Моделирование развития цветка на однородной структуре

А. О. Черников

1. Введение

Существование однородных структур, моделирующих различные процессы, и даже построение универсальной однородной структуры исследуются в работе [5]. Однородная структура моделирует некоторую последовательность объектов $A_1, \ldots, A_n$, если указаны конфигурации $F_1, \ldots, F_n$, реализующие эти объекты соответственно, и в последовательные моменты времени $t_1 < t_2 < \ldots < t_n$ она принимает эти конфигурации. При этом в моменты $t_i < t' < t_{i+1}$ ее конфигурация может быть любой. В настоящей работе сделана попытка построить однородную структуру, которая моделирует последовательность стадий развития цветка, при этом ее конфигурация в любой момент времени реализует объект описанного класса «цветок» и она обладает памятью, то есть в некотором смысле обратима.

2. Основные понятия и результаты

В работе будет использоваться понятие однородной структуры. Содержательно, однородная структура представляет собой бесконечную схему, построенную из копий одного и того же конечного автомата и такую, что правила соединения входов произвольного автомата с выходами других автоматов в ней везде одинаковы. В дальнейшем эти автоматы будем называть ячейками ОС. Дадим формальное определение описанного выше объекта. Однородной структурой называется четверка $(\mathbb{Z}^k, Q, V, \varphi)$, где $\mathbb{Z}^k$ есть множество k-мерных векторов с целыми координатами, Q — конечное множество состояний,

$V = \{\alpha_1, \ldots, \alpha_{n-1}\}$ — упорядоченный набор различных ненулевых векторов из $\mathbb{Z}^k$, называемый шаблоном соседства ОС и определяющий для каждой ячейки α ее окрестность $V(\alpha) = \{\alpha, \alpha + \alpha_1, \ldots, \alpha + \alpha_{n-1}\}$, $\varphi: \ Q_n \to Q$ — локальная функция переходов. Если в момент времени t состояния ячеек $\alpha, \alpha + \alpha_1, \ldots, \alpha + \alpha_{n-1}$ были равны соответственно $x_0, x_1, \ldots, x_{n-1}$, то состояние α в момент $t+1$ полагается равным $\varphi(x_0, x_1, \ldots, x_{n-1})$. Состоянием ОС будем называть функцию, сопоставляющую каждой ячейке ее состояние из множества Q. Если состояние ОС в момент времени t есть f, то ее состояние в момент $t+1$ есть функция g, определяемая равенством $g(\alpha) = \varphi(f(\alpha), f(\alpha + \alpha_1), \ldots, f(\alpha + \alpha_{n-1}))$. Это равенство задает основную функцию переходов $\Phi: \ g = \Phi(f)$. Поведением ОС назовем последовательность $\{f_i\}$ ее состояний такую, что $f_{i+1} = \Phi(f_i)$.

Существует один выделенный элемент из множества Q — символ пустоты. Будем обозначать его через Π. Будем рассматривать далее лишь такие ОС, у которых $\varphi(\Pi, \ldots, \Pi) = \Pi$. Состояние Π является состоянием покоя. Состояние ОС, у которого лишь конечное число ячеек находится в отличном от Π состоянии, назовем конфигурацией. Также будем говорить, что ячейка принадлежит конфигурации, если ее состояние отлично от Π.

Мы будем рассматривать однородную структуру на плоскости, то есть $k = 2$, а ячейки называть клетками.

Определение. Назовем клетку A ***соседней*** к клетке B, если B находится непосредственно справа, сверху, слева или снизу от A. Очевидно, что понятие соседства — симметричное отношение, поэтому чаще просто будем называть такие клетки A и B ***соседними***. Пусть F — произвольная конфигурация, $A \in F$, $B \in F$. ***Путем*** от A до B будем называть последовательность $\{A_i\}$, $0 \leqslant i \leqslant n$ клеток конфигурации F $A = A_0, A_1, \ldots, A_n = B$, таких что A_i, A_{i+1} — соседние для $\forall i: \ 0 \leqslant i \leqslant n-1$.

Определение. Конфигурация F ***связная***, если для любых клеток $A \in F$, $B \in F$ существует путь между ними. Связную конфигурацию будем называть ***фигурой***.

Будем говорить, что фигура разбита на (под)фигуры (состоит из (под)фигур), если указан набор непересекающихся (под)фигур, таких что их объединение содержит все клетки исходной фигуры.

Определение. Будем говорить, что фигуры F_1 и F_2 ***касаются***, если они не пересекаются, и существуют соседние клетки $A \in F_1$ и $B \in F_2$.

Определение. Фигуру F назовем ***«горизонтально выпуклой»***, если ее пересечение с горизонтальной прямой — связная фигура, то есть отрезок этой прямой.

Замечание 1. Из этого определения видно, что если фигура «горизонтально выпуклая», то она состоит из конечного числа горизонтальных отрезков ненулевых клеток (фигуру, состоящую из этого отрезка, мы будем называть ***уровнем***), причем на произвольной горизонтальной прямой лежит не более одного уровня. Клетку с наименьшей абсциссой будем называть **левой** клеткой уровня, а с наибольшей — **правой**. Назовем уровень ***верхним*** или ***нижним***, если ордината этого уровня наибольшая или наименьшая соответственно. Если при этом фигура связная, она представляет собой последовательность уровней $L_1, L_2, \ldots, L_n$, где уровни L_i и L_{i+1} касаются для $\forall i: \ 1 \leqslant i \leqslant n-1$, а уровни L_1 и L_n являются нижним и верхним соответственно. Будем называть n **высотой** фигуры.

