# Chapter 9

# Models of Self-Assembly

*Profound study of nature is the most fertile source of mathematical discoveries.*

Fourier, *Analytical Theory of Heat*

## 9.1 Introduction

One does not have to understand the science of optics to appreciate the beauty of a rainbow; but it helps. In the same way, one need not master the mathematics of self-assembly in order to appreciate the power of the concept; but here too, it helps. In the first two parts of this book we focused on descriptions of self-assembling systems. At times, we made use of mathematics, but ultimately our focus was on experiment rather than theory. In this chapter, we shift our focus and examine the various theoretical approaches to understanding the phenomenon of self-assembly.

There are as many different approaches to mathematically modelling self-assembly as there are examples of physical self-assembling systems. In the end, the type of model one constructs depends upon the type of question one wishes to answer. These questions can vary wildly. At one end of the spectrum, we have models built to illuminate the behavior of one specific self-assembling system. Such a model can have great utility. If accurate, it can help reduce the number of costly or time consuming experiments one needs to conduct. It can clarify the role of various parameters in the system and give a picture of parameter space that might otherwise be inaccessible. At its best, it can clarify a complex situation, help guide experiment, and identify new experimental regimes to be explored. At the other end of the spectrum we have abstract models of the phenomenon of self-assembly. These models are usually divorced from any particular experimental system; rather they seek to capture the behavior of some large class of self-assembling systems. These models too, can have great utility. At their best, they can help us answer "What is possible?" types of questions. Is it possible to self-assemble a Sierpinski Gasket in a system containing only two tile types? Is it possible

to self-assemble a cell given infinitely many tile types? These are the type questions that abstract models are best at answering.

However, there is no hard and fast boundary between these types of mod Models of a particular physical system are often found to apply to other tems, systems that at first might seem unrelated. These models are perh more abstract than we initially thought. Abstract models take their insp tion from physical systems and in seeking to capture general principles, o end up capturing real behavior remarkably well. At times, a model that tially seemed abstract may end up being physically realizable, and end showing us a new route to self-assembly.

Nor is there any mathematical distinction between these types of mo The equations of continuum mechanics can help us develop a detailed scription of the shape of a meniscus, but they can also be implemente a computer, governing the behavior of fictitious particles that have no c terpart in the real world. Seemingly pure branches of mathematics, suc graph theory, which lends itself nicely to several abstract approaches, lends itself nicely to robotic control schemes for real world engineered p cles. Similarly, computer simulation plays an important role in the anal of every kind of model. Both physically driven models and abstract mo have a tendency to become analytically intractable. In both cases, numer simulation becomes a necessity.

Nonetheless, for clarity in the discussion, we will make a distinction tween these two types of models. We'll divide this chapter into two m sections. In the first, *Physical Models*, we'll describe approaches that close to one physical system or some small subclass of physical systems. the second, *Abstract Models*, we'll examine approaches to "What is possib type questions.

In Section 9.2, *Physical Models*, we begin with a mathematical mod the structured surfaces discussed in Chapter 6. This model asks the ques What can be accomplished if an electric field is used to manipulate the imal energy surfaces of Chapter 6? This model is very much at the "s experimental system" end of the spectrum. Through this model we'll see key parameters in a problem may be identified and how a model can hel understand parameter space and suggest experimental directions. Next, examine a model that attempts to explain why the helix is such a fam structural motif in nature. In contrast to the structured surface model, model focuses on a class of self-assembling systems rather than on a s experimental setup. We'll see how such a model can be useful, both to insight into a broad problem, and to actually predict experimental res For our third model, we'll return to the first system we discussed in Ch 6: the self-assembling tile system of Hosokawa et al. The model we'll di is drawn from their original paper [62] describing their experimental and oretical results. We'll see how a model inspired by chemical reaction kin can capture the behavior of a tile based self-assembling system. Finall this section, we'll discuss the so-called *waterbug model*, due to Eric Kla



This final model is again unattached to any particular physical system, but is inspired by a class of such systems. With this model, we'll see how theory can aid in the design of physical systems.

In Section 9.3, *Abstract Models*, we focus on three abstract approaches to modelling self-assembly. The notion of a *conformational switch* is the focal point of the first of these models. We encountered conformational switching in Chapter 3 when we discussed the tobacco mosaic virus. We also encountered this notion when we discussed proteins and again in part two of this book in the context of several different engineered systems. The model of conformational switching presented in this section attempts to characterize the power of a conformational switch to encode for a given assembly sequence. The second model we consider is based on the notion of a graph grammar. This model generalizes the conformational switch model and within the context of the model is able to provide a constructive solution to the backward problem of self-assembly. The final model we consider is the *Tile Assembly Model*. This important model provides the link connecting self-assembly and computation. We'll see how this model has been used to explore the question of complexity of a self-assembling system and how this model provides a promising route to programmed self-assembly.

One final note before we begin – to understand the details of every model discussed in this chapter requires a broad mathematical background. Here, we won't focus on these details. Rather, we'll attempt to provide a sense of the thinking behind the model, the questions it seeks to address, and the importance of the answers to those questions. Further, where it seems most appropriate, we'll fill in the mathematical background needed to understand the basics of the model. However, this may not always be enough. If you find the details of a particular model in this chapter to be confusing or inaccessible, skip them. You should still be able to get a sense of the model. If you still find a particular model to be heavy going, skip it entirely. The subsections in this chapter are mostly independent.[1] I encourage you to find a modelling approach and a set of questions that excites you, and to continue from there.

## 9.2 Physical Models

In this section, we present four models that are focused on one particular physical system, or on a small subclass of such systems. We've chosen these models as representative examples. Many more such efforts exist. We refer the reader to [12, 64, 88, 148] for information on similar approaches to modelling self-assembly. Further, if you return to part two of this book, you'll find that the references to the experimental systems discussed often contain models of these systems.

### 9.2.1 Modelling Structured Surfaces

In Chapter 6, we examined an approach to self-assembly, pioneered the Whitesides group, that we called *structured surfaces*. Recall that in approach, droplets of PDMS were placed between rigid plates and that changing the wettability of these plates, the density of the surrounding fl and the orientation of these plates, the shape of the surface assumed by PDMS droplet could be controlled. In turn, the PDMS could be cross-lin or solidified, and hence objects with interesting shapes constructed with the use of a mold or template. In their original article, the Whitesides gr [70] also conjectured that electric or magnetic fields could be used to ob an additional level of control over the shape of these surfaces. The mode this section explores this idea. This model is due to Derek Moulton; fur details may be found in [93].

To begin, Moulton replaced the PDMS droplet of Whitesides by a soap spanning two identical concentric rings. Working with a soap film allo him to remove the volume constraint inherent in the PDMS system and w with simple boundary conditions at the endpoints. Additionally, Moulton able to carry out experiments with the system he devised. As we discus in Chapter 6, a soap film spanning two rings naturally forms a catenoid, see how this shape could be manipulated, Moulton added an outer electr to the system surrounding the two ring soap film structure. By applyin voltage difference, an electric field could then be created in the gap betw the soap film and the electrode. The basic setup of this system is show Figure 9.1.

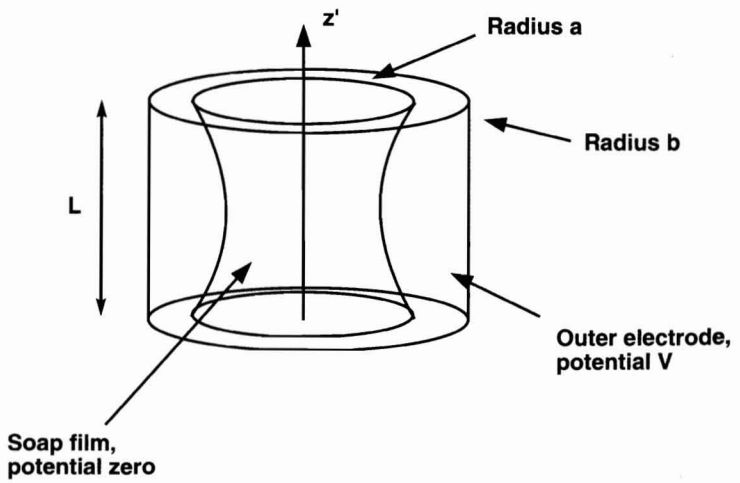


**FIGURE 9.1**: The geometry of the soap film and electrode system.



By adding an electric field, Moulton added a second energy to the problem. The soap film will attempt to minimize its surface energy, we saw this in Chapter 6. But, there is also energy stored in the electric field. The shape selected by this system will minimize the total energy, i.e., the sum of the surface and field energies.

To derive an equation that would predict this shape, Moulton first needed to derive expressions for each of these energies. In sketching this derivation, we use the notation of Figure 9.1. Note that in the figure, the shape of the surface is specified by the function $u'(z')$ and the electric field is specified in terms of the potential function $\psi'(r', \theta, z')$. The potential is assumed constant on the outer electrode and on the soap film. The rings supporting the film are of radius $a$, they are placed a distance $L$ apart, and the outer electrode has radius $b$.

In electrostatics, the electric field, $\vec{E}$, can be specified entirely in terms of the potential function through $\vec{E} = -\nabla\psi'$. Since in the absence of free charges, the field satisfies

$$\nabla \cdot \vec{E} = 0 \tag{9.1}$$

the potential, $\psi'$, satisfies the Laplace equation

$$\nabla^2\psi' = 0. \tag{9.2}$$

The fixed potential conditions on the soap film and outer electrode translate into the boundary conditions

$$\psi'(b, \theta, z') = 0 \tag{9.3}$$

$$\psi'(u'(z'), \theta, z') = 0. \tag{9.4}$$

Now, note that the soap film surface, $u'(z')$, must also satisfy boundary conditions. In particular,

$$u'(L/2) = u'(-L/2) = a. \tag{9.5}$$

Notice that we already have four parameters in this problem, $a$, $b$, $L$, and $V$. In models such as this, it is convenient to introduce nondimensional variables. This not only simplifies the discussion, but also helps one uncover the relative importance of various terms in the model and helps reduces the dimension of parameter space to a manageable level. This process is called *nondimensionalization*. You can find a thorough explanation of this process in [99]. Here, we introduce the variables

$$z = \frac{z'}{L}, \qquad r = \frac{r'}{b-a}, \qquad \psi = \frac{\psi'}{V}, \qquad u = \frac{u'}{a}. \tag{9.6}$$

With these substitutions, Equations (9.2) through (9.4) become

$$\frac{\partial^2\psi}{\partial r^2} + \frac{1}{r}\frac{\partial\psi}{\partial r} + \epsilon^2\frac{\partial^2\psi}{\partial z^2} = 0 \tag{9.7}$$

$$\psi = 1 \quad \text{at} \quad r = \frac{b}{b-a} \tag{9.}$$

$$\psi = 0 \quad \text{at} \quad r = \frac{a}{b-a}u(z). \tag{9.}$$

This set of equations is in non-dimensional form. The key dimensionl parameter that arises here is $\epsilon = (b-a)/L$. Physically, $\epsilon$ is an aspect ra comparing the size of the gap to the length of the device. In his experimen system, Moulton found that $\epsilon$ was in fact a small parameter. This fact be used to simplify the analysis below.

