Источник: Гуревич Л.А.,А.Н. Вахитов Мультиагентные системы // Введение в Computer Science. – 2005. –с.116-139.

Мультиагентные системы

Гуревич Лев и Вахитов Александр *

1 Введение

Мультиагентные системы созданы для решения различных задач искусственного интеллекта, в которых присутствует несколько участников. Основным понятием является *агент*.

Определение 1. *Агент* - нечто, что способно воспринимать свое окружение через сенсоры и изменять его своими действиями.

Современный подход к искусственному интеллекту основано на понятии *рационального агента*, который всегда старается оптимизировать соответствующую меру полезности своих действий. Например, люди могут рассматриваться как агенты с глазами в качестве сенсоров и способные что-то делать при помощи рук. Роботы способны воспринимать мир через камеры и передвигаться при помощи колес. Для программ графический интерфейс является средством и восприятия, и действия. Однако, агенты редко являются одиночными системами. Чаще они взаимодействуют друг с другом. Системы, содержащие группу агентов, которые могут взаимодействовать между собой и называются *мультиагентными системами*.

2 Характеристики мультиагентных систем

Какие характеристики отличают мультиагентные системы от одноагентных. Можно выделить следующие различия:

2.1 Структура агента

Часто агенты сконструированы по-разному, например программы, написанные разными людьми. Различия могут заключаться в оборудовании и программном обеспечении. Часто таких агентов называют *гетерогенными* в противоположность *гомогенным*, которые сконструированы идентичным образом и имеют изначально одинаковые возможности. Однако различие не так очевидно; агенты, основанные на одинаковых оборудовании и программах, но ведущие себя по- разному, тоже могут называться гетерогенными.

*конспект: Макарова Алёна с нашей помощью

2.2 Окружение

Окружение агентов может быть статическим (не зависящим от времени) или динамическим (не стационарным). Из-за простоты обработки и более строгого математического описания большинство методов искусственного интеллекта в одноагентном случае были разработаны для статического окружения. В мультиагентных системах же хотя бы само присутствие нескольких агентов делает окружение динамичным.

2.3 Восприятие

Информация, которой обладает система агентов, является обычно распределенной: агенты могут следить за окружением из разных положений, получать информацию в различные моменты времени или интерпретировать эту информацию по - разному. Таким образом, состояние окружения является частично обозримым для каждого агента.

3 Рациональный агент

Здесь будет описана проблема принятия оптимального решения. Это означает, что агенту на каждом шаге следует выбрать наилучшее действие, исходя из того, что ему известно об окружающем мире. Вводится т.н. мера полезности действий агента, которую он постоянно пытается оптимизировать. Рационального агента также называют *разумным*. В дальнейшем мы главным образом будем рассматривать только вычислительных агентов, т.е. тех, что созданы для решения специальных задач и приводятся в исполнение на вычислительных устройствах.

3.1 Принятие решения агентом

Предположим, что на каждом временном шаге $t = 1, 2, \ldots \infty$ агент может из конечного набора возможных действий A выбрать какое-то действие a_t. Интуитивно понятно, что, чтобы действовать рационально, агент должен оценивать и прошлое, и будущее при выборе дальнейших действий. Под прошлым подразумевается то, что агент воспринял и какие действия предпринял до момента времени t, а под будущим - что он ожидает и что собирается потом делать. Если обозначим o_τ наблюдение агента в момент времени τ, тогда для выбора оптимального действия в момент времени t в общем случае необходимо использовать всю историю наблюдений o_τ и историю действий a_τ, предшествующую моменту времени t.

Определение 2. Функция $\pi(o_1, a_1, o_2, a_2, ..., o_t) = a_t$, которая отображает набор пар наблюдение-действие до момента времени t в оптимальное действие a_t называется *стратегией* агента.

Если сможем найти функцию π, осуществляющую данное отображение, то часть задачи об отыскании оптимального решения на основе прошлого

будет решена. Однако, определение и реализация такой функции проблематично; сложная история может содержать большое количество пар наблюдений, которые могут меняться от одной задачи к другой. Более того сохранение всех наблюдений требует очень большого объема памяти, а соответственно и росту сложности вычислений. Этот факт приводит к необходимости использования более простых стратегий. Например, агент может игнорировать всю историю наблюдений за исключением последнего. В этом случае стратегия принимает вид $\pi(o_t) = a_t$, который отображает текущее восприятие агента в действие.