Введем следующие классы фигур:

Определение. Фигура F называется ***«цветоложе»***, если F является связной, «горизонтально выпуклой», а также удовлетворяет следующим требованиям:

1) F содержит клетку M и четыре клетки ML, MR, MT, MB слева, справа, сверху и снизу от M соответственно.
2) Клетка MT лежит на верхнем уровне F.

Замечание 2. «Цветоложе» может состоять только из пяти M-клеток.

Замечание 3. Верхний уровень «цветоложе» (и только он) может содержать специальные клетки O_n, необходимые для определения «цветка», причем не более одной для каждого n. При этом клетки O_n и O_m не могут быть соседними, $m \neq n$. Эти клетки будут служить основаниями (корнями) «отростков».

Замечание 4. Клетка O_n может совпасть с клеткой MT.

Определение. Фигура F называется «отросток_n», где n — идентификатор, если F является связной, «горизонтально выпуклой» и касается клетки O_n, находящейся на верхнем уровне «цветоложе». «Отросток$_n$» не касается «отростка$_m$», $m \neq n$. Наличие клетки O_n при этом обязательно. «Отросток» может представлять собой, например, лепесток, тычинку, пестик.

Определение. Фигура F называется ***«цветок»***, если она состоит из «цветоложе» и конечного набора «отростков», при этом не более одного для каждого n.

Так, например, если у «цветка» два лепестка, две тычинки и пестик, то он имеет 5 отростков. Каждому «отростку» должен быть приписан идентификатор от 1 до 5, на верхнем уровне «цветоложе» должны быть выделены клетки $O\_1, O\_2, \ldots, O\_5$, которые и будут основаниями (точками касания) соответствующих «отросток_1»,…, «отросток_5».

Определение. Зададим однородную структуру σ и функцию f из множества состояний в множество $\{$«цветоложе», O_n, «отросток$_n$», $\varnothing\}$, причем $f(\Pi) = \varnothing$. Будем говорить, что конфигурация σ реализует «цветок» F, если множество всех ненулевых клеток конфигурации совпадает с множеством клеток F и f переводит клетки фигуры F в ту (под)фигуру, которой она принадлежит.

Определение. Назовем клетку ***граничной***, если одна из ее соседних клеток находится в состоянии П. Множество граничных клеток «цветка» назовем его ***границей***.

Определение. Два «цветка» F_1 и F_2 назовем ***близкими***, если $F_1 \subseteq F_2$ или $F_2 \subseteq F_1$, а $F_1 \cap F_2$ состоит только из клеток границы.

Определение. Назовем последовательность фигур $F_1, F_2, \ldots, F_n$ ***«последовательностью роста цветка»*** (далее просто ПРЦ), если для $\forall i:\ 1 \leqslant i \leqslant n-1$ F_i и F_{i+1} — «цветки» и являются близкими.

Определение. ОС моделирует ПРЦ $F_0, F_1, \ldots, F_n$, если в моменты времени $t_0 < t_1 < \ldots < t_n$ ее конфигурации реализуют «цветки» $F_0, F_1, \ldots, F_n$ соответственно.

Пусть дан набор фигур $A_1, \ldots, A_k$, представляющих собой стадии развития цветка. Будем рассматривать только такие наборы, в которых каждая из фигур A_i, $1 \leqslant i \leqslant k$ удовлетворяет требованиям определения «цветка», то есть в ней можно выделить M-клетки, «цветоложе» и набор «отростков». При этом, если какой-нибудь отросток моделируемого цветка, например «пестик», присутствует на нескольких фигурах, то соответствующий ему «отросток_n» фигур имеет один и тот же идентификатор n. Понятно, что при таком условии, для $\forall n$ и $\forall m$, $m \neq n$ из того, что клетка «O_n» левее клетки «O_m» на одной из фигур A_i, следует, что она находится левее на всех фигурах, на которых присутствуют данные клетки. Будем называть такие «цветки» «цветками» **одного типа**.

Введем набор преобразований:

(1) Добавить новый нижний уровень к «цветоложе».

(2) Добавить клетку слева или справа к одному из уровней «цветоложе». Уровень не должен быть верхним.

(3) Удалить левую или правую клетку одного из уровней «цветоложе». Преобразование неприменимо, если:
– уровень является верхним;
– уровень состоит из одной клетки;
– удаляемая клетка является одной из «M-клеток» (см. определение «цветоложе»);
– если удалить клетку, то нарушится связность «цветоложе»;

(4) Удалить нижний уровень «цветоложе». Уровень должен быть ниже, чем уровень, содержащий MB.

(5) Добавить клетку слева или справа к верхнему уровню «цветоложе».

(6) Удалить левую или правую клетку верхнего уровня «цветоложе». Клетка не должна быть MT или O_n.

(7) Выделить клетку O_n (то есть пометить некоторую клетку верхнего уровня «цветоложе») на верхнем уровне «цветоложе», при ее отсутствии.

(8) Добавить к «отростку_n» новый верхний уровень. Если «отросток_n» отсутствует, то уровень добавляется над клеткой O_n. При этом новый уровень не должен касаться соседних «отростков», и клетка O_n обязательно должна быть выделена. В противном случае, преобразование неприменимо.

(9) Добавить по одной клетке справа или слева к одному из уровней «отростка_n». Новая клетка не должна касаться соседних «отростков».

(10) Удалить левую и правую клетки на одном из уровней «отростка_n», если уровень состоит из более чем одной клетки, и удаление клетки не нарушает связность.

(11) Удалить верхний уровень «отростка_n».

Замечание 5. Для применимости преобразований (9)–(11) необходимо наличие «отростка_n».

(12) Сдвинуть клетку O_n на одну позицию влево или вправо. При этом клетка, на которую она сдвигается, должна быть обычной клеткой «цветоложе», и после сдвига клетка O_n должна будет касаться «отростка_n» и не может оказаться рядом с клеткой O_m, $m \neq n$.