Now, the energy stored by the electric field is given by a volume inte taken over the region between the soap film and outer electrode. In particul

$$\text{Electrostatic Energy} = -\frac{\epsilon_0}{2}\int |\vec{E}|^2 = -\frac{\epsilon_0}{2}\int |\nabla\psi'|^2 . \tag{9.}$$

Notice that to compute this energy, we need to know $\psi'$ or equivalently This means that we need to solve Equations (9.7) through (9.9). Howe this is not easy. First, the shape of the domain is not regular, and seco the shape of the domain is not even known; it depends on $u(z)$. But, approximate solution for $\psi$ can be obtained by exploiting the fact $\epsilon$ is a sm parameter. This requires *asymptotic analysis*; we'll skip the details. Gi this approximate solution, an expression for the electrostatic energy can obtained. It can be further simplified by using the divergence theorem[2] a the boundary conditions. At the end, we obtain

$$\text{Electrostatic Energy} = -\pi\epsilon_0 V^2 L \int_{-1/2}^{1/2} \left(\log\frac{\delta}{u(z)}\right)^{-1} dz. \tag{9.}$$

Note that here, the dimensionless parameter $\delta$ is the ratio $b/a$ of the radi the outer and inner cylinders.

With the electrostatic energy in hand, we are halfway there. We still ne an expression for the surface energy in the problem. But, we know that t energy is simply proportional to the change in surface area, and we already s how to compute this in Chapter 6. In terms of our nondimensional variabl we can write the surface energy as

$$\text{Surface Energy} = 2\pi TLa \int_{-1/2}^{1/2} u\sqrt{1+\sigma^2 u_z^2}\, dz. \tag{9.}$$

Here, the subscript on the $u$ denotes differentiation with respect to $z$, t dimensionless parameter $\sigma$ equals $a/L$, and $T$ is the film tension.

Finally, forming the sum of our two energy expressions and dividing $2\pi TLa$ we obtain an energy functional for our system

$$E[u(z)] = \int_{-1/2}^{1/2} \left(u\sqrt{1+\sigma^2 u_z^2} - \frac{\lambda}{\log(\delta/u)}\right) dz. \tag{9.}$$



Here, the dimensionless parameter $\lambda$ is given by

$$\lambda = \frac{\epsilon_0 V^2}{2Ta}. \tag{9.14}$$

This parameter measures the relative strengths of the electrostatic and surface energies in our system. This function, $E[u(z)]$, is not so far from other energy functionals we've encountered in this book. It maps the shape of our surface, $u(z)$, to a real number denoting the energy of the system. As usual, we claim that nature chooses the shape that makes this energy as small as possible. Fortunately, this functional is in a form such that the Euler-Lagrange equation introduced in Chapter 6 can be applied. Doing so, we find that $u(z)$ satisfies

$$\frac{1+\sigma^2 u_z^2 - \sigma^2 u u_{zz}}{(1+\sigma^2 u_z^2)^{3/2}} = \frac{\lambda}{u\log^2(\delta/u)} \tag{9.15}$$

plus the boundary conditions

$$u(1/2) = u(-1/2) = 1. \tag{9.16}$$

Now, a complete and detailed analysis of Equations (9.15) and (9.16) is beyond the scope of this book. These details are developed further in the exercises and can be found in [93].[3] Rather, here, let's make a few observations about this model, about how it can be used, and why such a model is important in self-assembly.

First, note that the left hand side of Equation (9.15) is actually the negative of the mean curvature operator we encountered in Chapter 6. Hence, Equation (9.15) can be rewritten as

$$Hu = \frac{-\lambda}{u\log^2(\delta/u)}. \tag{9.17}$$

This is interesting to note because this represents a generalization of the standard equation of constant mean curvature surfaces. In fact, Moulton calls his surfaces *field driven mean curvature* surfaces. Self-assembly has given us a new and interesting problem in mathematics. More importantly, note that this formulation has greatly simplified and clarified the parameter space of the original problem. Here, we find three nondimensional parameters, $\lambda$, $\sigma$, and $\delta$. The parameter space of the original formulation was six dimensional. With this formulation we see that not every parameter independently effects the shape of the surface. Rather, it is the ratio of groups of these parameters that is important. Next, we can immediately see that there are two special exact solutions to this problem. The first occurs when no voltage is applied; in this case the surface assumes the shape of a catenoid. The second is a cylindrical solution, $u = 1$, that occurs when exactly the right voltage is applied. This already begins to give some insight into the control that can be obtained over surface shape via the application of an electric field. Further,

using *perturbation theory*, these special solutions can be used to constr approximations to nearby solutions. In this way Moulton was able to answ a rather interesting question. In particular, he knew that in the absence applied voltage, there was a critical value of $\sigma$, such that the catenoid soluti disappeared when this critical value was passed. We saw this in Chapter Moulton asked whether or not an applied voltage would allow one to s assemble nearly catenoid shaped surfaces beyond the critical value of t parameter. His analysis yielded an affirmative answer to this question. Y to obtain such surfaces, parameter values must be balanced very carefully, is unlikely that this parameter regime would be found by experiment alo Further, through analysis of this model, Moulton was able to show that th are limits to what the applied field can do. He showed that as the voltag increased, the soap film begins to bulge outwards towards the outer electro But, it does not continue to do so in a smooth way until it reaches the ou electrode. Instead, there is a critical value of the applied voltage beyond wh the soap film simply pops. Finally, we should note that Moulton has carr out several experiments with this and related systems. He has obtained go agreement between his theory and experimental results.

This type of model is important in self-assembly precisely because it he guide the experimenter through a large and treacherous terrain. To be tr successful, models such as these must be tightly coupled to experimental forts. The development of virtually every self-assembling system discussed this book can benefit from models such as these. The models may not t the form of the model discussed here; energy minimization may not ap or such an approach may be too difficult to be of use. But, tight coupl between experimental efforts and theoretical efforts such as these promise help us push the boundaries of experimental self-assembly rapidly forward

### 9.2.2 Modelling Helix Formation

In the cell, DNA naturally assumes a helical shape. This basic design re curs throughout nature. In Chapter 3, we saw that the secondary struct of proteins consisted of $\alpha$ helices and $\beta$ sheets. The $\alpha$ helix is, obviou another example of helix formation in nature. The $\beta$ sheet is yet another ample, consisting of a sequence of helices lying side by side. Helix formati in proteins is one restricted example of the general protein folding probl discussed in Chapter 3. In this section, we consider a model due to Yeh Snir and Randall D. Kamien [126] that asks why the helix design is so pre lent in nature. Basically they asked: Why do natural structures so frequen self-assemble into a helix?

Their model, while still in our class of physical models, is more abstr than the model above. Rather than attempting to understand helix for tion in a particular protein, or other particular physical system, Snir Kamien attempted to find minimal conditions under which helix formati would occur. They began by considering a long cylindrical rod of radiu



immersed in a solution. They also imagined that this solution contained some concentration, $n$, of hard spheres of radius $r$. Next, they posited a simple interaction mechanism between their spheres and their rod. They imagined their rod was surrounded by an annular region that was inaccessible to the hard spheres. Next, they considered the entropy of the spheres. If the inaccessible region were very large, the spheres would be confined to some small region of space. In turn, this would imply that their entropy was low. If this region were smaller, the spheres entropy would increase. Now, fixing the excluded volume, they required that the system attempt to maximize the entropy of the hard spheres. With the excluded volume fixed, the only way entropy could increase was if the rod were to bend or fold. Folding resulted in increased entropy because it created regions where the inaccessible annular region overlapped itself. This reduced the inaccessible region seen by the spheres and hence increased their entropy. But, there is a cost to this folding, increased elastic energy. That is, it requires more elastic energy for the rod to bend than for it to stay straight. They required that their system minimize this elastic energy as well as maximize the entropy of the hard spheres. It was a balance of these two that they speculated might lead to helix formation.

This balance of entropy and energy can be expressed in the *free energy* for the system. For the Snir and Kamien system the total change in the free energy due to bending of the rod is given by

$$\triangle F = \frac{1}{2} L l_p \kappa^2 - n V_0. \tag{9.18}$$

Here, $L$ is the length of the rod, $\kappa$ is the curvature of the rod in a helical formation, $l_p$ is the persistence length and measures the stiffness of the rod, $n$ is the hard sphere concentration, and $V_0$ is the reduction in the excluded volume that occurs when the helical shape is assumed.

Notice that in their model, Snir and Kamien assumed a helical formation for the rod. A helix has constant nonzero curvature, hence the $\kappa$ term in Equation (9.18) is a constant. This model could be generalized by allowing the rod to assume any shape. In this case the $\kappa^2$ term would be replaced by an integral of $\kappa^2$ over the length of the rod. Since the shape of the rod can be specified completely in terms of its curvature, this would turn Equation (9.18) into a functional, mapping the rod shape to the free energy. Unfortunately, the second term in Equation (9.18) is not so easily expressed as an integral. This means that there is no easy approach to deriving a differential equation for the rod shape. So, instead, Snir and Kamien took their cue from nature, and replaced the unmanageable space of all possible rod shapes, with the manageable space of helices. Also note that Equation (9.18) embodies the competition between elastic energy and sphere entropy discussed above. The first term on the right is always positive and captures the increase in free energy needed to bend the rod. The second term on the right is always negative and captures the increase in entropy of the spheres that occurs when the overlap region becomes larger. This second term is also proportional to

the concentration of hard spheres. As this concentration increases, a sm change in the overlap volume results in a large change in the entropy of t particles.

The difficult aspect of analyzing this model is in computing the overlap v ume for a given rod configuration. Restricting attention to helices made t task manageable, but numerical computations were still necessary. Perfo ing these computations, the team then characterized their results in terms three dimensionless parameters. The first dimensionless parameter, $c = P/$ allowed them to specify their helix in terms of its pitch, $P$, and radius, $R$. T pitch of a helix is the distance between successive turns. The second dim sionless parameter, $r/t$, allowed them to characterize the relative size of hard spheres as compared to the radius of the rod. The final dimensionl parameter, $\theta = nr^3/(l_p/t)$, serves as a control parameter, allowing them compare a reference entropy with a reference energy.

Now, if the parameter $r/t$ is fixed, there is a single value of $c$ that maximi the overlap volume. Hence, $c$ may be regarded as a function of $r/t$. Num cally, the group computed this functional dependence and showed that as went to zero, $c$ tended towards a limiting value, $c^* \approx 2.5122$. This meant t for small spheres, a helix would form with this given pitch to radius ra Remarkably, this ratio compared well with measured ratios found in heli proteins, where the lower bound $c \approx 2$ had been found. The group also plot $\theta$ as a function of $r/t$. This gave them a picture of the configuration space the helix. For low values of $\theta$, the tube forms a stretched helix, but as so threshold value is crossed, the helix collapses into a tightly wound spiral.

