

РАЗБОР ЗАДАЧИ «ПРОЖОРЛИВЫЙ ТЬЮРМИТ» КОНКУРСА КИО-2010

Вторая задача конкурса КИО-2010 называлась «Прожорливый тьюрмит». Мы приведем условие задачи, результаты участников обоих уровней, разберем решения участников, объясним, откуда в названии насекомого появилась «опечатка».

На квадратном поле 20 на 20 расположены яблоки. По полю перемещается тьюрмит и, если встречает на пути яблоко, съедает его. Задача участников состояла в том, чтобы заставить тьюрмита съесть как можно больше яблок. На рис. 1 и 2 изображены поля игры, соответственно, первого и второго уровней. В обоих уровнях поле не имеет границы, то есть если тьюрмит заходит за сторону, то появляется в аналогичной позиции с противоположной стороны.

Для управления тьюрмитом его необходимо было запрограммировать. Тьюрмит – это маленькое простое насекомое,



которое способно принимать решения о своих действиях только на основе текущей сиюминутной поступающей к нему информации. У него нет ни памяти о том, где он был и что делал, и он не способен планировать свои действия ни на какое время вперед. Тьюрмит выбирает, что ему делать, на основе всего двух параметров: во-первых, видит ли он в данный момент перед собой яблоко, во-вторых, от своего текущего настроения (состояния). На каждом шаге тьюрмит может совершать только одно из трех действий: двигаться вперед, поворачиваться на месте налево или направо. Другими словами, программа для тьюрмита выглядит как описание, куда ему двигаться или поворачиваться, в зависимости от того, видит он или не видит перед собой яблоко, и в зависимости от настроения, в котором находится. Например, программа может заставлять тьюрмита идти вперед, если он находится в определенном состоянии и видит яблоко, и требовать повернуться влево, если он не видит яблока. Графически программа для тьюрмита изображается в виде *графа* – диаграммы, в которой состояния обозначены кругами, а инструкции по поведению в случае наличия или отсутствия яблок – соединяющими их стрелками. На рис. 3 приведен пример отрывка программы для тьюрмита. Рассмотрим, например, вершину 3. Она описывает поведение тьюрмита следующим образом: если тьюрмит нахо-