(13) Переименовать клетку O_n в обычную клетку «цветоложе» (при наличии «отростка_n»).

Из определения предыдущих преобразований видно, что они сохраняют связность, «горизонтальную выпуклость» каждой из (под)фигур, условия касаний «отростков_n» и клеток O_n, а также изменяют состояния только граничных клеток. При этом, очевидно, тип «цветка» не меняется. Поэтому справедливо следующее утверждение:

Утверждение 1. *Класс «цветок» замкнут относительно любого из этих преобразований. При этом, если «цветок» F_1 после преобразований перешел в «цветок» F_2, то «цветки» F_1 и F_2 одного типа и этот набор преобразований составляет ПРЦ, переводящую F_1 в F_2. Далее мы будем рассматривать только такие ПРЦ.*

Теорема 1. *Для произвольного набора «цветков» одного типа $A_1, \ldots, A_k$ существует ПРЦ $F_{n1}, \ldots, F_{n2-1}$, $F_{n2}, \ldots, F_{n3-1}$, $F_{n2+1}, \ldots, F_{nk}$, такая что $F_{n1} = A_1, \ldots, F_{nk} = A_k$ для некоторых $n_1 < n_2 < \ldots < n_k$.*

Теорема 2. *Для произвольной ПРЦ существует ОС, которая ее реализует.*

Следующая теорема является следствием этих двух теорем.

Теорема 3. *Пусть дан набор «цветков» одного типа $A_1, \dots, A_k$. Существует ОС, моделирующая развитие цветка с этими стадиями развития, то есть будучи приведенной в конфигурацию, реализующую A_1, она в некоторые моменты времени $t_1 < t_2 < \ldots < t_k$ находится в конфигурациях, реализующих $A_1, \dots, A_k$ соответственно, а далее останавливается (ее конфигурация остается неизменной с течением времени). При этом она обладает следующими свойствами:*

– *В процессе ее функционирования, все конфигурации реализуют некоторый «цветок» (а не только в определенные моменты времени).*

– *Данная структура обладает «памятью»: до тех пор, пока она не остановится, она является обратимой. Это значит, что посмотрев на конфигурацию ОС в момент времени $t < t_k$, мы сможем восстановить конфигурацию ОС в любой момент времени t', $t_1 \leqslant t' < t$.*

3. Вспомогательные утвержения

Лемма 1. *Пусть A и B — два «цветка» одного типа. Существуют «цветок» C и ПРЦ $A = F_1, F_2, \dots, F_n = C$. Причем «цветок» C удовлетворяет требованиям:*

a) $C \subseteq A$;

b) если клетка «O_n» присутствует в C для некоторого n, то она присутствует и в B;

c) если «отросток_n» присутствует в C для некоторого n, то он присутствует и в B;

d) высота «цветоложе» C не больше высоты «цветоложе» B;

e) высота «отростка_n» C не больше высоты соответствующего «отростка» B при всех n, для которых этот «отросток» присутствует в C (и, следовательно, в B).

Доказательство. Для удаления «отростка» «цветка» A, который отсутствует в «цветке» B, используем нужное число раз преобразование (11) для этого «отростка», а затем преобразование (13).

Если высота «цветоложе» A больше высоты «цветоложе» B, используем нужное число раз преобразование (4).

Для каждого из «отростков», высота которых у «цветка» A больше соответствующей высоты соответствующего «отростка» у «цветка» B, используем нужное число раз преобразование (11).

Очевидно, что последовательность этих преобразований образует искомую ПРЦ. Заметим, также, что порядок выполнения преобразований здесь произвольный. Лемма доказана.

Лемма 2. *Пусть у нас есть «цветки» C и B, удовлетворяющие условиям b)–e) леммы 1. Тогда существуют «цветок» D и ПРЦ $C = F_1, F_2, \dots F_n = D$, причем «цветок» D удовлетворяет требованиям:*

a) D имеет высоту «цветоложе» такую же, как и «цветок» C;

b) D имеет те же «отростки», что и C;

c) высота каждого «отростка» D совпадает с высотой соответствующего «отростка» C;

d) $D \subseteq B$;

e) если какой-либо уровень «цветоложе» присутствует в D, то этот уровень совпадает с соответствующим уровнем B;

f) если какой-либо уровень некоторого «отростка» присутствует в D, то этот уровень совпадает с соответствующим уровнем B.

Доказательство. Как и в лемме 1 будем строить последовательность преобразований. Сначала перечислим все необходимые преобразования.

Так как М-клетки неподвижны, то для построения «цветка» D необходимо:

– Изменить начало и конец каждого уровня «цветоложе» «цветка» C так, чтобы они совпали с соответствующими началом и концом «цветка» B. Это можно сделать набором преобразований (2) и (3).

– Изменить начало и конец верхнего уровня, используя преобразования (5) и (6).

– Изменить начало и конец каждого уровня каждого «отростка» «цветка» C так, чтобы они совпали с соответствующими началом и концом соответствующего «отростка» «цветка» B. Это можно сделать набором преобразований (9) и (10).

– Сдвинуть все клетки O в нужном направлении на верхнем уровне «цветоложе», используя преобразование (12).

Итак, у нас есть набор преобразований, необходимых для построения «цветка» D. Покажем, что существует последовательность, составленная из всех преобразований этого набора. Для этого необходимо, чтобы после применения некоторого количества преобразований не сложилась ситуация, в которой все оставшиеся преобразования неприменимы к «цветку», полученному в результате сделанных преобразований, который мы далее будем называть **текущим** «цветком».