To illustrate the role of lattice models and also to gain more insight i the work of Snir and Kamien, let's examine a simplified lattice model vers of their energy-entropy helix formation system. To begin, we'll constr our "rod" on a lattice like those discussed in Chapter 3. We can imag specifying the configuration of our rod by starting at the origin, pickin direction, moving one step in this direction, and then repeating the proc At each step, we form an edge, or a bond. A sample rod is shown in Fig 9.2. If we don't allow our rod to fold back onto itself, this means that in first step we have four directions to choose from, but in subsequent steps, o three. We can specify our rod in terms of a vector

$$\vec{m} = [m_0, m_1, \ldots, m_N]. \qquad (9.$$

The first component of the vector, $m_0$, takes the value 1, 2, 3, or 4, accord as our initial step. We label a step in the positive $x$-direction by 1, in positive $y$-direction by 2, and so on. The remaining components of the ve take on one of three possible values, $-1$, 0, or 1. We label a step with turn by 0, a counterclockwise step by $-1$, and a clockwise step by 1. N we imagine that every time we bend our rod, or in this case take a perpendicular to the last step, that it costs elastic energy. The total ela



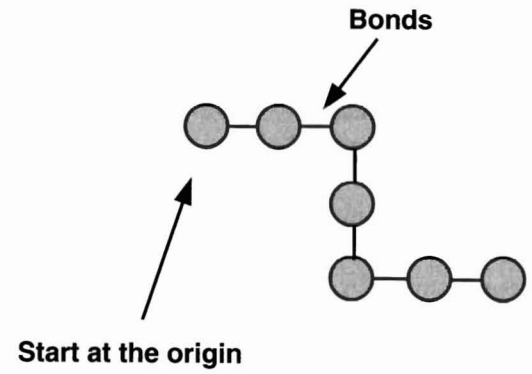


**FIGURE 9.2**: A bent rod in our lattice model.

energy of our rod can then be written as

$$\text{Elastic Energy} = \alpha \sum_{i=1}^{N} m_i^2 = \alpha m_b. \qquad (9.20)$$

Here, we have introduced the constant $\alpha$ to measure the magnitude of the energy required for one bend. The constant $m_b$ is simply the total number of bends in our rod. Note that the direction of the bend does not matter, this is why the $m_i$ are squared in the sum. Next, to capture the notion of excluded volume, we define $V_s$ to be the number of empty lattice sites adjacent to our straight rod, Figure 9.3. The reader may verify that for a rod with $N + 1$ bonds, $V_s = 2N + 6$. For a bent rod, we define $V_e$ to be the number of lattice sites adjacent to the rod. Hence, we can write an excluded volume energy as

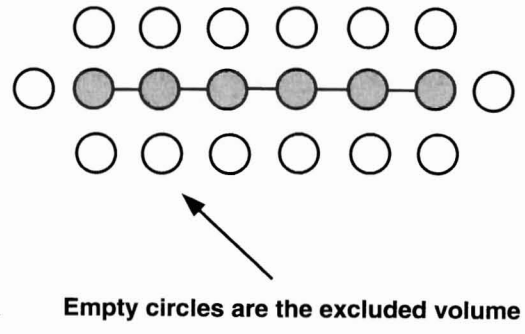


**FIGURE 9.3**: The definition of excluded volume in the lattice model.

$$\text{Excluded Volume Energy} = \beta(V_s - V_e) = V(\vec{m}). \quad (9.2$$

Here, the parameter $\beta$ may be thought of as a measure of the external c centration of hard spheres in our system. Our total energy is then

$$E[\vec{m}] = \alpha m_b - \beta V(\vec{m}). \quad (9.$$

Note that this too is an energy functional. This time, it maps our sh vector, $\vec{m}$, to the real number denoting the energy of the system.

Finding the shape assumed by our rod now reduces to computing $E[\vec{m}]$, all rods of a given length, and then picking the one with least total ene We illustrate with a simple example. If $N = 1$, we have rods consisting of o two steps. Ignoring symmetries, there are only two possible rod shapes, straight rod, or the right angled rod. These are shown in Figure 9.4. Th

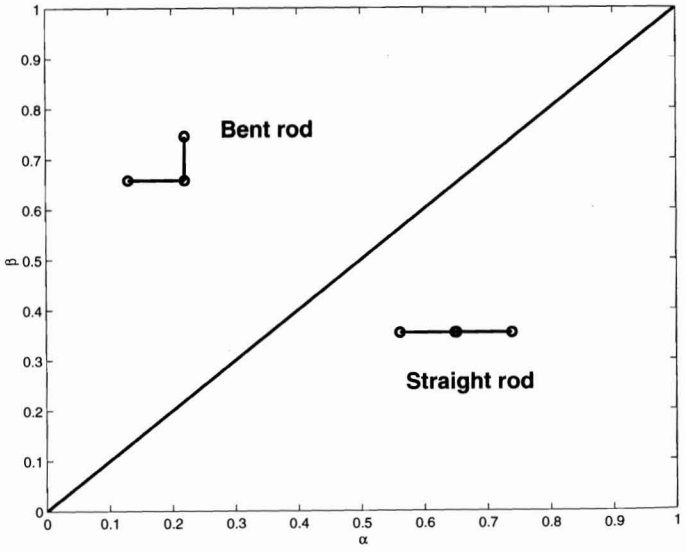


**FIGURE 9.4**: A phase diagram for our lattice model, $N = 1$.

are specified by the vectors $[1, 0]$ and $[1, 1]$. The energies are given by

$$E[[1, 0]] = 0 \quad (9.$$
$$E[[1, 1]] = \alpha - \beta. \quad (9.$$

Hence, we immediately see that if $\alpha > \beta$ the rod remains straight, whil $\alpha < \beta$, the rod bends. This lets us sketch the phase space shown in Fig 9.4.



This lattice model, while easily grasped, suffers from a similar defect to the model above. If $N$ becomes large, the number of possible rod shapes, or the size of the state space of the model, becomes enormous. This is a typical problem of models of self-assembly and one that is difficult to deal with.

The model of Snir and Kamien presents us with a nice example of a physical model that seeks to capture the behavior of a class of self-assembling systems. Their model provides good insight into minimal conditions for helix formation. The fact that their model predicts a pitch to radius ratio that is in good accord with experiment is remarkable. It illustrates the power of conceptually simple models to capture the actual behavior of real physical systems.

### 9.2.3 Chemical Kinetics Models

In Chapter 6, we examined the self-assembling tiles of Hosokawa et al. At the time, we noted that in their original paper, Hosokawa et al. presented a mathematical model of their system. In this section we return to the Hosokawa system and examine their modelling approach. We note that this model is a physical model, but relies on certain abstractions. Particle collisions are treated using chemical reaction kinetics; an assumption that may or may not be valid. Further, while Hosokawa et al. only applied their model to their self-assembling system, the basic approach is more widely applicable and has been used and extended by other authors.

Recall that Hosokawa et al. designed a set of tiles with the intent of self-assembling a simple finite cluster. In their first set of experiments, they encountered what they called *reverse coupling*. In their second set of experiments, Hosokawa et al. constructed tiles that enabled them to overcome the reverse coupling phenomenon of their first set. These tiles could exist in one of four stable cluster types. These clusters are pictured in Figure 9.5. Denoting these clusters by $X_i$, as shown in the figure, Hosokawa et al. proceeded by analogy with chemical kinetics and described cluster-cluster binding through a set of four reaction equations

$$\begin{aligned} 2X_1 &\longrightarrow X_2 \\ X_1 + X_2 &\longrightarrow X_3 \\ X_1 + X_3 &\longrightarrow X_4 \\ 2X_2 &\longrightarrow X_4. \end{aligned} \quad (9.25)$$

Next, they captured the state of their system at time $t$ in the state vector

$$\vec{x}(t) = [x_1(t), x_2(t), x_3(t), x_4(t)]^T. \quad (9.26)$$

Here, $T$ denotes the transpose so that $\vec{x}$ is actually a column vector, and the $x_i$ measure the amount of species $X_i$ at time t. Next, they formulated a discrete model for their system, based on reaction kinetics, and similar to the continuous reaction kinetic based models we have explored elsewhere in this

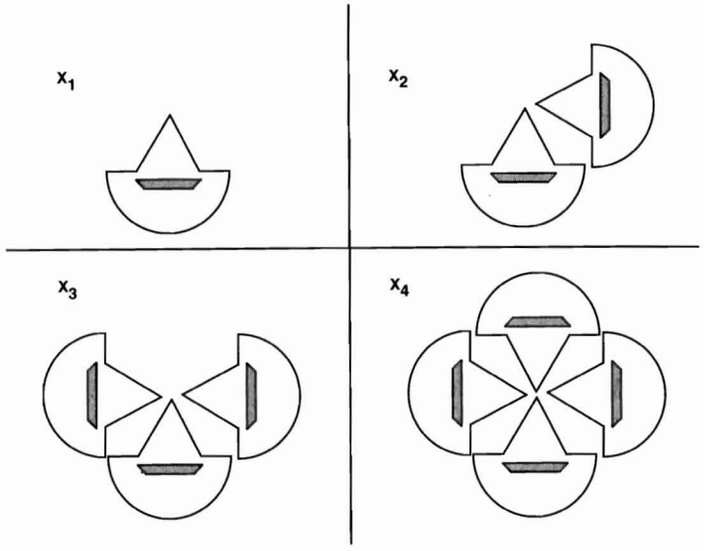


**FIGURE 9.5**: The four cluster types in the model of Hosokawa et al.

text. In particular, they assumed their system evolved according to

$$\vec{x}(t+1) = \vec{x}(t) + AP(\vec{x}(t)). \quad (9.2$$

To understand this discrete dynamical system, let us proceed in two ste For the moment, let's ignore $P$, and focus on the matrix $A$. The matrix $A$ given by

$$A = \begin{bmatrix} -2 & -1 & -1 & 0 \\ 1 & -1 & 0 & -2 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}. \quad (9.2$$

The components of this matrix come from the reaction equations. In partic lar, the $ij$th component, $a_{ij}$, is the number of $X_i$ in the $j$th reaction equatio So, $a_{11}$, is the number of $X_1$'s appearing in the reaction equation

$$2X_1 \longrightarrow X_2. \quad (9.2$$

The minus sign reflects the fact that $X_1$ is used up in this reaction, i.e it appears on the left hand side of the reaction equation. Now, we need return to $P(\vec{x}(t))$. Clearly, since it is to multiply the matrix $A$, $P$ must ta the vector $\vec{x}$ as input and return a column vector of the same length. That $P$ is itself a $4 \times 1$ vector. Now, the vector $P$ captures the rate at which o different reactions occur. Its $j$th component, $P_j$, represents the probabili



the $j$th reaction occurs in the given time step. If we proceeded directly from the Law of Mass Action, we'd find that

$$P(\vec{x}(t)) = k[x_1^2, 2x_1x_2, 2x_1x_3, x_2^2]^T \quad (9.30)$$

where $k$ is a rate constant for the reactions. But, this assumes that all reactions occur at the same rate. That was clearly not what Hosokawa et al. had observed in their experiments. Hence, they proposed a modified form for $P$, namely

$$P(\vec{x}(t)) = \frac{1}{S^2}[P_{11}^b x_1^2, 2P_{12}^b x_1x_2, 2P_{13}^b x_1x_3, P_{22}^b x_2^2]^T. \quad (9.31)$$

Here, $S$ is the total number of clusters of all types, and $P_{ij}^b$ is the conditional probability that a bond occurs between and $X_i$ and $X_j$ on the condition that they collide.

Hosokawa et al. offered a simple geometric model for approximating the $P_{ij}^b$. The reader may find this calculation in [62]. At this point, Hosokawa et al. turned to numerical simulation. Fortunately, their basic model is easy to implement computationally. They compared their experimental results with the output of their simulation. The agreement was not good. But, recall that in Chapter 6 when we discussed the Hosokawa system, we noted that the reaction

$$X_1 + X_3 \longrightarrow X_4 \quad (9.32)$$

hardly ever occurred. This observation did not accord with their calculated value of $P_{13}^b \approx 0.188$. So to more accurately reflect what was observed experimentally, they set $P_{13}^b$ to zero. This time, their model did give good agreement with their experimental results.

The model of Hosokawa et al. gives us another nice example of a theoretical model closely coupled to experimental efforts. The experiment and analogy with chemical kinetics suggested the form of the model. Experiment also helped identify the values of parameters in the model. This is a common and important form of feedback between theory and experiment. In a purely theoretical effort, the unusually low value of $P_{13}^b$ could not have been predicted. Yet, once the appropriate parameter values were uncovered, Hosokawa et al. could use their model with confidence to predict the behavior of systems larger than those experimentally accessible.