Определение 3. Агент, который, отображает текущее восприятие o_t в новое действие a_t называется *рефлексивным*, а его стратегию называют *реактивной* или *стратегией без памяти*.

Возникает естественный вопрос: насколько хорошим может быть такой рефлексивный агент? Как мы увидим дальше, он может быть довольно неплохим.

3.2 Окружение и свойство Маркова

Из сказанного выше следует, что понятия окружения и агента являются спаренными, так что одно из них не может быть определено без другого. Фактически, различие между агентом и его окружением не всегда отчетливо и это иногда осложняет проведению четкой границы между ними. Для простоты предположим, что существует мир, в котором есть один или более агентов, в котором они воспринимают, думают и действуют.

Определение 4. Коллективная информация, которая содержится в окружающем мире в момент времени t, и которая важна для исполняемой задачи, называется *состоянием мира* и обозначается s_t. Множество состояний мира обозначим через S.

В зависимости от природы задачи, мир может быть *дискретным* или *непрерывным*. Дискретный мир характеризуется конечным числом состояний. Примером может служить игра в шахматы. С другой стороны примером непрерывного мира может служить мобильный робот, который может свободно передвигаться в декартовой системе координат. Большинство из существующих технологий искусственного интеллекта созданы для дискретного мира, поэтому мы будем в дальнейшем рассматривать именно его.

3.3 Наблюдаемость

Важное свойство, характеризующее мир с точки зрения агента, связано с восприятием. Мы будем говорить, что мир полностью наблюдаем, если текущее восприятие агента o_t полностью описывает состояние мира s_t. В противоположность этому, в частично наблюдаемом мире текущее восприятие o_t описывает лишь часть информации о состоянии мира задающую вероятностное распределение $P(s_t|o_t)$ между действительными состояниями мира. Таким образом, s_t можно рассматривать как случайную величину, распределенную

на S. Частичная наблюдаемость может быть вызвана двумя фактами. Первый из них это помехи в сенсорной информации агента. Например, вследствие несовершенства сенсоров, в разные моменты времени агент может воспринимать одно и то же состояние мира по разному. Возможно так же что для агента некоторые состояния неразличимы.

3.4 Свойство Маркова

Рассмотрим снова рефлексивного агента с реактивной тактикой $\pi(o_t) = a_t$ в полностью обозреваемом мире. Предположение обозреваемости подразумевает, что $s_t = o_t$, и таким образом тактика агента $\pi(s_t) = a_t$. Другими словами, в обозреваемом мире тактика рефлексивного агента является отображением из состояний мира в действия. Выгода состоит в том, что во многих задачах состояние мира в момент времени t дает полное описание истории до момента времени t.

Определение 5. Про такое состояние мира, которое содержит всю важную информацию о прошлом в конкретной задаче, говорят, что оно является Марковским или обладает свойством Маркова.

Из выше сказанного мы можем сделать вывод, что в Марковском мире агент может безопасно использовать стратегию без памяти для принятия решения вместо теоретически оптимальной стратегии, которая может требовать много памяти.

До сих пор мы рассматривали, как стратегия агента может зависеть от последнего события и отдельных характеристик окружения. Однако, как мы обсуждали вначале, принятие оптимального решения может также браться из оценки будущего.

3.5 Стохастические переходы

Как было упомянуто выше, в каждый момент времени t агент выбирает действие a_t из конечного множества действий A. После выбора агентом действия мир меняется как его следствие. Модель перехода(иногда называют моделью мира) определят как меняется мир после совершения действия. Если текущее состояние мира s_t и агент совершает действие a_t, мы можем выделить два случая:

- В *детерминистическом* мире, модель перехода отображает единственным образом пару состояние-действие (s_t, a_t) в новое состояние s_{t+1}. В шахматах например, каждый ход изменяет игровую позицию.
- В *стохастическом* мире модель перехода отображает пару состояние - действие в распределение вероятности $P(s_{t+1}|s_t, a_t)$ состояний. Как и в рассмотренном выше случае с частичной обозреваемостью, s_{t+1} это случайная величина, которая может принимать все возможные значения в множестве S с соответствующей вероятностью $P(s_{t+1}|s_t, a_t)$.