Для «цветоложе» все просто. Если в наборе осталось хотя бы одно добавление клетки (преобразование (2)), то его всегда можно применить. Если же их не осталось, то в силу связности «цветка» B, очевидно, что все оставшиеся преобразования по удалению клеток применимы, то есть не нарушают связности.

Покажем, что и для «отростков» всегда найдется применимое преобразование. Для этого рассмотрим самый левый отросток. Очевидно, что если в наборе осталось преобразование «добавить к этому „отростку“ клетку слева», то оно применимо. Если его не осталось и есть преобразование «удалить у этого „отростка“ клетку справа», то оно применимо. В самом деле, если оно неприменимо, то удаление клетки ведет к нарушению связности, но, значит, эта клетка лежит на единственном пути от нижнего уровня к верхнему. Следовательно, либо над этой клеткой находится начало вышележащего уровня, либо под ней — начало нижележащего уровня. Так как мы эту клетку должны удалить, и при этом «цветок» B связный, то обязательно должно еще быть преобразование «добавить клетку слева» для вышележащего или нижележащего уровня (если уровень является нижним, то «добавить клетку слева» заменяется применимым в этом случае преобразованием «сдвинуть клетку O влево»), что противоречит предположению об отсутствии таких преобразований в наборе. Но если нет преобразований по удалению клеток справа, значит конец каждого уровня текущего «цветка» совпадает с концом соответствующего уровня «цветка» B. Поэтому аналогичное рассуждение применимо ко второму слева «отростку». Таким образом, по индукции, мы получим, что для любого «отростка» нет преобразований «добавить клетку слева» и «удалить клетку справа». Аналогично, для самого правого «отростка» всегда применимо преобразование «добавить клетку справа», поэтому их не должно быть. Значит нет и «удалить клетку слева». И далее, проводя аналогичные рассуждения, получим, что в наборе нет и преобразований «удалить клетку слева» и «добавить клетку справа» ни для одного из «отростков». Значит в наборе осдатись только сдвиги клеток O. Часть этих сигналов используются в процессе сдвигов и удалений «отростков» (если это необходимо для сохранения касания). Если они остались, очевидно, что их можно применить. Так как применены все необходимые преобразования, ПРЦ является искомой. Лемма доказана.

Лемма 3. *Пусть есть «цветки» D и B, удовлетворящие условиям d)–f) леммы 2. Тогда существует ПРЦ $D = F_1, F_2, \ldots, F_n = B$.*

Доказательство. Если в B есть «отросток», которого нет в D, то, используя преобразования (7) и (8) для этого «отростка», создаем его.

Для «цветоложе» нужное число раз применяем преобразование (1), а для «отростков» преобразование (8). Так как «цветок» B удовлетворяет всем условиям «некасания» «отростков», то, очевидно, что все преобразования применимы в любом порядке и искомая ПРЦ найдена. Лемма доказана.

4. Доказательство теорем

Доказательство теоремы 1.

Последовательно применяя леммы 1, 2 и 3 строим ПРЦ для каждой пары A_i, A_{i+1}, $1 \leqslant i \leqslant k-1$. При этом для того, чтобы ПРЦ моделировала развитие цветка более естественным образом, будем выбирать, при использовании каждой леммы, случайным образом преобразование из всех применимых на текущий момент, так как во всех леммах порядок выполнения применимых преобразований не важен. Объединение этих построенных ПРЦ и есть искомая ПРЦ.

Доказательство теоремы 2.

Без ограничения общности, будем считать, что состояние клетки описывается 5 флагами (так как число значений каждого из флагов конечно, то существует биекция между E_N и нашим набором флагов): TYPE, SIGNAL, SIDE, X, TIMER. Если какое-то значение флага относится к «отростку», то оно будет снабжено параметром: например, PETAL_TYPE(n) означает, что для каждого n свое значение типа, причем все они различны.

Флаг TYPE принимает значения INITIAL (пустая клетка, П), BUD (обычная клетка «цветоложе»), TOP (клетка верхнего уровня «цветоложе»), PETAL_ROOT(n) (клетка O_n), PETAL_TYPE(n) (клетка «отростка_n») и, очевидно, описывает функцию f для структуры, то есть указывает ту подфигуру, которой принадлежит эта клетка.

Флаг SIGNAL является идентификатором сигнала. Флаг X — параметром сигнала. Все преобразования, за исключением (1), (4), (8) и (11), реализуются одним сигналом. Все сигналы образуются в клетке M, а затем «проводятся» к месту применения.

Опишем как соотносятся сигналы и преобразования в момент образования в клетке M (если параметр X не указан, то это означает, что он не используется, то есть равен 0).

Преобразование (1):

Осуществляется последовательностью сигналов:
SIGNAL=GROW_BUD_DOWN, X $= k$ — добавить клетку, находящуюся под самой левой клеткой нижнего уровня «цветоложе». k — номер текущего нижнего уровня. $k > 0$ из-за присутствия клетки MB ($k = 0$ означало бы, что нужно добавить уровень с клеткой MB, а он есть по определению «цветка»).
Далее применяется нужное число раз преобразование (2), а затем преобразование (3), которые, очевидно, в этой ситуации всегда применимы.

Преобразование (2):

SIGNAL=GROW_BUD_LEFT, X $= k$ — добавить клетку слева к уровню с номером k. Уровень, содержащий клетку M, имеет номер 0, уровень под ним 1 и т. д.

SIGNAL=GROW_BUD_RIGHT, X $= k$ — добавить клетку справа к уровню с номером k.

Преобразование (3):

SIGNAL=DELETE_BUD_LEFT, X $= k$ — удалить левую клетку уровня с номером k.

SIGNAL=DELETE_BUD_RIGHT, X $= k$ — удалить правую клетку уровня с номером k.