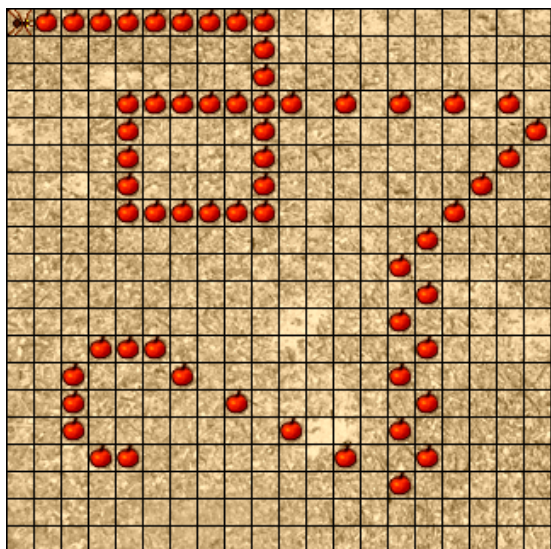


Рис. 1. Поле, уровень I

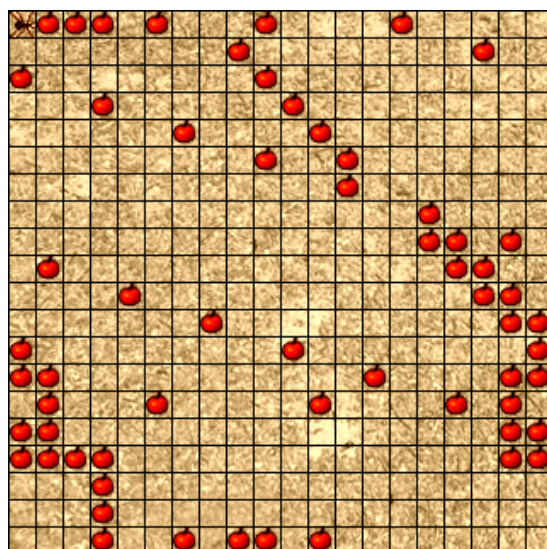


Рис. 2. Поле, уровень II

дится в состоянии 3 и перед ним яблоко (на стрелке подпись в виде круга), то необходимо изменить свое состояние на 4 (конец стрелки) и передвинутся вперед (под стрелкой подпись на стрелке в виде указателя вверх). Если яблока нет (подпись под стрелкой в виде зачеркнутого круга), то необходимо изменить свое состояние на 5 и повернуться налево или, что то же самое, против часовой стрелки. На рис. 3 только в состоянии 3 указаны оба действия: что делать, если яблоко есть и если яблока нет. Будем считать, что рис. 3 является лишь фрагментом программы, то есть пропущенные действия, возможно, есть, но не отображены. В конкурсе, если программа не назначала для тьюрмита никого действия, он останавливался на месте и ничего не делал.

Далее программу для тьюрмита мы будем называть *автоматом*, потому что именно так называются абстрактные устройства, которые в каждый момент времени находятся в одном из нескольких внутренних состояний, и, в зависимости от поступающей снаружи информации, изменяют свое состояние и производят какое-то действие.

По условию задачи конкурса требовалось не просто съесть как можно больше яблок, но при этом постараться использо-

вать в программе тьюрмита как можно меньше состояний. Смысл этого задания состоит в том, чтобы придумать простого тьюрмита, который изображает разумное поведение. Первоначальная оценка участников проходила по количеству съеденных тьюрмитом яблок, а если яблок съедено поровну, то сравнивалось, чей тьюрмит проще. Помимо прочего, чтобы участники не предлагали тривиальные решения, в которых тьюрмит последовательно, построчно проходит по всему полю, съедая без разбора все, что встретит, было введено ограничение на 200 шагов для первого уровня и на 170 для второго. (Все поле занимает $20 \times 20 = 400$ клеток). Один шаг соответствует одному действию тьюрмита, то есть движение вперед или поворот на месте.

Результаты участников распределились следующим образом. Из почти 600 участников первого уровня у 131 участника

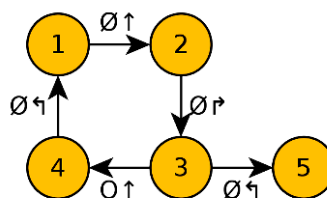


Рис. 3. Фрагмент программы тьюрмита

тюремнты съели все яблоки. При этом самый простой тюремнт у автора лучшего решения имел 17 состояний, а самый сложный тюремнт – 132 состояния. Промежуточные результаты распределились равномерно между 17 и 132 состояниями. Из тех тюремнтов, которые съели не все яблоки, 37 тюремнтов съели от 58 до 59 яблок, имея при этом от 19 до 131 состояния. На рис. 4 изображено лучшее решение первого уровня.

Во втором уровне все 60 яблок не съел никто, хотя решение, съедающее все яблоки существует. Далее мы обсудим принципиальные трудности, с которыми столкнулись участники, пытавшиеся «съесть» все яблоки.

Лучший тюремнт второго уровня съел 59 яблок, имея при этом 88 состояний, 7 тюремнтов съели 58 яблок, имея от 83 до 103 состояний. Далее еще 76 тюремнтов съели более 50 яблок, при этом они использовали от 16 до 171 состояний. Особенно хочется обратить внимание на то, что очень простой тюремнт всего из 16 состояний (а это в пять с половиной раз проще лидера) умудрился съесть 51, то есть почти все яблоки. Он даже направлялся доедать оставшиеся, но был остановлен закончившимися 170 шагами. Это решение мы привели на рис. 5.

Основным различием задач первого и второго уровня являлось то, что участни-

ки первого уровня имели простое расположение яблок, и тюремнт мог съесть их все, идя по очевидному маршруту. Участники первого уровня должны были направить основные усилия на то, чтобы разобраться, как создать автомат, который проведет тюремнта по маршруту, а потом оптимизировать автомат, чтобы он имел как можно меньше состояний.

Для участников второго уровня проблема возникала еще на этапе придумывания маршрута. Чтобы съесть за 170 шагов все яблоки, тюремнта надо вести по вполне определенному маршруту, любой лишний шаг или поворот в сторону означает, что тюремнт не успеет съесть всё. Участник, который реализовал бы этот маршрут, независимо от количества используемых состояний, победил бы в конкурсе, потому что все яблоки не съел никто. Обсудим этот маршрут позже.

