but probably it is not equal to NRE either."

The second half of the conjecture was not correct (I confess, to my great surprise): in his paper from the pre-proceedings volume of Workshop on Membrane Computing, Curtea de Argeş, August 2002, pages 371–382 (from now on, WMC 2002; the volume is downloadable), Petr Sosik has proved that also the catalytic systems are universal. The proof uses 6 catalysts and Petr formulates the problem whether or not this number can be decreased.

The number of catalysts necessary for universality remains both as an open problem, somewhat replacing the problem from the book, and a motivation for introducing features such as priorities and membrane permeability control (the universality proofs from the book dealing with priorities and δ, τ actions use only one catalyst: see Theorems 3.4.3, 3.6.1).

Q2. Which is the power of P systems with symbol-objects, using only non-cooperative rules (hence not also catalytic rules), and a priority relation among them? They cover at least the length sets of ET0L languages (Theorem 3.4.4).

Q3. Investigate further the non-synchronized systems (where "non-synchronized" means adding to each set of rules a rule $a \to a$ for each object used by the system, and not "non-parallel"; sequential P systems appear in several places, but this is not the case with non-synchronized systems, where any degree of parallelism is possible, but not necessarily maximal).

Q4. Investigate further the P systems with symbol-objects where the communication is defined according to the concentration of objects in each region. There is one universality result in the book (Theorem 3.6.2), but using (many!) bi-stable catalysts. What about using "normal" catalysts, and additional features such as priorities, membrane permeability control, etc.

The last two questions have good biological motivations (besides being mathematically challenging).

Q5. Consider rules of the form $a \to v|_{negb}$, where the symbol b acts as an *inhibitor*: the rule $a \to v$ is not applied in region i if b is present in that region. Which is the power of systems using rules with inhibitors?

In the book it is proved that for promoters we get universality (Theorem 3.6.4). From Sosik's result, we know now that in the catalytic case we get universality without any permitting or forbidding condition. The case which remains is the non-cooperative one (as well as the problem of diminishing the number of catalysts necessary for universality: Theorem 3.6.4 uses only one).

Q6. Find characterizations of recursively enumerable languages by using symbol-objects systems with external output. One such characterization is provided by Theorem 3.7.3, but using catalytic systems (one catalyst only!) with rules having promoters: of the form $u \to v|_b$, with $u \to v$ used in a region only if b is present in that region). What about other classes of P systems with symbol-objects which are known to be universal?

The interest of the problem lies in the fact that we work with multisets inside the system and get a language as the result of a computation, which is a qualitatively different data structure.

A similar characterization of RE languages is obtained by A. Romero-Jimenez and M.J. Perez-Jimenez ([217] in the book), but using cooperative rules. Also, a related result is obtained by using the traces of objects – see below.

Q7. This is related to Q4: which is the power of P systems with the communication controlled by the concentration of objects and using a number of bi-stable catalysts bounded in advance? One catalyst is already sufficient for generating the Parikh sets of matrix languages (generated without appearance checking), but the universality proof (Theorem 3.6.2) uses a number of catalysts which depends on the size of the matrix grammar with which the proof starts.

Q8. Let us call "universal triple" a triple of numbers (m, r, s) such that $NOP_m(sym_r, anti_s) = NRE$. Several triples were shown to be universal (Theorem 4.2.1: (2,2,2), Theorem 4.2.2: (5,2,0), Theorem 4.2.3: (2,5,0), Theorem 4.2.4: (3,4,0), Theorem 4.2.5: (3,0,2) – with $N'RE$ here), one one conjectures that "some of these results can be improved".

This time the conjecture was correct (it was rather expected to be so): Pierluigi Frisco and Hendrik Jan Hoogeboom have improved these results in their paper from the same WMC 2002 (pages 237–148). Here are their universal triples: (1,1,2), (4,2,0), (2,3,0), (1,0,2) (with $N'RE$ in the last case).

They also conjecture that (3,2,0) is universal. For systems with two membranes, the cases (2,2,1) and (2,2,0) remain of interest.

Q9. This is related to the previous one: clarify the power of systems with only one membrane, for small triples, if possible, of the form (1,r,1). Particular case: does the family $NOP_1(sym_2, anti_1)$ contain infinite sets of numbers? (After a number of unsuccessful attempts I was inclined to conjecture that the answer is negative, but maybe a clever "symport-antiport program" can do better.)