Самые практические приложения включают стохастические модели перехода, например, движение робота неточно из-за того, что его колеса скользят и т.п.

Как мы могли заметить, иногда частичная обозреваемость является следствием неточности восприятия агентом его окружения. Здесь мы видим другой пример, где неточность играет роль, а именно, мир меняется, когда совершается действие. В стохастическом мире эффект действия агента не известен заранее. Вместо этого есть случайный элемент, который определяет как меняется мир вследствие действия. Ясно, что стохастичность при переходе из одного состояния в другое вносит дополнительные сложности в задаче принятия оптимального решения агентом.

Для классического искусственного интеллекта целью отдельной задачи является желаемое состояние мира. Таким образом, *планирование* определяется как поиск оптимальной стратегии. Когда мир детерминистический, планирование переходит в поиск по графу для каждого варианта существующего метода. В стохастическом же мире может не перейти в простой поиск по графу, так как переход из состояния в состояние не является детерминистическим Теперь агент при планировании принимать в расчет неточности переходов. Чтобы понять, как это может быть использовано, заметим, что в детерминистическом мире агент предпочитает по умолчанию конечное состояние неконечному. В общем случае, агент сохраняет настройки при переходе из состояния в состояние. Например, робот, играющий в футбол, будет стараться забить очко, стараться меньше стоять с мячом перед пустыми воротами и т.п. Чтобы формализовать это понятие, свяжем с каждым состоянием s вещественное число $U(s)$, называемое полезностью состояния этого агента. Формально, для двух состояний s и s^* выполняется $U(s) > U(s^*)$ тогда и только тогда, когда агент предпочитает состояние s состоянию s^*, а $U(s) = U(s^*)$, если для агента эти состояния неразличимы. Интуитивно понятно, что чем выше полезность, тем выгоднее состояние, в котором находится агент. Заметим, что в мультиагентных системах желаемое состояние для одного из агентов может не быть желаемым для остальных. Например, в футболе забивание гола является нежелательным для противоположной команды.

3.6 Принятие решения в стохастическом мире

Теперь возникает вопрос, как агент может эффективно использовать функции полезности для принятия решения. Предположим, что у нас мир стохастический с моделью перехода $P(s_{t+1}|s_t, a_t)$, находящийся в состоянии s_t до тех пор, пока агент обдумывает какое действие ему совершить. Пусть $U(s)$- функция полезности состояния какого-то агента. Предположим, что у нас только один агент. Тогда принятие решения основано на предположении, что оптимальное действие агента должно быть максимумом полезности, т.е.

$$a_t^* = argmax \sum_{s_t+1} P(s_{t+1}|s_t, a_t)U(s_{t+1}),$$

где суммирование происходит по всевозможным состояниям s_{t+1}. Это значит, что чтобы увидеть насколько полезно действие, агент должен умножить полезность каждого возможного конечного состояния на вероятность попадания в это состояние, и потом просуммировать полученный результат. Тогда агент может выбрать действие a_t^*, у которого будет максимальная сумма. Если каждое состояние мира имеет величину полезности, агент может произвести вычисления и выбрать оптимальное действие для каждого возможного состояния. Тогда агент со стратегией может оптимально переводить состояние в действие.

4 Теория игр

Здесь мы обсудим принятие решения в мультиагентных системах, где группа агентов взаимодействуют и одновременно принимают решение. Мы используем теорию игр для анализа задачи, в частности, стратегических игр, где мы изучим итеративное исключение точно определенных действий и равновесие Нэша. Как мы уже видели, действие агента на окружение может оставаться неопределенным, и это может накладывать неопределенность на принятие решения. В мультиагентных системах, где агенты принимают решение в одно время, агент может не знать о решении других участвующих агентов. Ясно, что агент должен действовать в зависимости от действий других агентов. Мультиагентное принятие решения является предметом *теории игр*. Теория пытается предугадать поведение взаимодействующих агентов в состоянии неопределенности и основывается на двух предположениях. Во-первых, взаимодействующие агенты рациональны, а во-вторых, принятие решения происходит в зависимости от решения других агентов. В зависимости от способа принятия решения агентами мы можем выделить два типа игры. В *стратегической* игре каждый агент выбирает свою стратегию только один раз в начале игры, а затем все агенты производят действия одновременно. В игре в обобщенной формы агенты могут менять стратегию на протяжении всей игры. Другое отличие состоит в полноте или не полноте известности информации о других агентах. Будем рассматривать только игры, где агент владеет всей информацией, касающейся других агентов.