Разберем решение задачи первого уровня. Как было сказано, маршрут тюремнта, по которому ему следует идти, очевиден. Для любого заранее заданного маршрута существует автомат, который проведет по нему тюремнта. Построим такой

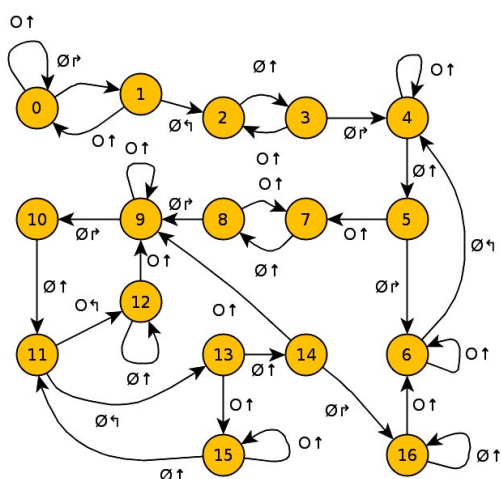


Рис. 4. Лучшее решение уровня 1 (Заварихина Екатерина, г. Вольск Саратовской обл.)

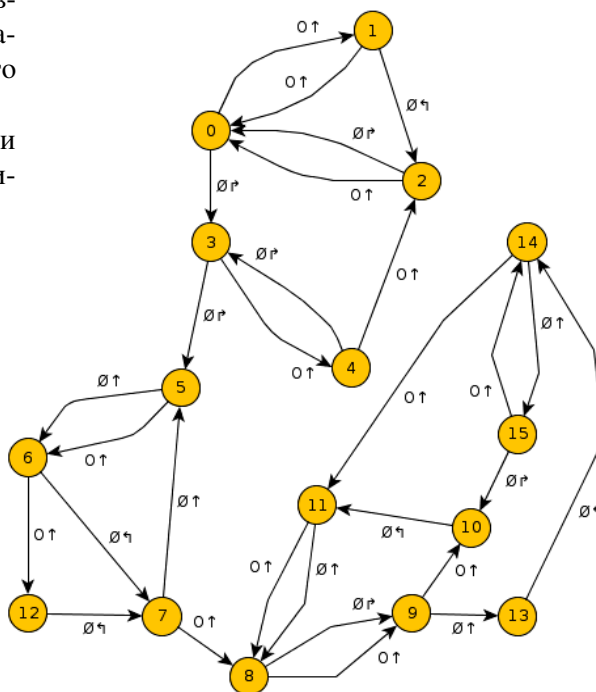


Рис. 5. Решение II уровня, 16 состояний, 51 яблоко (Стрельцова Екатерина, г. Майкоп)

автомат. Для каждого шага тьюрмита следует завести одно состояние. Первое состояние соответствует начальному положению тьюрмита, второе состояние соответствует положению тьюрмита после первого шага, третье – после второго и т. д. В каждом состоянии, независимо от того, видит тьюрмит яблоко или нет, его надо заставлять делать требуемое действие и переходить в следующее состояние. Маршрут для задачи первого уровня имеет длину порядка 125 шагов, поэтому и автомат, ведущий по маршруту, будет иметь порядка 125 состояний. Некоторые участники реализовали этот подход, они съели все яблоки, но в их автоматах было более 125 состояний, и они проиграли тем, кто придумал более эффективные решения. Разобранный только что автомат соответствует очень сложному тьюрмиту, который держит в своей голове полный маршрут и целенаправленно по нему идет.

Для оптимизации автомата и создания простого тьюрмита можно заметить, что маршрут состоит из нескольких отрезков пути, каждый из которых содержит повторяющиеся расположения яблок. Первые пять отрезков – это яблоки, расположенные горизонтально или вертикально друг за другом. Следующий отрезок – это яблоки, расположенные горизонтально через одну клетку, далее, яблоки расположены по диагонали вниз и влево. Разберем способ оптимизации автомата, который ведет тьюрмита по маршруту вниз и влево. Нам будет достаточно всего четырех состояний, но многие участники для того же самого справлялись и тремя.

Пусть тьюрмит уже съел первое из шести расположенных по диагонали яблок и смотрит *вниз*. (он стоит на самой правой клетке в пятой строке сверху). Не будем заставлять термита помнить все действия, с помощью которых он съест яблоки, научим его только одному шаблону действия: идти вперед, поворачивать направо, идти вперед (съедая яблоко), поворачивать налево. Яблоки расположены таким образом, что если повторять только эти действия, то будут съедены все яб-

локи на диагональном отрезке маршрута. Каждому действию необходимо сопоставить состояние, последнее состояние возвращает тьюрмита в исходное. Реализующий это поведение автомат изображен на рис. 3.