Преобразование (4):

Осуществляется последовательностью сигналов:

Преобразованиями (2) и (3) приводим нижний уровень к тому, что он состоит только из одной клетки, расположенной под левой клеткой предыдущего уровня.

SIGNAL=DELETE_BUD_LEVEL, X $= k$ — удалить нижний уровень, состоящий из этой одной клетки. k — номер нижнего уровня.

Преобразование (5):

SIGNAL=GROW_TOP_LEFT — добавить клетку слева к верхнему уровню «цветоложе».

SIGNAL=GROW_TOP_RIGHT — добавить клетку справа к верхнему уровню «цветоложе».

Преобразование (6):

SIGNAL=DELETE_TOP_LEFT — удалить левую клетку верхнего уровня «цветоложе».

SIGNAL=DELETE_TOP_RIGHT — удалить правую клетку верхнего уровня «цветоложе».

Преобразование (7):

SIGNAL=SET_NEW_PETAL(n), X $= k$ — задать клетку O_n. k показывает смещение абсциссы клетки O_n относительно клетки M: отрицательное — влево, положительное — вправо. Если $k = 0$, то над клеткой M.

Преобразование (8):

Осуществляется последовательностью сигналов:

SIGNAL=GROW_PETAL_UP(n), $X = k$ — добавить клетку над самой левой клеткой «отростка_n». k — номер верхнего уровня этого «отростка». Если $k = 0$, то клетка добавляется непосредственно над клеткой O_n. При этом, этот «отросток», состоящий пока из одной клетки, может касаться какого-либо соседнего. Далее применяем преобразования (9) и (10) нужное число раз, и в результате касания нет. Важно, что даже если «отростки» касаются, сигналы преобразований (9) и (10) отличны для разных «отростков» и не влияют на «чужие» «отростки».

Преобразование (9):

SIGNAL=GROW_PETAL_LEFT(n), $X = k$ — добавить клетку слева к «отростку_n» к уровню с номером k. Уровень с O_n имеет номер 0, уровень над ним — 1 и т. д. При этом, если $k = 0$, то сигнал реализует *преобразование (12)* и сдвигает клетку O_n влево.

SIGNAL=GROW_PETAL_RIGHT(n), $X = k$ — добавить клетку справа к «отростку_n» к уровню с номером k. При этом, если $k = 0$, то сигнал реализует *преобразование (12)* и сдвигает клетку O_n вправо.

Преобразование (10):

SIGNAL=DELETE_PETAL_LEFT(n), $X = k$ — удалить левую клетку «отростка_n» у уровня с номером k, $k > 0$.

SIGNAL=DELETE_PETAL_RIGHT(n), $X = k$ — удалить правую клетку «отростка_n» у уровня с номером k, $k > 0$.

Преобразование (11):

Осуществляется последовательностью сигналов:

Преобразованиями (9) и (10) приводим верхний уровень соответствующего «отростка» к тому, что он состоит из одной клетки, расположенной над левой клеткой предыдущего уровня. При реализации, как и в преобразовании (8) может возникнуть касание этого «отростка» с соседним, и в этом случае это касание тоже не мешает.

SIGNAL=DELETE_PETAL_LEVEL(n), $X = k$ — удалить верхний уровень «отростка_n», состоящий из этой одной клетки. Здесь k — номер верхнего уровня.

Преобразование (12):
Реализуется сигналами преобразования (9).

Преобразование (13):
SIGNAL=DELETE_PETAL(n) — превратить клетку O_n в клетку «цветоложе».

Введем дополнительные сигналы, посылаемые клеткой M:
CALM — сигнал спокойствия. Необходим, чтобы посланные сигналы не пересекались, а также является сигналом клетки, которая ничего в данный момент времени не передает.
NEXT_TIMER — сигнал для инкрементации TIMER у клетки ML.
STOP — сигнал остановки структуры.

И введем вспомогательные сигналы, которые используются при реализации: GROW_BUD_DOWN_DIRECT и GROW_PETAL_UP_DIRECT(n).

Флаг SIDE является вспомогательным для проведения сигнала и может принимать 4 значения:
CENTER — либо флаг SIDE не используется, либо сигнал нужно распространять в две стороны уровня в зависимости от сигнала.
LEFT — сигнал надо перемещать влево.
RIGHT — сигнал надо перемещать вправо.
DOWN — сигнал надо перемещать вниз, если это сигнал для «цветоложе» и вверх, если это сигнал для «отростка».

Флаг TIMER не равен 0 только для трех клеток:

– у клетки M он равен -1, если структура работает, и -2, если закончила работу.
– У клеток MR и ML TIMER > 0 и указывает, какой сигнал будет следующим в клетке M.

Клетки в состоянии П имеют следующие значения флагов: TYPE=INITIAL, SIGNAL=CALM, SIDE=CENTER, X=0, TIMER=0.

Функционирование структуры происходит следующим образом: в зависимости от значений TIMER у MR и ML (эти пары в разные моменты времени различны), заполняются значения флагов

SIGNAL, SIDE и X у клетки M (TYPE=BUD, TIMER=-1). Далее этот сигнал реализуется на структуре, и конфигурация, реализующая F_i переходит в конфигурацию, реализующую F_{i+1}. После образования в клетке M сигнала, отличного от CALM, для того, чтобы не произошло наложение сигналов, необходимо несколько раз пустить сигнал CALM (хотя многие сигналы не требуют этой задержки). Клетка MR имеет TYPE=BUD, SIDE=CENTER, X=0, SIGNAL=CALM или SIGNAL=GROW_BUD_RIGHT, если добавляется клетка справа к уровню 0. Другие сигналы через нее не проходят. При этом на каждом шаге ее значение TIMER увеличивается на 1, а если достигает некоторого, фиксированного при построении структуры, числа NT, то становится равным 1. Аналогично клетка ML имеет TYPE=BUD, SIDE=CENTER, X=0, SIGNAL=CALM или SIGNAL=GROW_BUD_LEFT. Но ее TIMER увеличивается только при значении у клетки M SIGNAL=NEXT_TIMER. Этот сигнал заносится в M, если у MR TIMER=$NT-1$. Таким образом, пара ML.TIMER и MR.TIMER пробегает последовательно натуральные числа, записанные в NT-ичной системе счисления. Клетки MB и MT ничем не отличаются от обыкновенных клеток и необходимы лишь для проведения сигналов.