### 9.2.4 The Waterbug Model

As our final example of a physical model, we discuss the *Klavins' Waterbug Model* first presented by Eric Klavins in [71]. This model aims at both capturing the behavior of and develops methods for control of self-assembling systems. The model was motivated by the many different tile based systems we examined in Chapter 6, as well as the simple Cheerios effect phenomenon of Chapter 5. Klavins noted that this model, while formulated in terms of

the capillary bond, could easily be adapted to systems that use magne electrostatic, or other forces as their binding force.

As the basic subunit, or particle, in his model, Klavins used a *waterb* This is simple cross-shaped structure reminiscent of the Reif group's t from Chapter 8. The Klavins waterbug is shown in Figure 9.6. Note t

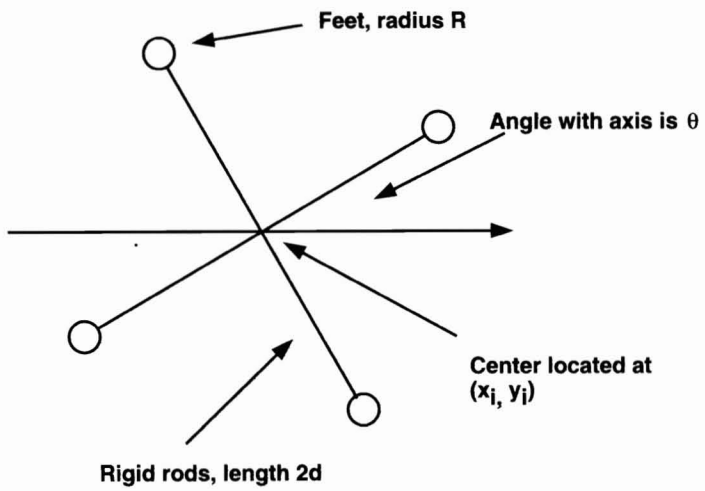


**FIGURE 9.6**: The geometry of the waterbug model.

this particle consists of two equal length rigid rods joined at their cent At the ends of each rod we find "feet." These are buoyant particles wh wettability can be controlled. As with the many tile systems of Chapte these waterbugs are constrained to live on a liquid surface. The meniscus ef then creates attraction or repulsion between the feet of various waterb Next, let's sketch the derivation of the waterbug model.

To capture the behavior of a system of $n$ waterbugs, we begin by specify the orientation and location of each particle. In particular, we define

$$q_i = [x_i, y_i, \theta_i]. \qquad (9.$$

The components $x_i$ and $y_i$ give the location of the center of the bug, component $\theta_i$ gives the angle of its rotation from the $x$-axis. These coordina are shown in Figure 9.6. From the $q_i$, the location of each foot of a given can be computed. We find

$$w_{i,j} = [u_{i,j}, v_{i,j}] \qquad (9.$$

where $w_{i,j}$ gives the position of the $i$th foot on the $j$th tile. Note that the and $v_{i,j}$ are specified in terms of the $q_i$. For example, $u_{i,1} = x_i + d\cos($



Hence, the complete state of the system is specified by the $q_i$. It is convenient to define

$$q = [q_1, q_2, \ldots, q_n] \qquad (9.35)$$

and use $q$ to denote the state of the system.

To model the dynamics of the system, again, an energy approach is useful. Here, the idea is to derive an expression for the potential energy, an expression for the kinetic energy, and then to use *Lagrangian dynamics* to derive equations of motion. In previous systems, we've minimized energy to derive governing differential equations. In Lagrangian dynamics, we minimize the *action* of the system. The action is defined as the difference between the total kinetic and total potential energy in the system. The reader will find more information on Lagrangian dynamics in the related reading section.

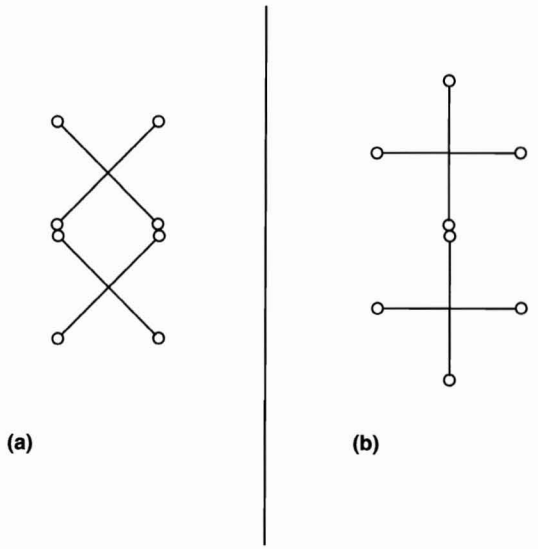


**FIGURE 9.7**: Stable and unstable configurations in the waterbug model. Type (a) is stable, type (b) is unstable.

The potential energy in this system is the sum of the pairwise potential of all possible pairs of interacting particles. The potential energy between particles $i$ and $k$ is given by

$$u(q_i, q_k) = -\sum_{j=1}^{4}\sum_{l=1}^{4} c_{ijkl} K_0(\rho||w_{i,j} - w_{k,l}||). \qquad (9.36)$$

This expression for the potential was derived by assuming that the feet inter-

acted like spheres on the surface of a fluid. In Chapter 5, we saw that t force between such particles could be obtained approximately in terms of t modified Bessel function $K_1$. The potential energy can be expressed in ter of the gradient of the force, and the derivative of the modified Bessel functi $K_0$ is the negative of $K_1$. This is the origin of the modified Bessel functio $K_0$, in Equation (9.36). Note that the argument of the Bessel function is t distance between a pair of feet multiplied by the parameter $\rho$. The para eter $\rho$ captures the difference in mass density between the particles and t fluid.[5] The term $c_{ijkl}$ originates in the coefficient of the force we examined Chapter 5. Here, this coefficient is

$$c_{ijkl} = 2\pi\gamma Q_{ij} Qkl. \tag{9.}$$

The terms $Q_{ij}$ and $Q_{kl}$ denote the wettability coefficient of the $j$th foot of t $i$th tile and the $l$th foot of the $k$th tile. Here, $\gamma$ denotes the surface tension the liquid on which the waterbugs float. It is also necessary to assume t $||w_{i,j} - w_{k,l}|| > 2R$, where $R$ is the radius of a foot. This assumption ensu that feet are not touching. The total potential energy of the system can n be expressed as

$$U(q) = \sum_{1 \le i \ne k \le n} u(q_i, q_k). \tag{9.}$$

We should note that there is an assumption here. We are assuming that t total potential energy can be obtained by simply summing over pairwise teractions of the particles. This requires that the disturbance of the menis between particles $i$ and $j$ due to other particles is negligible. This assumpt may or may not be valid in a given system, yet it is a common simplify assumption and serves to make the model tractable.

The kinetic energy is more easily obtained. The kinetic energy of a sin tile is

$$K_i = \frac{m}{2}(\dot{x}_i^2 + \dot{y}_i^2 + d^2\dot{\theta}_i^2) \tag{9.}$$

and hence the total kinetic energy is given by

$$K = \sum_{i=1}^{n} K_i. \tag{9.}$$

Here, $m$ is the mass of a waterbug and dots denote differentiation with resp to time.

Now, in the absence of friction, the equations of motion are obtained defining the Lagrangian as $\mathcal{L} = K - U$ and applying the appropriate Eu Lagrange equation. To include friction, the Lagrangian can be equated to sum of the frictional forces and again the appropriate Euler-Lagrange equat



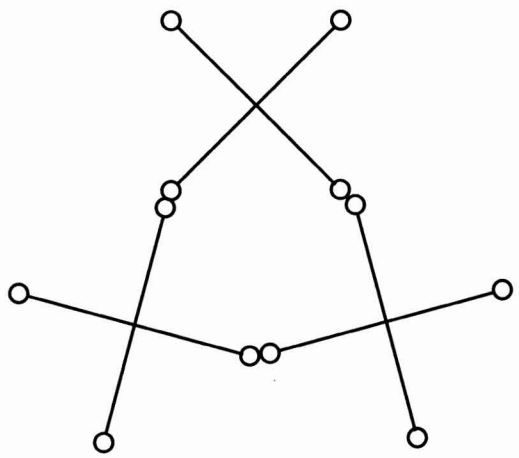

**FIGURE 9.8**: An example of a defective waterbug assembly.

applied. Here, this yields

$$8m\ddot{x}_i + \sum_{k=1,k\ne i}^{2} \frac{\partial}{\partial x_i} u(q_i, q_k) = -4k_f\dot{x}_i \tag{9.41}$$

$$8m\ddot{y}_i + \sum_{k=1,k\ne i}^{2} \frac{\partial}{\partial y_i} u(q_i, q_k) = -4k_f\dot{y}_i \tag{9.42}$$

$$8md^2\ddot{\theta} + \sum_{k=1,k\ne i}^{2} \frac{\partial}{\partial \theta_i} u(q_i, q_k) = -4k_f d^2\dot{x}_i. \tag{9.43}$$

Analytically this model is intractable. To study the behavior, Klavins turned towards numerical simulation. But, one additional modelling assumption was necessary. As formulated, the model does not prevent or account for the contact of feet. In order to simulate this model numerically, Klavins assumed that when two feet made contact, they became the same foot. That is, they overlapped. Further, the singularity in the potential energy that occurs when feet made contact was smoothed for numerical purposes. We note that numerical simulations of this system are not trivial. Computations involving only 40 particles took several hours. Nevertheless, in numerical simulations, the waterbug model did appear to capture the appropriate self-assembly dynamics.

Klavins was able to analytically investigate the stability of assemblies of waterbugs that formed during his simulations. In particular, he was able to show that the assembly of two tiles in Figure 9.7 (a) was stable, while the assembly in Figure 9.7 (b) was not. Similar results could be obtained for

systems of more than two tiles. But, the original intent of the model was
simply to investigate how waterbug tiles self-assembled, but to investig
how dynamic control over the wettability of the waterbug feet could transl
into control over the self-assembling system. To study this, Klavins po
two problems. The first problem was to use the control over wettability
eliminate defects in a given self-assembled structure. An example of suc
defect is shown in Figure 9.8. The second problem was to use the con
over wettability to build finite structures. This is reminiscent of the prob
posed by Hosokawa et al. who wanted to build stable structures consistin
only four particles. Klavins showed that standard ideas from *control the*
could be applied to this problem. In particular, he exhibited an open-loo
controller that specified the wettability of individual feet as time evolved
could be used to eliminate a certain class of defects. Similarly, he exhibite
control scheme that allowed the construction of certain finite structures.

The Klavins waterbug model illustrates how a theoretical framework
help aid in the development of future self-assembling systems. In develop
a relatively flexible model, Klavins was able to provide experimentalists
a platform to test and develop control schemes before costly experime
were conducted. It also illustrates some of the difficulties with theoreti
approaches to self-assembly. In particular, even relatively simple models
quickly become analytically intractable and computationally challenging.

## 9.3 Abstract Models

In this section, we present three models that are abstracted from the g
eral behavior of self-assembling systems. These models are representa
examples of this approach. We refer the reader to [2, 3, 102, 123, 132]
information on similar abstract approaches to modelling self-assembly.