Q10. Theorem 4.3.1 shows that the triple (3,2,0) is universal for the case when the symport rules have promoters or inhibitors, objects which have to be present, respectively absent, in/from the region of the membrane with which the rules is associated (and applied). The case of promoters and inhibitors associated with antiport rules has remained open.

If the above mentioned conjecture of Frisco and Hoogeboom will be confirmed, then we do not need systems of type (3,2,0) in order to get universality, but the question remains whether a smaller triple is universal in the case of using promoters and/or inhibitors (associated with symport and/or antiport rules).

Q11. This is somewhat meant to stress that the P systems with symport/antiport deserve a special interest, because of their qualities (bio-motivation, elegance, conservation law, etc): what about using other ingredients in these systems, such as δ, τ actions (why not also priorities)?

Q12. Consider a distinguished object (the "traveler") and follow its trace across the membranes of a system (with symport/antiport) during a halting computation, in the form of the sequence of labels of regions visited during the computation. Take a weak coding of this sequence, for more flexibility. Again we get a language as the result of computations in a system dealing with symbol-objects. Theorem 4.4.4 characterizes the recursively enumerable languages by using P systems with permitting or forbidding conditions and with antiport rules of arbitrary weight. The question is to improve this result from these points of view,

...and this has been already done by P. Frisco and H.J. Hoogeboom in the above mentioned WMC 2002 paper: systems without conditions about the use of rules can trace-characterize RE. More specifically, it is proved that any language over an alphabet with n symbols can be the trace language of a system corresponding to any of the next triples: $(n+1,0,2)$, $(n+1,3,0)$, $(n+2,2,0)$. So, there is not too much to do around this problem. Maybe to try to improve the previous results. Or to see which is the power of systems with shorter symport or antiport rules. (Are the families of languages trace-generated by them strictly included in RE?) Or to define trace languages for other types of P systems. (Actually, it seems that there is some further efforts needed here...)

Q13. From now on, some problems about systems with string-objects. The one with the unlucky number asks whether or not the matrix languages (generated without appearance checking) can be generated by rewriting P systems (without any additional feature, such as priorities, membrane permeability control, etc) with less than 4 membranes. The book gives a characterization of MAT by system with 4 membranes – Corollary 5.1.2. A better result could lead to improvements of other results about rewriting P systems.

Q14. Rewriting P systems with replication characterize RE when using 6 membranes (Theorem 5.2.12), which is a "huge" number among the universality results. Can this result be improved? *Conjecture*: yes.

Q15. The case of rewriting P systems with a parallel rewriting of string-objects, although biologically motivated (remember the L systems), raise many problems when defining the communication, because of the conflicting target indications of the used rules. The book considers only the "partial parallelism", of S.N. Krishna and R. Rama ([114] in the book), and the problem is formulated to systematically investigate the parallel processing of strings.

In the meantime, the topic has been addressed in a series of papers by D. Besozzi, C. Ferretti, G. Mauri, C. Zandron – one in the WMC 2002 volume (but still there is something to work here).

Q16. Theorems 5.3.1 (characterization of RE by means of P systems with string-objects processed by splicing rules of diameter (1,2,2,1), using two membranes) and 5.3.2 (this characterization can be obtained for systems with global rules, the same for all membranes), are using the extended feature, a terminal alphabet for selecting the accepted strings. Is it possible to get rid of this feature, and to use non-extended systems (but not increasing the number of membranes)?

Q17. Extended P systems with string-objects using splicing rules and the so-called *immediate communication* (each result of a splicing should leave the region where the splicing was done, going *in* or *out*, nondeterministically choosing the direction), characterize RE, but, almost unique in the book!, using arbitrarily many membranes (two for each rule of the starting type-0 Chomsky grammar from the proof). Improve this result, by finding a well defined bound on the number of membranes. (Anyway, because of the universality, the hierarchy on the number of membranes is not infinite, so the question is to find a small bound on the number of membranes.)

Q18. A problem similar to Q16, this time for P systems with string-objects processed by means of contextual rules: can RE be characterized by contextual P systems with finite selection but without using a terminal alphabet for selecting the generated strings? The *conjecture* was formulated that the answer is positive.

Q19. Investigate language-theoretic properties of families of languages generated by systems which are not known to be universal (and which are believed/hoped not to be universal). In particular, investigate closure properties.