4.1 Стратегические игры

Стратегическая игра(также называемая *нормальной формой*) является простейшей моделью взаимодействия агентов в теории игр. Ее можно рассматривать как мультиагентное расширение прошлой модели, и характеризующееся следующими свойствами:

- Агентов не меньше одного ($n > 1$)
- Каждый агент i выбирает действие a_i(называемое *стратегией*) из собственного множества действий A_i. Вектор $(a_1, ?, a_n)$, т.е. действие, называется *совместным действием* и обозначается (a_i). Мы будем пользоваться обозначением a_{-i} для обозначения всех действий агента

кроме i-го, а также (a_{-i}, a_i) для обозначения совместного действия, где агент i совершает действие a_i.

- Игра происходит в фиксированном состоянии s. Состояние может быть определено как содержащее n агентов, их множества действий A_i и их выигрыши.
- Каждый агент i имеет собственное значение функции действия $Q^*_{i(s,a)}$ которая вычисляет полезность совместного действия для агента i. Каждый агент может отдавать разные предпочтения разным совместным действиям. Мы предполагаем, что функция выигрыша заранее определена и фиксирована.
- Состояние полностью обозреваемо для агентов. То есть они знают друг о друге, множества действий друг друга и выигрыши друг друга. Все это в игре и есть общее знание агентов.
- Каждый агент выбирает единственное действие. Более того, агенты выбирают свои действия одновременно и независимо. Ни один агент не знает о принятии решения другого агента до тех пор, пока решение не будет принято.

В итоге в стратегических играх каждый агент выбирает одно действие и получает результат в зависимости то выбора общего действия. Это совместное действие называется *исходом* игры. Хотя функция выигрыша агента является общим знанием, агент не знает заранее о выборе действия другим агентом. Самое лучшее, что он может - так это попытаться угадать действие другого агента. *Решением игры* является предсказание исхода, используя предположение, что все агенты рациональные и стратегические.

4.2 Итеративное исключение строго доминирующих действий

Первое важное понятие основано на предположении, что рациональный агент никогда не выбирает подоптимальное решение. Под подоптимальным решением мы подразумеваем действие, которое вне зависимости от того, что делают другие агенты, всегда будет в результате менее выигрышным для агента, чем какое-то другое действие. Сформулируем это иначе:

Определение 6. Будем говорить, что действие a_i агента i является строго доминирующим над другим действием a'_i, если $u_i(a_{-i}, a'_i) > u_i(a_{-i}, a_i)$ для всех действий a_{-i} других агентов.

В этом определении $u_i(a_{-i}, a_i)$ является выигрышем агента i, который получается, если он выберет действие a_i пока остальные агенты выполняют действие a_{-i}

Характеристикой итеративного исключения строго доминирующих действий является то, что агенты могут не сохранять веру в стратегии других агентов для того, чтобы вычислить их оптимальное действие. Единственное, что требуется это предположение в общем знании, что все агенты рациональны.

4.3 Равновесие Нэша

Равновесие Нэша является более значительным понятием чем *итеративное исключение строго доминирующих действий* в том смысле, что она порождает более точное предсказание результатов в широком классе игр. Это формально может быть определено следующим образом:

Определение 7. Равновесие Нэша это общее действие a^* такое, что для каждого агента i выполняется $u_i(a^*_{-i}, a^*_i) \geq (a^*_{-i}, a_i)$ для всех действий $a_i \in A_i$

Заметим, что в отличие от итеративного исключения строго доминирующих действий (ИИСДД), которое описывает решение игры посредством алгоритма, Равновесие Нэша описывает в терминах возможных состояний. В равновесие Нэша каждого агента является оптимальным ответом на действие другого агента.

Определение 8. Совместное действие a называется оптимальным действием *Парето*, если ни для какого из действий a' не выполняется $u_i(a') > u_i(a)$ для любого агента i.