### 9.3.1 Conformational Switching

In Chapter 3, we learned that proteins can assume different *conformatio*
For example, in examining the tobacco mosaic virus, we saw how interact
with RNA could induce washer shaped protein subunits to switch to a lo
washer conformation. In this new conformation, lock washers could bi
together and ultimately form the complete viral unit. This is a typical exam
of a *conformational switch* in self-assembly; in one conformation a particle
unable to bind, but once it switches to a new state, binding becomes possib

The first abstract model we'll consider is a model developed by Kazuhi
Saitou and Mark J. Jakiela [112, 113, 114] that attempts to clarify the *encod*



*power* of conformational switches. In particular, Saitou and Jakiela addressed the yield problem in self-assembly. They realized that a self-assembly process could be viewed as a sequence of small steps. This sequence, their *subassembly sequence*, dictated the final structures that were formed. Saitou and Jakiela wanted to know if particles capable of switching between different conformations could be used to encode for a particular subassembly sequence. In turn, if the subassembly sequence could be specified and unproductive sequences ruled out, they speculated that perhaps this would increase the yield of the final product.

First, to understand how a conformational switch might encode a subassembly sequence, let's consider a highly simplified situation. Imagine we have a system consisting of a mixture of three particle types. We'll label these $A$, $B$, and $C$. Let's picture our particles as squares and imagine that each particle has certain bond sites on its surface. We'll take our first set of particles to be as shown in Figure 9.9. Here, we'll use the same notation for bond sites


**FIGURE 9.9**: Particles and bonds for our first look at conformation switching.

that we did in discussing the DNA tiles of Chapter 8. In this system, that means that $A$ can bond to $B$ on the left and that $C$ can bond to $B$ on the right. Note that $B$ cannot bond to itself, it can only bond to a matching site on a complementary particle. The same is true for $A$ and $C$. Now, imagine that we have a large bag containing an equal mixture of these three tile types. At each time step, we'll reach into the bag, randomly grab two particles or clusters, pull them out, and if they can be bound, we'll bind them. We then return either the particles or the bound cluster to the bag and repeat the process. With the three particle types of Figure 9.9, it is clear that if we repeat this process long enough, we'll be left with a bag of $ABC$ clusters. But, in constructing these clusters, there are two possible subassembly sequences. We can envision these as reactions. In particular,

$$\begin{aligned} A + B &\longrightarrow AB \\ B + C &\longrightarrow BC \\ AB + C &\longrightarrow ABC \\ A + BC &\longrightarrow ABC. \end{aligned} \tag{9.44}$$

This means that we can get to the $ABC$ clusters by either first forming a $B$ and then adding an $A$, or by first forming an $AB$ and then adding a $C$. The is no preferred subassembly route.

To conveniently represent subassembly sequences, Saitou and Jakiela troduced the following notation. We write a complete subassembly seque using parenthesis, at each step in the process placing a set of parenth around a completed subassembly. So, the subassembly sequences above wo be denoted $(A(BC))$ and $((AB)C)$. Working our way inwards from the out most pair of parenthesis, we recover the subassembly sequence in reverse. $(A(BC))$ denotes the sequence where $BC$ formed first and an $A$ was add while $((AB)C)$ denotes the sequence where $AB$ formed first and a $C$ added. A second way of viewing possible assembly sequences of a compl collection of particles was also introduced. In this notation, we form a as shown in Figure 9.10. Here, we begin in the top bin with a set of o

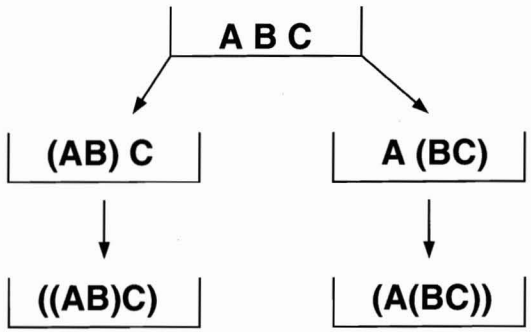


**FIGURE 9.10**: Tree notation for representing assembly sequences.

three particles, $A$, $B$, and $C$. We then follow the arrows downwards throu each possible assembly sequence. In either case we arrive at the end produ $ABC$.

Whatever our notation, it is clear that in the particle system of Figure all subassembly sequences are equally likely. We can create a preferred ro or preferred assembly sequence by introducing a conformational switch. particular, we introduce a switch in the $B$ particles. The type of switch w use was called a *minus device* by Saitou and Jakiela. A mechanical vers of the switch is pictured in Figure 9.11. The binding sites on the new particle are as before, but jutting out from the sides of the particle are pu rods that can act to restrict binding. If a $C$ particle attempts to bind with the push rod will prevent this binding. On the other hand, if an $A$ partic attempts to bind with $B$ on the left, it will be able to push the push r inwards and complete the bond. When this happens, the interior block slid



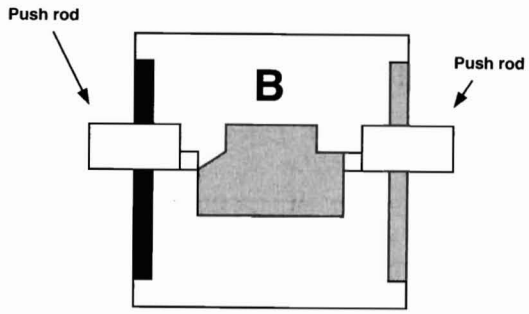


**FIGURE 9.11**: A mechanical conformational switch. The push rod on the left moves when an $A$ particle attempts to bind. This pushes the center block downwards, freeing the right side for binding with a $C$ particle.

downward, this frees the right hand push rod and makes the right hand bond site accessible. When an $A$ binds with $B$ and frees the right hand bond site, we say that $B$ has switched its conformation. We denote this new conformation by $B'$. In our new system, the set of possible reactions is restricted. In particular, we have

$$\begin{aligned} A + B &\longrightarrow AB' \\ AB' + C &\longrightarrow AB'C. \end{aligned} \tag{9.45}$$

This means that only one subassembly sequence is possible, the sequence denoted by $((AB)C)$. The introduction of a conformational switch has allowed us to *encode* for a particular subassembly sequence.

But, how does this affect the yield of the process? In our $ABC$ examples it is not easy to see this effect. In fact, there are two different ways that the presence of a conformational switch can effect the yield of a particular self-assembly process. To understand the first way, consider the two systems shown in Figure 9.12. In either case, imagine that our goal is to assemble an $ABC$ complex. In this case, that would result in a complete square. In Figure 9.12 (a), the bond sites for both the $B$ and $C$ particles are exposed. Hence, the following reactions can occur

$$\begin{aligned} A + B &\longrightarrow AB \\ AB + C &\longrightarrow ABC \\ A + C &\longrightarrow AC. \end{aligned} \tag{9.46}$$

Note that here, once an $AC$ complex has formed, there is no pathway leading to the desired end product. That is, there are two assembly sequences, $((AB)C)$ and $(AC)$. The first leads to the desired product, the second does not. In Figure 9.12, we have the same system, but the $A$ particle now has a

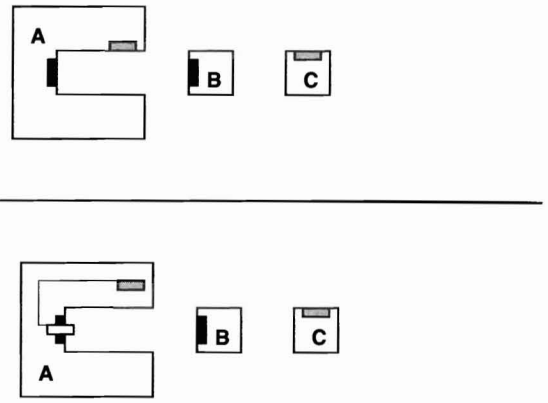


**FIGURE 9.12**: An example illustrating the effect of conformational swi ing on yield. In (a), the $C$ particle can bind first, blocking the $B$ particles (b), the bond site for $C$ is unavailable until the $AB$ bond has been made

conformational switch. The bond site for $C$ particles is not exposed unle $B$ particle binds first. The possible reactions for this system are

$$A + B \longrightarrow A'B \qquad (9.$$
$$A'B + C \longrightarrow A'BC$$

and the only assembly sequence is $((A'B)C)$. The presence of the confor tional switch prevents particles from getting trapped in the $AC$ configura and thereby increases the yield.

Saitou and Jakiela actually focused on the second way a conformatio switch can effect they yield of an assembly process. To understand this eff imagine we have a simple two particle system consisting of particle type and $B$. This time, we imagine that $A$'s can bond to $B$'s from the left hence the only reaction in the system is

$$A + B \longrightarrow AB. \qquad (9.$$

We'll also imagine that our target complex is the $AB$ complex. But, in process, imagine that we have a large concentration of $A$'s and a small conc tration of $B$'s. This implies that the number of favorable reactions occurr in the system will be very small. If we again imagine our particles be drawn from a large bag a pair at a time, most of the time we'll draw a p of $A$'s and return them to the bag with no bond being formed. This mea that when stopped at some finite time, the yield of our process will be ve low. Now, imagine we introduce a conformational switch. We'll let the



bond with one another pairwise in the reaction

$$A + A \longrightarrow AA'. \qquad (9.49)$$

The right hand $A$ particle has switched to the conformation $A'$. This new conformation allows the $A'$ to detach from the $A$ when a bond with a $B$ particle is formed. That is, the information about the bonding of $B$ is propagated through the $A'$ particle and delivered to the $A$ particle. Saitou and Jakiela called this type of conformational switch a *plus device*. Here, this is represented by the reaction

$$AA' + B \longrightarrow A + AB. \qquad (9.50)$$

We also imagine that once an $AB$ is formed no further bonding is possible. Now, if we start with a high concentration of $A$'s and a small concentration of $B$'s, the concentration of $A$'s will be rapidly reduced as $AA'$ complexes form. This has the effect of increasing the frequency of reactions involving $B$ and hence increasing the yield of the process.

Having seen the power of the conformational switch, let's consider a four particle system. We'll label our particles $A$, $B$, $C$, and $D$ and allow the following reactions to occur

$$\begin{aligned} A + B &\longrightarrow AB \\ B + C &\longrightarrow BC \\ C + D &\longrightarrow CD \\ AB + C &\longrightarrow ABC \\ ABC + D &\longrightarrow ABCD \\ B + CD &\longrightarrow BCD \\ A + BCD &\longrightarrow ABCD \\ AB + CD &\longrightarrow ABCD. \end{aligned} \qquad (9.51)$$

Our target structure will be $ABCD$; there are five possible assembly sequences. They are: $(((AB)C)D)$, $((AB)(CD))$, $(A((BC)D))$, $(A(B(CD)))$, and $((A(BC))D)$. In the first system we considered, the $ABC$ system, we could use the minus device to encode for any assembly sequence. However, in this four particle system, Saitou and Jakiela realized that the minus device was not powerful enough to encode for all of the different assembly sequences. In particular, they realized that $((A(BC))D)$ and $(A((BC)D))$ were *unencodable*, no matter how many conformational switches of the minus device type were used. Further, by simulating the set of reactions above numerically they observed that these unencodable sequences were among the highest yielding sequences in the system. In order to be able to encode these sequences using conformational switches, Saitou and Jakiela had to use both plus and minus devices. With this combination, any of the assembly sequences could be encoded.

But, now the general question remains. Suppose we have $n$ labelled pa
cles and we let them interact pairwise following the trend above. This quic
leads to enormous reaction sets and numerous assembly sequences. In exa
ining the cases $n = 3$ and $n = 4$, Saitou and Jakiela had showed that th
two conformational switch types were sufficient to encode for any assem
sequence. But what about the general case? Saitou and Jakiela focused
two major questions:

1. If we're given an assembly sequence can we tell whether or not it
be encoded using only minus devices? Or, some combination of mi
and plus devices?