Есть особенный сигнал STOP. При его появлении в клетке M, в следующий момент времени устанавливается ее TIMER=-2, а TIMER клеток MR и ML обнуляется и, таким образом, структура останавливается.

Опишем теперь более подробно реализацию каждого из сигналов.

Сначала опишем сигналы изменения «цветоложе».

Если клетка находится под клеткой M (у которой TIMER=-1) и в ней сигнал sign по изменению «цветоложе», то ее SIGNAL становится равным sign, X на 1 меньше, а SIDE=DOWN, если под ней клетка «цветоложе» (TYPE=BUD), и CENTER в противном случае.

Если над клеткой находится клетка «цветоложе» с сигналом sign, то она принимает этот сигнал, на 1 уменьшает параметр X и в зависимости от наличия под ней клетки «цветоложе» получает направление CENTER или DOWN. Если SIDE=CENTER, то под клеткой нет клетки «цветоложе», а значит ее нужно искать. В следующий момент, клетка слева от клетки, принявшей сигнал сверху, принимает сигнал sign не меняя параметра X и устанавливает SIDE=DOWN, если под ней клетка «цветоложе», и SIDE=LEFT в противном случае. Вообще сигнал переходит влево только в том случае, если SIDE клетки-источника сигнала равен CENTER или LEFT. Клетка справа действует аналогично, только в случае отсутствия клетки «цветоложе» под ней ее SIDE=RIGHT. Так как фигура горизонтально выпуклая, то только один из этих двух сигналов («левый» и «правый») пройдет вниз, а флаг SIDE не позволяет сигналу, который ищет переход вниз слева идти вправо и наоборот. Таким образом осуществляется проведение сигнала вниз до нужного уровня. Сигнал передается вниз, пока $X > 0$.

Как только $X = 0$, сигнал не передается вниз (точнее не принимается снизу), а проводится в нужную сторону: влево для сигналов GROW_BUD_LEFT, DELETE_BUD_LEFT, GROW_BUD_LEVEL и DELETE_BUD_LEVEL и вправо для GROW_BUD_RIGHT и DELETE_BUD_RIGHT.

Обработка сигнала GROW_BUD_DOWN и $X = 0$: как только его получает левая клетка уровня (причем его получает именно клетка нижнего уровня в силу выбора k), она превращает его в сигнал GROW_BUD_DOWN_DIRECT. Если над клеткой в состоянии П этот сигнал, то она становится клеткой BUD.

Обработка сигнала DELETE_BUD_LEVEL и $X = 0$: как только его получает какая-либо клетка, в следующий момент времени она принимает состояние П.

Обработка сигнала GROW_BUD_LEFT и $X = 0$: как только он появляется справа от клетки в состоянии П, то она становится клеткой BUD. Аналогично, для GROW_BUD_RIGHT.

Обработка сигнала DELETE_BUD_LEFT и $X = 0$: как только слева от клетки находится клетка в состоянии П, она тоже переходит в состояние П. Аналогично, для DELETE_BUD_RIGHT.

Сигналы GROW_TOP_LEFT и GROW_TOP_RIGHT сразу принимаются клеткой над M. А затем проводятся клетками TOP и PETAL_ROOT(n) до крайних клеток. Как только справа от клетки в П появляется сигнал GROW_TOP_LEFT, она становится клеткой TOP. Аналогично, для GROW_TOP_RIGHT. Здесь и далее, под фразой «сигнал sign проводится клетками TYPE=...» будет означать, что если слева от клетки указанного типа находится клетка с сигналом sign и SIDE CENTER или LEFT, то она получит SIGNAL=sign и SIDE=LEFT с тем же параметром X. Аналогично для проведения сигнала вправо, только там SIDE=RIGHT.

Сигналы для «отростков» (они уникальны для каждого n) также проводятся клетками TOP и PETAL_ROOT(n), но если надо проводить сигнал, который находится в клетке PETAL_ROOT(n) с тем же n, что и сигнал, то он не проводится. Над клеткой PETAL_ROOT(n), как и под M, всегда находится клетка PETAL_TYPE(n) (в силу определения «отростка»), за исключением случая, когда клеток этого типа вообще нет. Поэтому они обрабатываются аналогично сигналам для «цветоложе». Разница в том, что сигнал проводится вверх, а не вниз, и GROW_BUD_DOWN_DIRECT заменяется на GROW_PETAL_UP_DIRECT(n), причем необходимо проверять, что сигнал принадлежит именно этому «отростку» (это обеспечивает корректную работу ОС даже если «отростки» касаются; сигнал будет работать только на «своем» «отростке»).

Отдельно оговорим случай сигнала GROW_PETAL_LEFT(n) с параметром $X = 0$, то есть преобразование (12) (случай сигнала GROW_PETAL_RIGHT(n) аналогичен). В этом случае, если сигнал находится в клетке PETAL_ROOT(n), то в следующий момент времени делаем ее клеткой TOP, а если клетка справа от клетки TOP обладает этим свойством (причем по условию применимости преобразования (12), это должна быть именно клетка TOP), то делаем ее клеткой PETAL_ROOT(n).