Действие, переводящее тьюрмита из состояния 3 в состояние 5 необходимо, чтобы не дать ему зациклиться. Тьюрмит должен остановиться, когда съедены все диагональные яблоки, и приступить к поеданию яблок на следующем отрезке маршрута. Переход из состояния 3 в состояние 5 происходит в тот момент, когда тьюрмит должен съесть очередное диагональное яблоко, но не обнаруживает его, потому что они кончились. Состояние 5 должно реализовывать шаблонные действия, необходимые для следующего отрезка маршрута, то есть движения вниз.

Итак, программа для прохождения каждого отрезка пути состоит всего из нескольких состояний, что позволяет значительно уменьшить размер автомата с числа, большего сотни, всего до нескольких десятков. Дальнейшие улучшения можно произвести за счет переиспользования состояний. Например, первые пять отрезков пути являются прямолинейными, и проходить их надо, совершая действие вперед. Это действие может совершаться одними и теми же состояниями. Решение на рис. 4 демонстрирует, как можно пройти первые пять отрезков, имея всего 4 состояния. Это решение переиспользует значительное количество состояний, поэтому оказывается таким маленьким.

Еще один секрет решения на рис. 4 состоит в том, что путь, который проходит тьюрмит, не совпадает с «очевидным». После диагонального отрезка яблок тьюрмиту следовало бы идти вниз, съедая попеременно яблоки справа и слева. Но тьюрмит идет вниз, съедая только яблоки, расположенные справа, и возвращается за яблоками слева в самом конце решения после того, как съест оставшиеся. Путь тьюрмита изображен на рис. 6. Такое изменение маршрута позволяет значительно сократить количество состояний, потому

что нет необходимости создавать группу состояний для реализации специального маршрута, съедающего попеременно левые и правые яблоки.

В задаче второго уровня, как было отмечено ранее, первым делом участникам требовалось построить эффективный маршрут. Маршрут, съедающий все яблоки изображен на рис. 7. Он имеет 168 шагов, то есть как раз укладывается в требуемые 170. Найти его было трудно, потому что участники не имели инструментов для прокладки маршрута, они могли либо рисовать и считать шаги на бумаге, либо передвигать тьюрмита с помощью программы. В обоих случаях перебор маршрутов требует времени.

Автомат, ведущий по маршруту, имеет порядка 170 шагов, но поддается оптимизации, потому что многие отрезки пути могут быть пройдены с помощью повторяющихся движений – по прямой, по диагонали и т. п.

Вернемся, наконец, к названию насекомого, поедающего яблоки. Префикс «тью», конечно же, является отсылкой к имени Алана Тьюринга или, точнее, описанной им машине Тьюринга – абстрактному устройству, которое способно исполнять произвольные алгоритмизуемые вычисления, основываясь всего на одной

простой операции. Машина Тьюринга состоит из бесконечной ленты, разделенной на клетки, в которых написаны символы, и читающей/пишущей головки. На каждом шаге, в зависимости от символа, находящегося под головкой, и состояния (аналогично состояниям тьюрмита), машина определяет, какой символ написать вместо того, который был, какое движение произвести – подвинуться налево, направо, остаться на месте – и в какое состояние перейти. Машина Тьюринга используется в теории алгоритмов, потому что, с одной стороны, исполняет всего одну операцию и по этой причине удобна для исследований, с другой стороны, по вычислительным возможностям эквивалентна многим другим устройствам, включая современные компьютеры. Поэтому полученные для Машины Тьюринга результаты часто могут быть распространены и на них.