2. If a sequence can be encoded, how many conformations are necessary
encode the given assembly sequence?

Note that this second question was motivated by the four particle syst
In order to encode the sequences $((A(BC))D)$ and $(A((BC)D))$ they had
use two types of switches. In addition, they had to use particles that carr
both plus and minus device switches. These particles had more than
different conformations. Hence, if a combination of plus and minus devi
was to be used, they had to allow for the possibility of particles with m
conformations.

To answer these questions, Saitou and Jakiela defined a *one-dimensio*
*self-assembling automaton* (SA). That is, they set up a formal structure, ty
cal of those used in theoretical computer science, such that within the cont
of this structure they could answer the questions posed above. In their mod
an (SA) was defined to be a pair $M = (\Sigma, R)$, where $\Sigma$ was a finite set of co
ponents and $R$ was a finite set of assembly rules. The assembly rules w
of two forms. The first, was an attaching rule that abstracted the notion
minus devices. In particular, attaching rules were rules of the form

$$a^{\alpha} + b^{\beta} \rightarrow a^{\gamma} b^{\delta}. \qquad (9.5$$

Here, the $a$ and $b$ are particle labels, and the exponents track the conformati
of the given particle. The second rule was a propagation rule abstracted fr
the notion of plus devices. These rules were of the form

$$a^{\alpha} b^{\beta} \rightarrow a^{\gamma} b^{\delta}. \qquad (9.5$$

Once this structure was in place, Saitou and Jakiela were able to classify S
based on the presence of different rule types. They were able to prove tha
both rule types were allowed any assembly sequence could be encoded usi
only three conformations per particle. The reader is referred to [112, 113, 1
for details of the proofs of these results.

The work of Saitou and Jakiela nicely illustrates the role of the abstra
model in self-assembly. Their model was not based on any particular physic
system, but rather on the basic notion of conformational switching seen



many self-assembling systems. With this type of model Saitou and Jakiela were able to ask and answer "What is possible?" type questions. Note that without this formal structure, little progress can be made with these questions. It is hard to imagine how within the context of one specific experimental system, questions such as these could be addressed. On the other hand, models such as this are always open to the criticism that they abstract too much. That too much of the real world is left out for the model to be able to be useful in designing any real world self-assembling system. The challenge is to bridge that gap, using the power of the abstract approach, but adding enough of the complexity of the real world to make such an approach useful in system design.

### 9.3.2 Graph Grammars

In Chapter 7, we discussed the dynamic self-assembling robotic tiles built by the Klavins' group. We noted that the tiles relied upon a *graph grammar* in order to make binding decisions. In this section, we'll examine this basic graph grammar approach as developed by Klavins and his collaborators [72, 73].

While the graph grammar approach incorporates the idea of conformational switching, it is aimed at different questions than the model of Saitou and Jakiela. To understand the questions addressed by the graph grammar approach, let's re-consider the simple self-assembling system we began with above. In particular, we'll again assume we have an equal mixture of three particle types, $A$, $B$, and $C$, and that they bind according to the reactions given in Equations (9.45). Now, above we assumed the complex $ABC$ was our target structure. But, this time, let's imagine that we have two target structures, $AB$ and $CC$. The first of these, $AB$, does occur during the assembly process. If we stopped the process at some finite time, we can expect to find some percentage of our particles in the $AB$ state. But, the longer the process continues, the less of these particles we will find. Eventually, given enough time, all $AB$ particles will bind with $C$'s and form $ABC$ complexes. Klavins et al. called structures like these *unstable*. This reflects the fact that these structures do not persist for all time. In contrast, structures like $ABC$ are called *stable*. Once they form, no further change is possible. Our second target structure, $CC$, is an example of what Klavins et al. called *unreachable*. Given the particles and the binding interactions, there is no assembly process that will lead to the $CC$ structure. Structures such as $ABC$, $AB$, or $BC$, are *reachable* structures within this system.

The graph grammar model is focused on these *stable* and *reachable* structures. Notice that the notion of a reachable structure is equivalent to the forward problem in self-assembly. The forward problem asks: Given a set of particles and a set of binding rules, what structures can form? In the language of the graph grammar model: Given a set of particles and a set of binding rules, what structures are *reachable*? A characterization of the set of reachable structures is one aspect of the graph grammar model. Klavins

et al. also focused on the backward problem in self-assembly. In the langua of the graph grammar model they asked: How can a set of binding rules constructed to ensure that a desired target is reachable? To this they add the notion of *stable* structures. This is an aspect of the yield problem in se assembly. By focusing on stable structures, Klavins et al. not only wanted ensure that a target was reachable, but that target structures would app once the assembly process was complete. In combining the questions ab reachable and stable structures, Klavins et al. actually asked the very spec question: Is it possible to design a set of binding rules such that a desi target structure is the only stable element in the reachable set for the syste

To address this question, Klavins et al. worked within the context of mathematical structure known as a *graph grammar*. A *graph* is simply collection of vertices and edges. We encountered graphs in Chapter 6 wh we studied the folding system of Griffith. If each vertex is given a name, have a *labelled graph*. Formally, we say that a labelled graph, $G$, over alphabet $\Sigma$, is a triple, $G = (V, E, l)$, where $V$ is a set of vertices, $E$ is a of edges, and $l$ is a labelling function. This function maps the vertices in to the alphabet, $\Sigma$. That is, the labelling function gives each vertex a na Throughout the remainder of this section, when we use the term *graph*, w mean a labelled graph.

Now, to build a graph grammar, we need to attach a rule set to our gra $G$. The rule set simply describes what we usually think of as binding rules conformational changes for a self-assembling system. As an easy example, develop a graph representation and a rule set for a simple system of partic We imagine that we have one type of particle that can be in any one of th conformations. This means we need three letters in our alphabet, we'll $a$, $b$, and $c$. To be concrete, let's assume that we start with eight parti all in conformation $a$. Each individual particle is identified by a vertex labelled by a letter from our alphabet. Our initial setup is pictured in Fig 9.13. Note that initially our graph has no edges, only labelled vertices. rule set captures how this graph evolves during an assembly process. As example, consider the rule set, denoted $\Phi$, defined by

$$\begin{array}{ll} a & a \longrightarrow b - b \\ a & b \longrightarrow b - c \\ b & b \longrightarrow c - c. \end{array} \qquad (9.$$

The first rule says that a pair of vertices, each with label $a$, may be repla by a pair of vertices, each with label $b$, and connected by an edge. The sec and third rules are similar. These are examples of *constructive rules*. the context of self-assembly, we are taking a pair of particles, creating a b between them, and in the case of the rules above, changing their conformati The action of this rule set on our initial graph is shown in Figure 9.13.

Notice that unlike the model of Saitou and Jakiela, here rules need be applied one at a time. Rather, an assembly sequence is valid if so



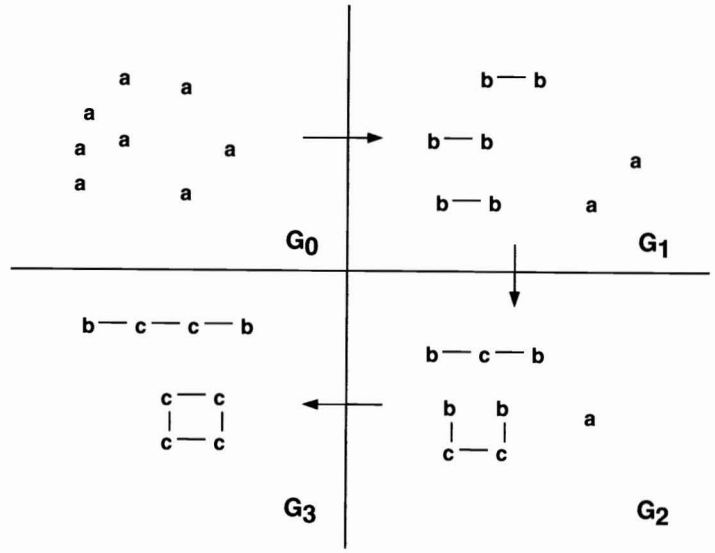


**FIGURE 9.13**: An example of an assembly sequence in the graph grammar model. Arrows indicate the direction of time.

application of the rule set to the graph at step $i$, $G_i$, produces the graph at step $i+1$, $G_{i+1}$. In Figure 9.13, we also see elements of the reachable set. Any structure that appears as we follow the arrows in the figure is, by definition, reachable. However, not all of these reachable structures are stable. We see that single particle pairings or chains of such particles are unstable. Only the closed loop of four $c$ type particles is a stable structure. If you examine this structure and the rule set $\Phi$, you'll see that no $c$'s appear on the left hand side of any rules. Hence, no changes to this structure are possible and it must be stable.

Now, rules need not be constructive and they need not be limited to acting on only two vertices. We can have *destructive rules*, for example

$$b - b \longrightarrow a \quad a \qquad (9.55)$$

or relabelling rules such as

$$b - b \longrightarrow a - c. \qquad (9.56)$$

We can also have rules such as

$$c - c - c \longrightarrow c - a - c \qquad (9.57)$$

that act on larger components of our graph. Putting the rules and graphs together, Klavins et al. defined their *graph assembly system* as the pair $(G_0, \Phi)$ consisting of an initial graph, $G_0$, and a rule set $\Phi$.

Drawing on classical topological techniques, Klavins et al. were able to cha
acterize the set of reachable structures and stable structures for their gra
assembly system. More importantly, they were able to exhibit an algorith
that allowed them to construct a rule set $\Phi$ such that a given graph $G$ cons
tuted the entire reachable set for their graph assembly system. This argume
also allowed them to find an algorithm that allowed them to construct a r
set $\Phi$ such that a given graph $G$ constituted the entire stable set for th
graph assembly system. Note that these algorithms are constructive.
their input they take a graph $G$ and as their output they return a rule set
and a labelling function, $l$. The reader is referred to [72] for proofs of th
results.

In some ways, the graph grammar approach of Klavins et al. is simply
generalization of the model of Saitou and Jakiela. However, it is an imp
tant generalization as it captures features of self-assembly not captured
the Saitou and Jakiela model. Additionally, it provides a constructive so
tion to an instance of the backward problem of self-assembly. Further,
graph grammar model has been implemented in the Klavins self-assembl
robotic system. It serves as a language that allows the robotic particles
make decisions about binding and unbinding. Unfortunately, the model
its limitations. In particular, it does not account for geometry. Particles
treated abstractly as vertices with labels and there is no way to capture
geometric configuration of a given set of particles. Also, the algorithms de
oped by Klavins et al. can lead to rule sets that are not physically realiza
There is no way to prevent this within the context of the model. As with
such abstract models, the next step is to bring the model closer to physi
reality.

### 9.3.3 The Tile Assembly Model

The *Tile Assembly Model* was introduced by Erik Winfree and developed
Winfree and his collaborators in [140, 141, 142, 107]. This important mo
links computation and self-assembly. As we saw in Chapter 8, by exploit
this link Winfree et al. were able to self-assemble a Sierpinski Gasket us
DNA tiles. This link is the most promising route to programmed self-assem
available today.

To understand this connection between computation and self-assembly,
need to introduce two concepts, the *Turing Machine* and *Wang Tiles*. T
notion of a Turing Machine allows us to think about computation abstract
i.e., divorced from any particular computer architecture or platform. T
notion of Wang Tiles connects computation to arrays of tiles. The Tile A
sembly Model marries the notion of Wang Tiles to the notion of self-assemb
thereby completing the connection to computation.