Остался сигнал DELETE_PETAL(n). Он также проводится как и сигналы для «отростка_n», но как только попадает в клетку PETAL_ROOT(n), в следующий момент времени она становится клеткой TOP.

В случае, если клетка не может принять никакого сигнала и не содержит сама сигнала, изменяющего ее состояние, ее SIGNAL=CALM, а все остальные флаги не изменяются.

Построенная ОС удовлетворяет условиям теоремы 2. Теорема доказана.

Утверждение 2. *Построенная ОС обладает «памятью» в указанном в п. 1 смысле, и в любой момент времени ее конфигурация реализует некоторый «цветок».*

Доказательство. То, что любая из ее промежуточных конфигураций является «цветком», очевидно.

Покажем, что она обратима в любой момент времени до остановки.

По флагам TIMER клеток MR и ML мы можем точно определить текущий момент времени (так как это его запись в NT-ичной системе). Какие-то посланные сигналы еще не реализованы ОС, так как каждый сигнал должен пройти путь до места применения. Заменим функцию перехода для клетки M. Будем считать, что все время подается сигнал CALM. Достаточно подождать некоторое время (например, достаточно подождать число тактов, равное числу непустых клеток), и все посланные сигналы будут реализованы. Теперь достаточно просто посмотреть на последовательность сигналов, посланных с начала работы и отменить их в обратном порядке, и мы получим конфигурацию ОС в начальный момент времени, а следовательно, и во все промежуточные. В силу специального подбора сигналов (именно поэтому пришлось разбить преобразования (1), (4), (8) и (11) на несколько сигналов) видно, что любой из них легко обращается, кроме STOP, но сигнала STOP нет по условию утверждения (берется момент времени до остановки ОС). Утверждение доказано.

Замечание 6. Построенная согласно теореме 1 ПРЦ моделирует развитие цветка достаточно искусственным образом. Для того, чтобы придать развитию более естественный вид, последовательность сигналов ОС строится следующим образом.

Для каждой пары (A_i, A_{i+1}) последовательных стадий развития цветка сначала строим ПРЦ. Далее эта ПРЦ преобразуется в набор сигналов для ОС без учета сигналов CALM и NEXT_TIMER как было описано выше. Теперь выбираем случайным образом сигнал, который применим к «цветку» A_i из этого набора, и изменяем в соответствии с ним «цветок». Иногда при этом необходимо добавить несколько сигналов в набор. Опишем эти случаи.

Заметим, что для «цветоложе» и любого из «отростков» результаты лемм 1 и 3 не применяются одновременно. Следующие рассуждения верны и для «цветоложе» и для «отростков», поэтому при указании сигналов не будем указывать, к какой части «цветка» они применяются.

Если применяются результаты лемм 1 и 2, то необходимо обработать три случая:

- Выбран сигнал DELETE_LEVEL. Тогда из набора убираем все сигналы, относящиеся к удаляемому уровню (при этом, очевидно, в наборе уже нет сигналов, относящихся к уровням с большим индексом).
- Выбран сигнал DELETE_LEFT на таком уровне, что в наборе еще есть сигнал DELETE_LEVEL для следующего уровня. Для корректного удаления этого уровня необходимо сдвинуть его вправо. Для этого добавляем для него по одному сигналу GROW_RIGHT и DELETE_LEFT.
- Выбран сигнал GROW_LEFT на таком уровне, что в наборе еще есть сигнал DELETE_LEVEL для следующего уровня. Для корректного удаления этого уровня необходимо сдвинуть его влево. Для этого добавляем для него по одному сигналу GROW_LEFT и DELETE_RIGHT.

Если применяются результаты лемм 2 и 3, то необходимо обработать случай сигнала GROW_BUD_DOWN и GROW_PETAL_UP.

Они «выращивают» клетку напосредственно под (над) самой левой клеткой предыдущего уровня после всех применений сигналов к этому предыдущему уровню (лемма 3 применяется после всех преобразований леммы 2). Поэтому нужно вычислить, насколько нужно сдвинуть новый уровень. Каждый непримененный к текущему моменту времени сигнал GROW_LEFT предыдущего уровня приводит к сдвигу на 1 влево, а сигнал DELETE_LEFT — на 1 вправо. Для сдвига на n влево добавляем n раз пару сигналов GROW_LEFT и DELETE_RIGHT, а для сдвига вправо — пару GROW_RIGHT, DELETE_LEFT.

При этом, из получившегося в результате всех этих добавлений сигналов набора удаляем все пары сигналов (DELETE_RIGHT, GROW_RIGHT) и (DELETE_LEFT, GROW_LEFT), так как они являются обратными.

Повторяем описанную процедуру выбора случайного сигнала до тех пор, пока не будут применены все сигналы. Аналогично доказательству леммы 2 легко показать, что все сигналы будут выбраны. В полученную последовательность сигналов вставляем сигналы CALM и NEXT_TIMER.

Объединяя эти последовательности для всех пар стадий развития «цветка», получим искомую последовательность сигналов.

5. Оценка числа состояний построенной ОС

Введем следующие параметры:
N_{pet} — число различных «отростков» на всех «цветках».
H_{pet} — максимальная высота «отростка» на всех «цветках».
H_{bud} — максимальная высота «цветоложе» на всех «цветках».
W_{bud} — максимальная высота верхнего уровня «цветоложе» на всех «цветках».
W_{fl} — разность между максимальной абсциссой клетки в состоянии отличном от П на всех «цветках» и минимальной абсциссой, «максимальная ширина цветка».
H_{fl} — «максимальная высота цветка».
N_{fl} — число фигур, представляющих стадии развития цветка.