Q20. A more general problem, which I recall it here as in the book: "Also of interest are *absorption results*, as those from [142] mentioned above. In general, given a (finite or infinite) hierarchy of language families, $FL_m, m \geq 1$, and an operation g with languages – say, an n-ary one – such that the families FL_m are not (known to be) closed under the operation g, it is of interest to find results of the following form: *if* $L_1, \ldots, L_n \in FL_m$, *then* $g(L_1, \ldots, L_n) \in FL_{m+p_g}$, *for some* p_g *depending on* g. When $p_g = 0$ we have a usual closure property."

Q21. Let us pass now to tissue-like P systems, with the membranes arranged in the nodes of a graph, with or without direct communication channels between them. In the "with" case (Theorem 6.2.3: three membranes, each communicating with each other, and Theorem 6.2.4: four membranes, communicating two by two) we get universality. What about membranes without any direct link, swimming in a common environment? The *conjecture* is that such systems cannot compute "too much". However, after having the results of Frisco, Hoogeboom mentioned above about cell-like symport/antiport systems, the problem should be formulated in a more precise manner. Indeed, a usual system with one membrane can be considered a tissue P system, hence the fact that the triple (1,1,2) is universal already solves the problem in its general formulation (that is, without restrictions on the weight of the rules).

By the way, both Theorems 6.2.3 and 6.2.4 use symport rules of weight 2 and antiport rules of weight 1, hence they are not "touched" by the results of Frisco and Hoogeboom.

Thus: which is the power of completely unstructured tissue-like P systems with symport and antiport rules with weights (r, s) smaller than (1, 2) or incomparable with this vector?

Q22. Neural-like systems with certain numbers of cells and certain numbers of states in each cell characterize NRE (Theorem 6.3.1: 2 cells, 5 states, Theorem 6.3.2: 4 cells, 4 states, Theorem 6.3.3: 2 cells, 2 states, but using cooperative rules). Are these results optimal in the number of cells and states?

Q23. Section 7.1 defines complexity classes for membrane computing, in a specific way, and concludes with: "in the following sections we will prove that there are **NP**-complete problems which belong to $\mathbf{LMC}_X$, for various classes X of membrane systems. Typically, the problems considered here are SAT (the satisfiability of propositional formulae given in the conjunctive normal form), and HPP (the existence of a Hamiltonian path in a directed graph). From the previous observation it follows then than $\mathbf{NP} \subseteq \mathbf{LMC}_X$: in a polynomial time, we can pass from any problem A from **NP** to SAT, then in a polynomial time we can construct a family of systems Π which solves SAT in linear time, hence Π also solves A *in linear time*, because the passing from A to SAT is included in the polynomial time for constructing Π.

Admittedly, this is a little like cheating, but again the reduction time is counted for a Turing machine; what about reductions done by means of membrane systems? Are there problems which can be reduced to SAT by means of membrane systems, in linear time? How large is their family? Then, what about the time necessary to construct the reduction systems themselves? The question can be iterated. Anyway, if the reader does not like them, then (s)he may ignore the classes $\mathbf{LMC}_X$, and consider only the classes $\mathbf{PMC}_X$, which are 'fair': polynomial reduction, polynomial construction of the membrane system, polynomial time of work in the membrane system.

The theoretical study of the classes $\mathbf{LMC}_X, \mathbf{PMC}_X$, or of other possible similar classes, for various types X of P systems, is beyond the scope of this book, and it remains as a *research topic*. (An important question: do the classes $\mathbf{LMC}_X,\ \mathbf{PMC}_X$ depend on X, or are they robust from this point of view – at least for the three types X of systems we will consider below?) Here we confine ourselves (mainly) to looking for P systems able to solve SAT and HPP as fast as possible, taking into account only the time spent by the P system itself."

Q24. I take again some paragraphs from the book, this time from the end of Section 7.2: "when we defined the classes $\mathbf{LMC}_X, \mathbf{PMC}_X$, we did not pay much attention to non-determinism – provided that the systems are confluent, in the precise sense defined at the beginning of this chapter. These observations raise the natural question whether or not we can find **NP**-complete problems which are in $\mathbf{PMC}_X$ for classes X of P systems with active membranes which do not use membrane division, hence using only rules of the forms (a), (b), (c), (d). The problem is non-trivial, because by means of rules of type (a) we already have the possibility to create exponentially many objects in linear time: consider rules of the form $[_i a \rightarrow aa]_i^\alpha$. However, the objects created in this way are placed in the same membrane (or in the bounded number of membranes of the initial configuration of the system), which seems to be a difficulty in using them efficiently. Structuring the multisets of objects in a better manner, for instance by dividing membranes *or* by creating new membranes, produces the extra power sufficient for solving **NP**-complete problems in polynomial time."