The Turing Machine was introduced by the mathematician Alan Turi
in 1936 as a way of defining computation and as a tool for exploring t
computability of objects. With this definition Turing formalized the questi



raised by David Hilbert in 1900, that is, whether it was possible or not to decide whether a given mathematical statement could be proved. In answering Hilbert's question, Turing also launched the field of theoretical computer science and provided its most powerful model.

A Turing Machine can be visualized as follows. Imagine we have an infinitely long tape that is subdivided into equal size squares. Each square can either be left blank, or can contain a zero or a one. Hovering over one square of the tape is a read-write head. This read-write head can erase a symbol, write a symbol, and advance the tape one square in either direction. The read-write head makes decisions about whether to read, write, or erase based on its internal state. The head has a finite number of states and a look-up table that dictates how it should behave once it reads the tape. That is, the head reads the value from the square below, checks its state, decides what its new state should be, what should be written in the square below, and how the tape should be advanced. The machine then executes this instruction and repeats. Usually we specify a special *halting state* where the Turing Machine stops all operation.

This definition of a Turing Machine can be made precise. It is typical to define a Turing Machine as a quintiple $(S, A_T, N, s_0, F)$, where $S$ is the finite set of available states, $A_T$ is the tape alphabet, $N$ is a transition function that tells the machine how to respond to a given input, $s_0$ is the initial state, and $F$ is the set of halting states.

Whether we proceed formally or not, the important fact to remember about Turing Machines is that they provide a model of computation. Furthermore, this model is *universal*. Turing was able to show that there existed a *Universal Turing Machine* that could simulate any other Turing machine. This implies that any computing process that can be simulated by a Turing Machine can in fact be simulated by one machine, the Universal Turing Machine. One caveat - there are other possible models of computation. The famous *Church-Turing Thesis* lets us sidestep this complication. This thesis states that all sufficiently complex models of computation are equivalent. Hence, to study computation we need only study Turing Machines, and to study Turing Machines we need only study the Universal Turing Machine.

Now, suppose we have a model that we think is capable of simulating computation. In order to prove this, we simply need to prove that it is equivalent to a Universal Turing Machine, or more simply, we need to show that our model is *Turing Universal*. One such model was proposed by Hao Wang in 1961 [136]. In [136], he introduced what have become known as *Wang Tiles*. Wang tiles can be visualized as a set of square tiles colored or shaded as in Figure 9.14. For a given set of tiles, Wang imagined that infinitely many copies of each were available. Using these tiles, he then asked if they could be made to tile the plane. But, in placing each tile, one had to observe the rule that colored sides of adjacent tiles had to match. For example, the first tile in the figure could be placed below a copy of the second tile, but nowhere else. Tiles were not allowed to be rotated.

## *Profile - Erik Winfree*

Erik Winfree is the architect of the *Tile Assembly Model.* With hi troduction of this model in 1998, Winfree showed that not only was assembly equivalent to computation, but that implementing computatio self-assembly was practical. His method for constructing computing assembling tiles opened the door to the world of programmable self-asse This promises to usher in a new era of self-assembled, atomically pr highly structured nanomaterials.

Fully embracing the interdisciplinary nature of self-assembly, Winfre not only made seminal contributions to theoretical self-assembly, but to e imental self-assembly as well. In conjunction with Nadrian C. Seeman, free demonstrated that DNA tiles could self-assemble into ordered per structures. A few years later, this time working with Paul W.K. Rothem Winfree fabricated a Sierpinski Gasket from DNA tiles, thereby demons ing that the full promise of the Tile Assembly Model could be realized e imentally. For this work, Rothemund and Winfree shared the 2006 Fore Institute's Feynman Prize for Nanotechnology – capturing the prize in the experimental and theoretical categories.

In a recent conversation, Winfree shared his thoughts on the blind s facing researchers in self-assembly today. He writes:

> *This is one blind spot: many researchers look at information and algorithms and can only see data processing programs running on conventional electronic computers – they don't see that information and algorithms are intrinsic to the behavior of molecular systems and that understanding this aspect of molecular behavior is key to fully exploiting what molecules can be designed to do. Or they do see that, but they don't yet see that it is no longer just the realm of science fiction.*

Winfree was trained as a mathematician and a computer scientist. received an undergraduate degree in mathematics from the Universit Chicago and a Ph.D. in Computation and Neural Systems from the Calif Institute of Technology. He is the recipient of numerous prizes and aw including a MacArthur Fellowship and a PECASE award from the Nati Science Foundation. In 1999, MIT's *Technology Review* named Winfree of their "Top 100 Young Innovators."

Presently, Winfree is an associate professor of computer science at the ifornia Institute of Technology. He currently leads several research effor the areas of DNA self-assembly, DNA computing, the study of gene regula networks, and DNA and RNA folding.



In his original paper, Wang conjectured that any set of tiles that could be placed so that they tiled the plane, did so periodically. In 1966, Robert Berger constructed a set of Wang tiles that tiled the plane, but aperiodically. At this time, it had already been shown that any Turing machine could be represented in terms of Wang tiles. By providing an example of an aperiodic Wang tiling, Berger showed that the tiling question of Wang was the same as the halting question in the theory of Turing Machines. This crucial result established that Wang tiles were in fact Turing Universal.

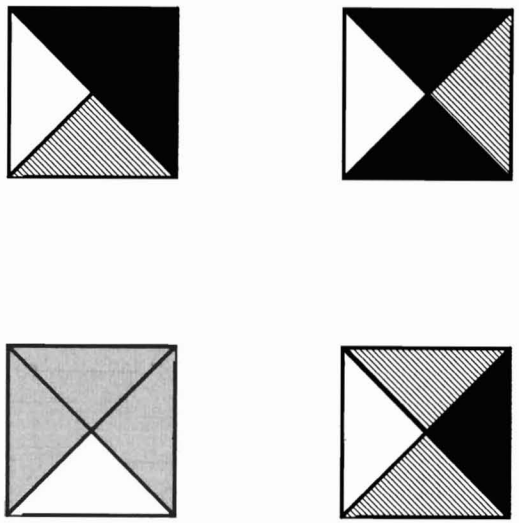

**FIGURE 9.14**: A set of four Wang tiles. Tiles must remain oriented as shown and can only be placed next to tiles if the edge colors match.

The crucial insight of Winfree was to realize that the colors on Wang tiles could be replaced by specific binding rules. But, binding rules meant self-assembly and that meant that tile based self-assembly could *compute.* In the other direction, this meant that tiles could be designed to produce programmed structures. Fully programmable self-assembly was possible.

Winfree did more than simply establish this result theoretically. With his Tile Assembly Model, Winfree showed how to construct tiles capable of computation via self-assembly that were physically realizable. The DNA tile systems of Chapter 8 are the most prominent examples of this construction. To specify the Tile Assembly Model, Winfree first needed to adapt Wang's notion of a tile. This was straightforward; a tile in the Tile Assembly Model *is* a Wang tile, but one described in the language of self-assembly. Instead

of thinking of a tile as having colors, Winfree thought of his tiles in terms
*binding domains*. A tile in the Tile Assembly Model is a unit square with
belled edges and just like Wang tiles, these tiles cannot be rotated. The edg
of these tiles possess bond sites on their four sides, and it is by matching bo
sites that tiles can assemble.

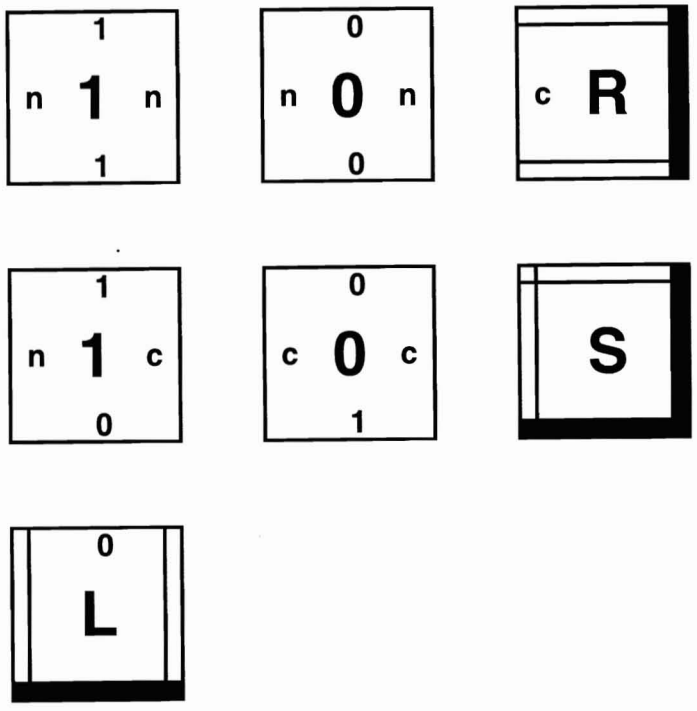


**FIGURE 9.15**: Tiles in the Tile Assembly Model. Small characters deno
binding sites, large characters specify tile type, and edge thickness indicat
bond strength.

A simple set of seven tiles is shown in Figure 9.15. Note that each tile
labelled in three ways, a large character in the center, small characters arou
the edges, and by edge colors. The small characters represent the differ
types of binding sites. In this set, $c$'s can bind to $c$'s, 1's can bind to 1's and
forth. The edge colors indicate the *strength* of these possible bonds. The da
edges denote binding sites that have zero strength. The single lined ed
denote binding sites that have strength one. The double lines indicate ed
with strength two. The large characters in the center of each tile are used
denote tile type. The tiles with the $L$, $R$, and $S$ are called *nucleating ti*



These tiles provide a nucleating center around which other tiles can grow and compute. The tiles with the 1 and the 0 are called *rule tiles*. These are the tiles that actually perform computation in the Tile Assembly Model.

Now, notice that in the Tile Assembly Model, tiles don't simply bond, they do so with a particular strength. By introducing this notion of bond strength, Winfree was also able to introduce the notion of *temperature* into the Tile Assembly Model. Suppose we denote the temperature of the system by $\tau$. We can now restrict possible binding interactions by only allowing a tile to bind to the assembly if the sum of the bond strengths it makes with that assembly exceeds or equals the temperature, $\tau$. For example, suppose we set $\tau = 1$ and we start with an $L$ tile. Then, any tile that can match bond sites with the $L$ tile can join the assembly. However, if we take $\tau = 2$, the growth is more restrictive. Initially the only tile that can join the $L$ tile is an $S$ tile; it's the only one capable of making a strength two bond. But, as the assembly grows we begin to encounter sites were a tile labelled 1 or 0 can make two bonds in joining the assembly. Since the sum of the bond strengths made by such tiles is now greater than or equal to the temperature, they are allowed to join. This type of bonding is called *cooperative bonding*. An example of a growing structure with temperature $\tau = 2$ is shown in Figure 9.16. This

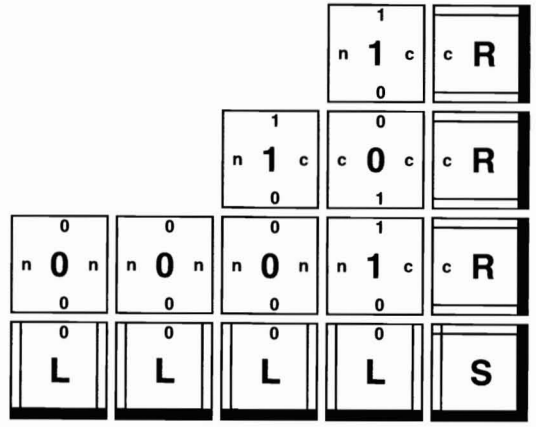


**FIGURE 9.16**: A sample configuration of tiles in the Tile Assembly Model. Here, the temperature, $\tau = 2$.

structure began with a single $L$ tile, added an $S$ tile to the right, added an $R$ tile above the $S$, and then finally, a 1 could bind in the corner.