Машина Тьюринга может быть двумерной. В этом случае лента заменяется на бесконечное клетчатое поле, а движения совершаются не только горизонтально, но и вертикально. Двумерный вариант машины Тьюринга напоминает тьюрмита, хотя тьюрмит значительно отличается от нее тем, что имеет ограниченное поле и не способен управлять символами на поле, только лишь заменять яблоко на съеден-

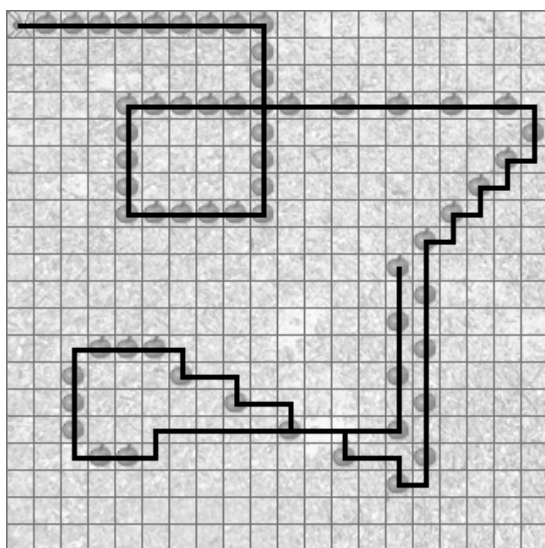


Рис. 6. Путь тьюрмита из лучшего решения уровня 1

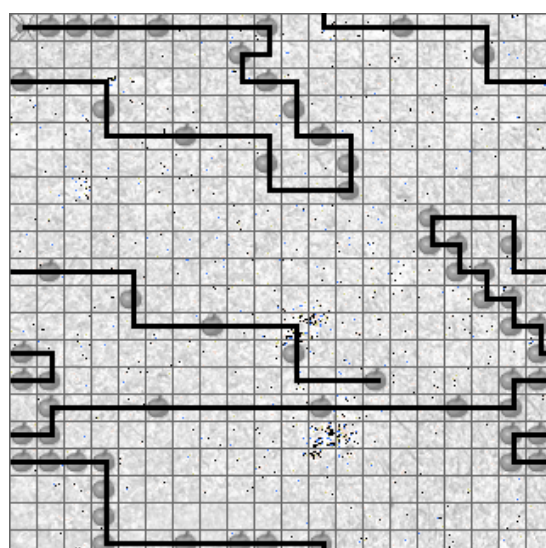


Рис. 7. Путь тьюрмита, уровень II, 168 шагов, съедены все яблоки

ное яблоко. Стоит отметить, что одномерная машина Тьюринга способна совершать все те же вычисления, что и двумерная. Другими словами их вычислительные мощности совпадают.

Тьюрмит и Машина Тьюринга объединены тем, что их сложное поведение основано всего на нескольких простых командах или правилах. Существуют и другие ситуации, когда сложный процесс имеет простую основу. Упомянем в этом отношении фракталы и роевой интеллект. С помощью фракталов на основе одной или нескольких формул удается получить живописные изображения. Фракталы иногда используются в компьютерных играх для изображения облаков и крон деревьев, потому что нет необходимости фотографировать, рисовать или хранить изображения этих объектов, они могут быть быстро сгенерированы автоматически в нужный момент с использованием нескольких формул.

Роевой интеллект позволяет группе животных (насекомых, рыб, птиц) в совокупности совершать осмысленные действия, при том что каждое отдельное существо само по себе исполняет только несколько простых правил. Муравьи могут искать кратчайшие маршруты между точками (муравейником, источником еды), для этого каждый муравей знает только, что ему в определенных ситуациях необ-

ходимо оставлять пахучие следы и ходить вдоль пахучих следов, оставленных другими. Это поведение муравьев легло в основу нескольких муравьиных алгоритмов, используемых для построения маршрутов в графах.

Для завершения опишем алгоритм, позволяющий имитировать поведение птиц в стае, подобные алгоритмы используются в фильмах, компьютерных играх для искусственного создания эффекта реальной стаи птиц. Представим, что в стае на поведение каждой птицы влияют не все птицы, а только некоторые, находящиеся поблизости. Область влияния может быть шаром или быть несимметричной, например, она больше спереди, чем сзади, потому что птица лучше видит то, что впереди. Каждая птица перемещается, объединяя три следующих движения:

1) движение в направлении среднего положения птиц из области влияния,

2) движение в направлении от других птиц, приблизившихся слишком близко,

3) движение в том же направлении, что и среднее направление всех птиц области влияния. Кроме этого, стая сможет правдоподобно огибать препятствия и улетать от хищника, если к этим правилам добавить еще два соответствующих. Визуализацию стаи птиц, основанную на этих правилах, можно найти в Интернете по ключевому слову *birds*.

**Посов Илья Александрович,
ассистент кафедры ВМ-2
СПбГЭТУ «ЛЭТИ».**



Наши авторы, 2010.
Our authors, 2010.