Several recent paper by Sevilla group (Mario Perez-Jimenez, Fernando Sancho-Caparrini, Alvaro Romero-Jimenez) are related to the last two questions – but not directly addressing them. For instance, they find a nice reduction of the **P**$\neq$**NP** conjecture to a problem related to "decision P systems".

Q25. Is it possible to construct a uniform family of P systems with membrane creation (without priorities or membrane thickness control) which is able to solve SAT in linear time? (At the beginning of Subsection 7.3.1 one gives such a solution to SAT, but the P systems have exponentially many rules – of a "uniform" form).

Q26. The book considers only symbol-object P systems with membrane creation as a framework for writing "P algorithms" for solving **NP**-complete problems in polynomial time. What about considering the membrane creation feature in systems with string-objects?

Q27. Besides the three already well-investigated ways to get exponential space, membrane division, string replication, and membrane creation, another idea is to start with an exponential (even arbitrarily large) space provided "for free" in advance, but not "too structured" (whatever that means) and to *activate* the necessary portion of this space during a computation, so that a polynomial solution to a hard problem is obtained by activating "enough" space. Investigate this possibility in some details – for various possibilities of formalizing the involved notions. One such possibility is that from the book, another one was proposed by Eugen Czeizler during WMC 2002 (pages 193–206 in the pre-proceedings volume).

Q28. Find a class of P systems (defined in a "decent" manner, that is, not in such a way to be visible that it is defined in order to get the result in an easy manner) for which the number of membranes induces an infinite hierarchy of the generated sets of numbers (or languages). For instance, using traces is not a "decent" solution, because it is clear that systems with m membranes can generate languages over alphabets with less than m symbols, hence the cardinality of the alphabet directly induces the infiniteness of the hierarchy.

Q29. Investigate systematically *unary* P systems (systems with symbol-objects, using only one object; catalysts are allowed).

Q30. This problem falls in the general category of issues related to alternative ways of defining the output of a computation. Here are the words from the book: "In all previous chapters of the book, the purpose of a computation was related to the objects present in the regions of the system, symbols or strings, while the membrane structure was only a tool used in the computation. Let us reverse the perspective, and take the membrane structure – and the tree describing it – obtained at the end of a halting computation as the result of that computation, ignoring at the end the objects present in the system or sent out of it. In the case of systems which use only the action of membrane dissolving, the set of all trees associated in this way with a system Π is finite, but this is not the case when we can make use of membrane division or of membrane creation. Moreover, further operations with membranes can be considered, for instance modelling the biological phenomena of exocytosis and endocytosis (in Section 8.5 we will briefly discuss some operations of this kind). We do not address the general problem here, it remains as a promising *research topic*, but we deal with a related one, about a possibility of representing the context-free languages by making essential use of the membrane structure."

Q31. Again a research topic, not a punctual problem: "There are numerous distributed – parallel or not necessarily parallel – models in computer science. Many of them have a completely different style and goal than membrane systems, but many of them are computing devices in the Turing sense, often dealing with strings (languages) or with numbers. The comparison of such models with membrane systems is a potentially interesting and fruitful *research topic*. The first 'temptation' is to simulate a mechanism in terms of the other mechanism, thus transferring results from one area to another area (because the power of P systems of various types is well known and because this power is in many cases equal to that of Turing machines, it is possible to transfer in this way universality results from the membrane systems area to other formalisms). Perhaps more important is the transfer of notions and ideas from one domain to another domain."

The book briefly discusses the connection with X-machines, addressed in a series of papers by Marian Gheorghe, Mike Holcombe and their collaborators. The possible link with Petri nets was mentioned at the end of their WMC 2002 paper by Pierluigi Frisco and Sungshul Ji (pages 249–264 in the volume), without entering any details. This particular connection seems particularly interesting, as the notion of a multiset is central both to Petri nets and to P systems. It has been also briefly explored by Adam Obtulowicz in his WMC 2002 paper (only a two pages abstract appears in the volume, at pages 331–332) – see below.