To relate this growing structure to computation, we interpret the rows in the structure, read from the bottom up, as output bit strings. Computation is performed on these bit strings by the tiles labelled 1 and 0 in the same way

as computation was performed by the Rothemund system of Chapter 6. T
lower and right edges of the 1 and 0 tiles serve as inputs. In matching bo
sites on the growing structure, these edges are taking two tiles as input i
a computation. The upper and left edges of the 1 and 0 tiles serve as t
output. These output surfaces are exposed and available to serve as input
the next step of the computation.

Now, as with previous systems we've encountered, the Tile Assembly Mo
can be formally defined. In particular, a tile system, $T$, may be defined
a quadruple, $(T, S, g, \tau)$, where $T$ is the set of tiles, $S$ is a set of $\tau$-sta
*seed assemblies*, $g$ is the strength function, and $\tau$ is the temperature. He
the *strength function*, $g$, simply makes precise the notion of summing bo
strengths, and the definition of $\tau$-stable makes precise the notion of assemb
forming constrained by the temperature of the system. Within this mo
self-assembly is a means of going from one configuration of a tile system
another configuration. A tile may be added to a configuration if the resul
still $\tau$-stable; that is, tiles must satisfy the temperature rules outlined abo

Within the context of the Tile Assembly Model, Winfree and his collabo
tors have been able to answer a wide range of questions. Here, we'll descr
just one of these results. The reader is referred to [107, 140, 141, 142]
other related results.

In [107] Rothemund and Winfree sought to study the complexity of s
assembly. Recall that in Chapter 5, we introduced the notion of Kolmogo
complexity. We defined this measure of complexity for bit strings and loos
said that the complexity of a bit string was the length of the shortest compu
program needed to produce that string. As an example, we considered stri
of magnetic cubes created by self-assembly. If we started with cubes of t
colors, one containing two north faces, the other two south faces, and label
these by 1 and 0 respectively, we produced strings that looked like 0101010
That is, we produced an alternating colored string of cubes. If we started wi
eight cubes, four of each type, we would produce this result exactly. Since c
assemblies were not oriented we treated this string and the string 101010
as identical. In terms of the Kolmogorov program length definition, we co
compute the complexity of this string. But, we can also relate this noti
of complexity to our self-assembling system. The method of Winfree et al
simply to count the number of distinct particle types needed to uniquely s
assemble a target object. That is, the complexity of a self-assembled object
the number of distinct particle types needed to guarantee that this object
the unique result of the self-assembly process. In this example, the complexi
would be 2. We are able to assemble a string of alternating colored cubes usi
only two distinct particle types.

A less trivial question asks: What is the complexity of a self-assembled $N$
$N$ square?[6] The step from one dimension to two dimensions is large. Not
that for our alternating colored cube example, the complexity of assembly w
2, independent of the length of the target object. It takes two particle types
assemble an alternating colored chain of length 8 or of length 800. This se



assembling system is not very complex. But, to guarantee that we assemble a square *and nothing else* requires a more complex system. If you think back to the many tile assembly systems of Chapter 6, you'll realize that none of them are capable of this task. Winfree et al. proved that the complexity of an $N \times N$ square was $N^2$ for systems with temperature $\tau = 1$. From the discussion of temperature above, we see that in the $\tau = 1$ case, bonding is not cooperative. A tile either bonds using one edge, or it cannot bond at all. When $\tau = 2$, Winfree et al. proved that the complexity dramatically decreased to $O(\log(N))$.[7] In this case, the bonding is cooperative. Tiles can bond either by activating a bond site of strength two, or by activating two bonds sites of strength one. That is, two particles already in the assembly could cooperate to add this new particle.

The Tile Assembly Model is an important abstract model of self-assembly. While it is abstracted from a wide range of self-assembling systems and while there are nonphysical aspects to the model, it does closely capture the behavior of self-assembling tile systems. In particular, it captures the behavior of DNA tiles; in fact, the design of DNA tiles was at least partially inspired by the Tile Assembly Model. Even an abstract model can be closely coupled to experiment. Further, the model allows us to address the question of the complexity of a self-assembling system. In addressing this question, the Tile Assembly Model provides useful answers to "What is possible?" type questions. Finally, it is the model that makes the important link between computation and self-assembly; this is likely the route that future progress will follow to achieve programmed self-assembly.

## 9.4 Chapter Highlights

- Theoretical approaches to self-assembly are widely varied. They are best defined by the questions they seek to answer.
- *Physical models* attempt to capture the behavior of one specific experimental system or some small set of closely related systems. They are useful for clarifying complicated parameter spaces and guiding experimental efforts.
- *Abstract models* attempt to answer "What is possible?" type questions. They are usually abstracted from a wide class of experimental systems. They are useful for clarifying minimal conditions under which a particular behavior of self-assembling systems can be achieved.
- The model of field driven mean curvature surfaces, the model of entropy driven helix formation, the chemical kinetics based model, and the waterbug model, are all examples of *physical models*. They all operate at

different levels of abstraction, but are motivated by a small set of closel related experimental systems.

- The conformational switch model, the graph grammar model, and th Tile Assembly Model, are all examples of *abstract models*. They a motivated by a broad class of experimental systems or the general pro erties of self-assembly.

## 9.5 Exercises

### Section 9.2

1. The parameter $\lambda$ in Equations (9.15) and (9.16) contains the *square* the applied voltage. Why? What does this imply for the possible shap that can be obtained by applying an electric field?

2. Show that Equations (9.15) and (9.16) reduce to the catenoid proble of Chapter 6 when the applied voltage is removed.

3. Show that Equations (9.15) and (9.16) admit the special solution $u =$ Find the value of the parameters in the problem for which this solutio exists.

4. Consider a hard sphere and rod model where two identical rods are solution with a concentration of hard spheres. Suppose that the entrop and excluded volume ideas of this section hold, but that the rods a perfectly rigid and cannot bend. Instead, imagine that the rods a repulsive and that the energy needed to bring them together increase with the inverse of the square of the distance between their center Formulate a free energy and analyze this model. You may assume tha the rods remain parallel and that their tops and bottoms are aligned.

5. The helix formation model of this section strongly resembles the mag netic system of Chapter 5 where an external field was applied. Discus this relationship.

6. Return to the first two parts of this text and identify systems where a energy minimization principle came into play. Discuss how the mod elling techniques of this section could be applied to construct models o those systems.

7. For the lattice model of this section, compute the possible configuration and energies for $N = 2$. Sketch a phase diagram for this system.



8. For the lattice model of this section, find two different rods, of the same length, that have the same total energy. Your rods should be truly different. That is, they should not be equivalent under rotation or reflection. Can you choose $\alpha$ and $\beta$ such that your rods are the minimal energy configuration for your system?

9. The Hosokawa model is easily implemented on a computer. Construct a simulation using the parameter values $P^b_{11} = 0.438$, $P^b_{12} = 0.375$, $P^b_{22} = 0.25$, $P^b_{23} = 0$, $P^b_{13} = 0.188$, and $P^b_{33} = 0$. (Assume $P^b_{12} = P^b_{21}$ etc.) Plot $x_1$ versus $x_2$ for different instants in time. Do the same for $x_3$ versus $x_4$. Now, change $P^b_{13}$ to zero and repeat. How do your results change?

10. The waterbug model is difficult to simulate numerically. However, based on the material of Chapter 5 and the ideas of this section, you should be able to build a similar, more tractable model. In particular, consider two interacting spherical particles on the surface of a fluid. In Chapter 5, we studied the attractive force between these particles. Use this force and the ideas of this section to build a dynamical model governing the attraction between two spherical particles. See [133] for one such model.

### Section 9.3

11. Return to the self-assembling systems of the first two parts of this text and uncover all possible types of *conformational switching* that you can find. Compare and contrast these different types of switches. Which ones fit into the framework of the models of this section?

12. Numerically simulate the $ABC$ model with and without a conformational switch. Keep track of the assembly sequences followed by the system. Compare the behavior of the two.

13. Again return to the first two parts of this book and find examples of self-assembling systems that can be described by the conformational switching model and examples of systems that cannot be described by this model. For one of those that can, formulate the model. For those that cannot, clearly explain what features of the system the model cannot capture.

14. Return to the first two parts of this book yet again and find examples of self-assembling systems that can and cannot be described by the graph grammar model. For one that can, formulate the initial graph and rule set. For one that cannot explain what features of the system the model cannot capture.

15. The Tile Assembly Model is closely related to DNA tiling as discussed in Chapter 8. Return to Chapter 8 and compare DNA tiling with the Tile

Assembly Model. What features of the physical system are captured? What features are left out?

16. The set of tiles presented here for the Tile Assembly Model are designe to count in binary. Extend the structure of Figure 9.16 and show ho this binary counting takes place.

17. As noted in Chapter 5 and again here, the complexity of one-dimension self-assembling systems are generally low. Devise a system that assem bles a chain of length $3N$ such that the chain consists of repeatin subunits that alternate colors, say, red, white, and blue. What is th complexity of your system?

18. Throughout this section, we avoided discussing defects in self-assembly Each of the abstract models discussed incorporates or considers the po sibility of defective bonding. How might the introduction of defects int these models affect the answers to the questions each model sets out t address?

## 9.6 Related Reading

The book by Lin and Segel is a great introduction to both mathematic modelling and continuum mechanics.

C.C. Lin and L.A. Segel, *Mathematics Applied to Deterministic Pro lems in the Natural Sciences*, SIAM, 1988.

If you wish to learn more about electromagnetism, Jackson and Stratto are the place to start.

J.D. Jackson, *Classical Electrodynamics*, Second Edition, Wiley, 1975.

J.A. Stratton, *Electromagnetic Theory*, McGraw-Hill, 1941.

An introduction to mathematical modelling in the context of micro- an nanosystems may be found in:

J.A. Pelesko and D.H. Bernstein, *Modeling MEMS and NEMS*, Cha man and Hall/CRC, 2002.

A good place to learn more about Lagrangian dynamics is in the boo by Weinstock.

R. Weinstock, *Calculus of Variations*, Dover, 1974.



Abstract models of self-assembly often use the language and tools of computer science. You can learn more about these tool in the book by Sipser.

M. Sipser, *Introduction to the Theory of Computation*, PWS Publishing, 1997.

A gentler introduction to the ideas of theoretical computer science can be found in the books of Dewdney.

A.K. Dewdney, *The Tinkertoy Computer*, W.H. Freeman and Company, 1993.

A.K. Dewdney, *The New Turing Omnibus*, W.H. Freeman and Company, 1993.

## 9.7 Notes

1. The one exception to this is the first two abstract models, which are closely related and share notation.

2. If you've forgotten, the divergence theorem relates volume integrals to surface integrals. In this model, the integrals taken over the gap between the soap film and the electrode can be replaced by integrals over the soap film and electrode surfaces.

3. Note that this reference is to a Ph.D. thesis that was not yet complete when this book was written. I expect that it will be available by the time this book appears. If not, Derek Moulton will be happy to provide you with details of this work. He may be reached at moulton@math.udel.edu.

4. It is not yet clear whether or not this parameter regime is experimentally accessible *at all*.

5. In Chapter 5, $\rho$ was $q$. Here $q$ is used to denote the state of the system.

6. This problem was actually posed by Len Adleman. See profile in Chapter 2.

7. The "big O" notation is used in asymptotic analysis. The statement $O(\log(N))$ roughly means that as $N$ tends to infinity the complexity grows like a constant times $\log(N)$.