Выше мы предположили, что во время игры агент i будет выбирать действие, которое берется из множества его действий A_i. Однако, это не всегда так. Во многих случаях у агента есть причина проявлять стохастичность в своем поведении. В общем случае мы можем предположить, что агент i выбирает действие a_i в процессе с некоторым распределением вероятностей $p_i(a_i)$, которое может быть разным для разных агентов.

Определение 9. Смешанная стратегия агента i это распределение вероятностей действий $a_i \in A_i$

В своей известной теореме Нэш показал, что стратегическая игра с конечным числом агентов и конечным числом действий всегда достигает равновесия в смешанной стратегии.

5 Координация

Рассмотрим задачу координации, то есть, как отдельное решение агента может повлиять на принятие общего решения. Различающееся будущее мультиагентных систем фактически означает, что процесс принятия решения может быть распространенным. Это значит, что нет центрального агента, который контролирует, что делает каждый агент в каждый момент времени. Каждый агент сам решает, какое действие он совершит. Координация реализуется с помощью распределения агентов по ролям. То есть агент, понимая, что он наиболее подходит для выполнения определенной роли, присваивает ее себе. Кроме того, может существовать заранее определенный набор правил, которыми должны руководствоваться агенты. Например, таким правилом является пропускание помехи справа в правилах дорожного движения. Другим

примером может послужить игра роботов в футбол. Команда роботов старается забить мяч в ворота противоположной команды. Здесь координация обеспечивает то, что например, два робота из одной команды не пытаются ударить мяч в один одновременно. Здесь нет контролирующего агента, который мог бы инструктировать роботов в реальном времени бить или не бить мяч. Они сами должны это решать.

6 Общее знание

Раньше мы предполагали, что мир полностью обозрим. Теперь ослабим это предположение и рассмотрим случай, где некоторые состояния невидимы для агента. В частично обозреваемом мире агент должен всегда размышлять о том, что он знает об окружающем мире и том, что знают другие агенты, прежде чем принять решение. Чтобы действовать рационально, агент должен всегда реагировать на то, что он узнает о текущем состоянии мира. Если мир полностью обозреваем, то агент может действовать практически не размышляя. В частично обозреваемом мире агент должен внимательно рассматривать, что он знает и что не знает перед выбором действия. Интуитивно понятно, что чем больше агент знает о состоянии, тем лучшее решение он может сделать. В мультиагентных системах рациональный агент может принимать в расчет знание других агентов о состоянии(то есть информацию, которой они владеют о состоянии). Также он должен рассматривать то, что другие агенты знают о его действиях и знают ли они что знает ли он о том, что он о них знает. То, что два рациональных агента всегда хотят заключить пари, не является общим знанием.

В качестве примера приведем головоломку о шапках. Трое агентов(например, девушки) сидят вокруг стола, на каждой из них надета шапка, которая может быть либо белой либо красной. Каждый агент знает цвет шапки двух других, но не знает цвета своей. Человек, который наблюдает за ними со стороны (ведущий), просит их повернуться и спрашивает, знают ли они какого цвета их шапка. Каждый агент дает отрицательный ответ. Тогда ведущий объявляет, что, по крайней мере, на одном из них красная шапка, а затем снова просит их повернуться. Первый агент говорит Нет, второй агент говорит Нет, но, когда он спрашивает третьего агента, он отвечает Да. Как получилось, что третий агент в конце может определить цвет своей шапки.

Если у агентов одинаковые стратегии и им известно о том, что делают агенты в текущем состоянии, то агенты должны делать одинаковые действия

7 Коммуникация

Мультиагентное взаимодействие часто ассоциируется с некоторой формой общения. Мы используем общение в повседневной жизни, для совместного решения задач, для того, чтобы договариваться с другими, или просто для