Subsection 8.6.6 ("Further Research Topics") is explicitly devoted to the formulation of open problems and topics which seem interesting/promising, hence I recall it almost completely.

Q32. "So far, the membrane systems were always considered in the *generative sense* (one starts from an initial configuration and one collects all outputs, numbers or strings, which the system produces), and as *decidability devices* (one encodes a problem in the initial configuration and one looks for a specified object, *yes*, in order to have an answer to the problem). The generative approach corresponds to grammars, the decidability is automata-like, but not completely. What about using P systems as *accepting* devices? The idea is simple: consider a system Π (membranes, rules, objects), and introduce in its initial configuration a further multiset, M, of objects (we now consider the symbol–object case). We say that M is accepted if, starting from this augmented configuration, the system halts (or halts and produces a special object *yes*, as in the decidability case). Then, the set $N(\Pi)$ consists of all numbers describing the cardinality of all multisets M 'recognized' in this way by Π."

This problem has been already addresses in a series of papers, one of them in the same WMC 2002 volume (pages 177–192): Erzsebet Csuhaj-Varju and Gyorgy Vaszil considers *P automata*, in the form of P systems using only symport rules of the form $(x, in)|_z$ (hence with permitting contexts, in the form of multisets), with the transitions defined in a sequential manner (one rule only is used in each region), and with the sequence of multisets brought into the system from the environment during a computation being considered as the recognized string (see the paper, for some other details of the definition). Each RE language is proven to be recognized by such a system with seven membranes.

A variant was proposed by Mutyam Madhu (manuscript at the moment of writing these notes), also considering state-symbols in each region and rules of the form $(qy, in)|_{px}$, which also change the state p with state q after application. Systems with two membranes are found to be universal – but recognizing sets of numbers.

Then, a usual P system with symport/antiport rules, recognizing the sequence of terminal symbols taken from the environment during a computation, were considered (and proved to be universal even in the case of only one membrane), by Rudolf Freund and Marion Oswald (paper to be published in *Bulletin of the EATCS*, October 2002).

The accepting P systems still ask for (and deserve) serious further efforts, as many problems are still unsettled in this area.

Q33. "A second technical – but general – research topic, somewhat suggested already in the book, is to systematically investigate the case of systems with a small number of membranes. This means systems that are not necessarily universal, hence their properties are not known, as direct consequences of the equality with NRE or RE. The closure properties of such families have already been mentioned, but other classic questions can be raised: comparison with known families from Chomsky and Lindenmayer hierarchies, decidability properties, descriptional complexity (do the numbers of symbols or of rules, the lenght of rules, and other parameters of this type introduce infinite hierarchies in the family of number sets/languages generated by systems by a given type, with a given number of membranes?). Suggestions from the well-developed area of descriptional complexity for Chomsky grammars and languages can be useful."

Q34. "Let us now consider a problem with a 'practical' motivation, but with an important technical content. When we introduced the systems with symport/antiport we mentioned that they observe the conservation law: nothing is created and nothing is destroyed, all objects are moved only from one compartment to another compartment. This property can be easily extended to all types of systems dealing with symbol–objects provided that cooperative rules are allowed, by adding dummy objects to the rules, just to balance the number of objects on the two sides of rules (the dummy objects never evolve and are not considered when defining the output). We do not enter into further details here.

Not so easy to handle seems to be another important biochemical suggestion, the *reversibility*. Many reactions are reversible, and can go in either direction (possibly depending on a change of reaction conditions, for instance temperature – hence energy availability). In dynamic systems, reversibility is somewhat opposed to non-determinism: we have to uniquely find the previous configuration of the system starting from the present configuration. Strictly chemically speaking, reversibility means reversing the direction of an equation. What about reversible P systems in the dynamic systems sense, what about 'local reversible P systems', with the rules allowed to be used in both directions (for each $u \to v$ we have also to add $v \to u$ to the same region)? (By the way, it has been known for many years that reversible Turing machines are computationally universal. Can reversibility be brought 'for free' from the Turing machines area to the P systems area?)"

Q35. "A further research topic, both mathematically and practically interesting, is related to the fact that membrane computing comes from biochemistry, where most processes are non-deterministic, and the results are only approximately/probabilistically known/true. Up to now, only 'crisp' mathematics has been used in the P systems area. What about 'approximate' mathematical approaches, using probabilities, fuzzy sets, or rough set theory? (We refer to [204] for the rough set theory; the basic idea is to approximate a set from the interior and from the exterior, by converging unions of equivalence classes.) What about 'approximate' computing, whatever this can mean?"