Клетка BUD может проводить сигналы GROW_BUD_LEFT, GROW_BUD_RIGHT, DELETE_BUD_LEFT, DELETE_BUD_RIGHT, GROW_BUD_DOWN, DELETE_BUD_LEVEL с SIDE, равным CENTER, LEFT, RIGHT и DOWN, и параметром $0 \leqslant X \leqslant H_{bud}$. Всего $24 \cdot H_{bud}$ состояний.

Сигналы GROW_BUD_DOWN_DIRECT, CALM дают еще 2 состояния.

Всего: $24 \cdot H_{bud} + 2$ состояния. У них у всех TIMER=0, в отличие от клетки M.

Клетки TOP и PETAL_ROOT(n) проводят сигналы GROW_PETAL_LEFT(n), GROW_PETAL_RIGHT(n), DELETE_PETAL_LEFT(n), DELETE_PETAL_RIGHT(n), GROW_PETAL_UP(n), DELETE_PETAL_LEVEL(n) с SIDE, равным CENTER, LEFT, RIGHT и $0 \leqslant X \leqslant H_{pet}$. Всего $18 \cdot N_{pet} \cdot H_{pet} \cdot (N_{pet} + 1)$.

Сигналы SET_NEW_PETAL(n) дают еще $N_{pet} \cdot (N_{pet}+1) \cdot W_{bud}$ состояний, так как содержат как параметр смещение относительно клетки M.

Сигналы DELETE_PETAL(n) дают $3 \cdot N_{pet} \cdot (N_{pet}+1)$, так как SIDE может быть равен CENTER, LEFT и RIGHT.

Сигналы GROW_TOP_LEFT, GROW_TOP_RIGHT и CALM добавляют еще $3 \cdot (N_{pet}+1)$ состояния.

Всего: $(18 \cdot H_{pet} \cdot N_{pet} + N_{pet} \cdot W_{bud} + 3 \cdot N_{pet} + 3) \cdot (N_{pet}+1)$.

Клетки PETAL_TYPE(n) аналогично клеткам BUD дают $N_{pet} \cdot (24 \cdot H_{pet} + 3)$.

Клетка M проводит все сигналы, кроме DELETE_CELL, GROW_PETAL_UP_DIRECT(n), GROW_BUD_DOWN_DIRECT, но при этом SIDE=CENTER. Всего $6 \cdot H_{bud} + 6 \cdot H_{pet} \cdot N_{pet} + N_{pet} \cdot W_{bud} + N_{pet} + 3$.

Просуммировав, получим $30 \cdot H_{bud} + (18 \cdot N_{pet} \cdot N_{pet} + 48 \cdot N_{pet}) \cdot H_{pet} + (N_{pet} \cdot N_{pet} + 2 \cdot N_{pet}) \cdot W_{bud} + (3 \cdot N_{pet} \cdot N_{pet} + 7 \cdot N_{pet} + 6)$.

Осталось оценить число состояний клеток ML и MR. Оно равно $3 \cdot NT$, где NT — максимальное значение TIMER для этих клеток. Очевидно, что NT на 1 больше, чем целая часть квадратного корня из числа сигналов, которые необходимо послать.

Оценим сначала число необходимых преобразований. Очевидно, что для построения ПРЦ для двух любых «цветков» $N_{pet} \cdot H_{fl} \cdot W_{fl}$ преобразований заведомо хватит («цветки» целиком содержатся в прямоугольнике размером $H_{fl} \cdot W_{fl}$, а из построении ПРЦ видно, что любая клетка этого прямоугольника может из «отростка_n» превратится в клетку другого типа и наоборот только один раз для каждого «отростка»). Значит, общее число преобразований не превышает $N_{fl} \cdot N_{pet} \cdot H_{fl} \cdot W_{fl}$.

Все преобразования, кроме (1), (4), (8) и (11), реализуются одним сигналом, и достаточно трех сигналов CALM следом за ними, чтобы избежать наложения сигналов. Но при оценке числа преобразований, мы вместо одного применения указанных выше преобразоваий считали W_{fl} (как максимальную возможную ширину уровня). Поэтому и для реализации преобразований (1), (4), (8) и (11) хватит $4 \cdot W_{fl}$ сигналов. А всего хватит $4 \cdot N_{fl} \cdot N_{pet} \cdot H_{fl} \cdot W_{fl}$ сигналов. Значит, число состояний клеток ML и MR есть $6 \cdot (\lfloor\sqrt{N_{fl} \cdot N_{pet} \cdot H_{fl} \cdot W_{fl}}\rfloor + 1)$.

Первая из полученных оценок числа состояний (для всех клеток, кроме MR и ML) является практически точной и достигается почти всегда. Оценка, полученная для числа состояний клеток MR и ML, является грубой и сильно зависит от ПРЦ. Реальное число состояний может приблизиться к этой оценке только в случае построения по двум фигурам, либо в случае, если фигуры чередуют большие цветки и маленькие, что практически невозможно.

Список литературы

[1] Кудрявцев В. Б., Алешин С. В., Подколзин А. С. Введение в теорию автоматов. — М.: Наука, 1985.

[2] Кудрявцев В. Б., Подколзин А. С., Болотов А. А. Основы теории однородных структур. — М.: Наука, 1990.

[3] Мур Э. Ф. Математические модели самовоспроизведения // Математические проблемы в биологии. — М.: Мир, 1966. С. 36–62.

[4] Улам С. Некоторые математические проблемы, связанные с процессом роста фигур // Математические проблемы в биологии. — М.: Мир, 1966. С. 63–77.

[5] Подколзин А. С. Об универсальных однородных структурах // Проблемы кибернетики. Вып. 34. — М.: Наука, 1978. С. 109–131.