Again, some attempts to address this topis have been reported. One is by Mutyam Madhu, in a manuscript not yet finished, where probabilities are associated with rules in the style of probabilistic grammars (one specifies the probability to use a rule r' after another rule r, and one accepts only the strings with the probability of a derivation greater than a given threshold – where the probability of a derivation is the product of the probabilities of all rules used).

A different approach is chosen by Adam Obtulowicz in his WMC 2002 paper mentioned above: one introduces both stochastic P systems (starting from stochastic Petri nets already known in the specific literature) and randomized P systems, used for implement randomized algorithms (which can solve hard problems in linear time, with a high enough probability, making use of a sub-exponential workspace).

Q36. "A promising research topic, again with a practical flavour (see also the discussion from Section 9.5), is to 'translate' the idea of an eco-grammar system, as introduced in [45], in terms of membrane systems. In short, an eco-grammar system consists of a set of agents, evolving in a common environment. The agents have their own rules for internal evolution, as well as rules according to which they can act on the environment or on other agents. In turn, the environment has its own evolution rules, and, depending on its state, it determines the state of the agents (the set of rules active at a given moment). In [45], both the agents and the environment are described by strings, hence the evolution rules are rewriting rules (used either in the sequential or in the parallel manner). In the framework of membrane computing, the agents can be membranes, placed into a larger membrane which represents the environment. Bearing in mind an eco-system, both the agents and the environment should be finite, but it could be of interest to have one more region around the environment (the 'world', the 'universe'), infinite, providing raw materials to the environment, but not active otherwise, for instance not in the range of communication of agents. The agents can exchange objects with the environment (for instance, by means of symport/antiport rules), the environment can send objects into the universe, and can grab objects from the universe (an object which exits the environment is 'lost'). For the agents it is probably better to have rules as simple as possible (following somehow the style of *colonies* – see [105]), for instance only renaming rules, of the form $a \to b$, or symport/antiport rules of a minimal weight; it is expected that a careful choice of innocent-looking ingredients will produce an interesting emergent behaviour, as often met in the grammar systems (colonies included) area."

Q37. "We end this short list with another very general question, actually a list in itself of 'exotic' problems. What about introducing some quantum computing ideas in membrane computing (or conversely)? Imagine, for instance, that the objects b and c produced by a rule $a \to bc$ are *entangled*, each one "senses" the other, irrespective of where they are placed after several steps of a computation. Can this be used in our framework – in particular, can it be used for speeding up the computations? The entanglement is only one quantum notion; many others can be considered: quantum object, superposition, teleportation.

In the same 'exotic' area, what about the reproduction of a membrane system? The problem has a long history for cellular automata (S. Ulam, J. von Neumann, etc.). By the division of a membrane we have directly introduced reproduction at the level of inner membranes, but this is not allowed for the skin membrane. Is there any non-trivial way to approach this issue?

What about also trying to go 'beyond Turing'? We have mentioned in the Introduction that in the framework used in this book this is not possible: the systems are always discrete, finite, hence recursively enumerable. However, one can introduce ingredients which are not Turing coverable, such as infinite computations, continuous elements, accelerations (consider a membrane which *learns*, so that it works normally the first time unit, but the job for the next time unit is done in half the time – while the other membranes work normally – the job for the third time unit is done in half of the time necessary for the previous step, and so on, hence using $1/2^n$ time units for the nth step of the computation). An easy way to 'compute' non-recursively enumerable languages is to work on the complement, in the sense of the so-called *computing by carving* from [184], which is an abstraction of the filtering procedures from DNA computing; can the computing by carving also be used in membrane computing?"

Q38. This is about the study of conformon-based P systems (see the book for what a conformon is supposed to be; very reductionistically, it is an object of the form (name, number), where the name can be from an alphabet and the number is a natural one, giving, for instance, "the energy" associated with the object), but the framework can be larger, related also to the dynamic system behavior of the P systems. I skip the details here.

Q39. Relate more membrane computing with ambient calculus, for instance, via systems with gemmation. References can be found in the book.

Q40. Also the (non)formulation of this problem can be found in the book, at page 399 according to the table from the beginning of this note... Actually, it is a meta-problem, of the type: find good research problems (in the membrane computing area)...