обмена информацией с другими. Как мы видели, в головоломке о шляпах путем ответов на вопросы агенты смогли сформулировать более точное утверждение задачи и, в конечном счете, решить ее. Когда мы говорим о вычислительных агентах, общение рассматривается на некотором уровне абстракции. На более низком (network) уровне можно сказать, что можно быть уверенным, что сообщения, пересылаемые агентами друг другу, безопасно и вовремя дойдут до адресатов. Для этого существуют протоколы. На среднем языковом уровне можно основываться на множестве основных слов языка и их стандартных изменениях, так что агенты, говорящие на одном языке, могли понимать друг друга. И последний уровень- уровень применения, т.е. можно эффективно использовать общение для решения стандартных мультиагентных задач, например, координация или согласование. Как мы уже поняли, состояние мира характеризуется количеством агентов, их возможными действиями и выгодой агентов, если они вовлечены в стратегическую игру, и другими аспектами внешнего окружения. В таких случаях какой-то агент может быть дезактивирован в некоторых состояниях, потому что ему присвоена некоторая роль. Аналогично, если состояние мира частично обозреваемо для агентов, каждый агент обладает различными уровнями знания о настоящем состоянии мира. Формально можно описать коммуникацию рассмотрением каждой первообразной коммуникации как действия, которое обновляет знание агентов о состоянии. Наиболее общие типы процесса коммуникации(они используются,например, в правилах дорожного движения):

- *Информирующие*
- *Предупреждающие*
- *Подтверждающие*
- *Запрещающие*
- *Указывающие*

Каждый процесс коммуникации может повлиять на знания агентов по-разному. С другой стороны он может быть использован для информирования о выборе действий агентом. Можно дать простую интерпретацию другому процессу коммуникации. Например, запрещение может играть роль, запрещающую деактивацию какого-то действия в отдельной ситуации. Указывающий процесс может проявляться в организованной группе агентов, в которой агент с большим авторитетом может давать команды другим агентам. Некоторые языки коммуникации агентов могут получаться в объединениях агентов, которые направлены на стандартизацию процесса коммуникации. Действие можно рассматривать как действие, которое меняет знание вовлеченного агента о состоянии. Рассмотрев множество возможных коммуникационных актов, агент должен задуматься над вопросом какой акт использовать и кому передавать информацию. Мы приписываем телекоммуникационному акту значение такое, как индикатор полезности акта. Но откуда мы возьмем это значение? Структура, которая позволяет должным образом определить

значение коммуникации, это *Баессова* игра. Эта модель является стратегической игрой, но с частичной обозреваемостью. Мы имеем некоторое количество агентов, множество состояний мира, функцию информации для различную для каждого агента. Также предполагаем, что каждое попадание в состояние случается с некоторой вероятностью, которая равна для всех агентов, и что это определяет стратегическую игру с соответствующим выигрышем. Таким образом, агенты могут вычислять значение всех возможных коммуникационных актов в некоторой ситуации и затем выбрать одну, у которой значение самое большое. На практике, однако, принятие оптимального решения что передавать и кому может потребовать больших вычислительных затрат, которые могут уменьшить способности агента.

Преимуществом использования коммуникации является то, что больше не нужно пытаться угадывать действия других агентов. Путем коммуникации агенты могут определить способности друг друга и распределить роли в игре. На практике, однако, это не всегда возможно. Например, в частично обозреваемом мире. Когда общение между агентами возможно, агенты сами вычисляют свои способности к той или иной роли, а затем посылают эту информацию остальным агентам.

8 Структура агента

8.1 Самозаинтересованные агенты

Рассмотрим мультиагентную систему, где агенты могут кооперироваться. Тот факт, что агенты объединяются для достижения общей цели, приводит нас к разработке алгоритма, подобного координационному алгоритму, в котором агенты стараются передать как можно больше информации о себе и вести себя по инструкции. Футбольный робот, например, никогда не начнет присваивать чужую роль, так как это может потенциально нанести вред всей команде. На практике, однако, мы часто имеем дело с самозаинтересованными агентами, например, агенты, которые защищают интересы владельца, который хочет максимизировать свою прибыль. Типичный пример подобного программного агента это электронный аукцион в Интернете. Разработка алгоритма или протокола для подобной системы чаще встречающаяся задача, чем в кооперирующем случае. Во-первых, мы должны руководить агентом, участвуя в протоколе, который заранее тоже не известен. Во-вторых, нужно принимать во внимание факт, что может попытаться начать использовать протокол в собственных интересах, приводя к подоптимальному результату. Это не исключает возможности, что агент может начать врать, если это необходимо.

9 Обучение

Предположим, что наш мир стохастичен и пусть для каждого состояния мира определена некая функция $R(s)$, которая определяет полезность состояния s для нашего агента. Агент хочет найти такую последовательность действий

$\{a_t\}$ (политику, обозначенную как π), что в итоге он придет в максимально полезное состояние. То есть он хочет найти максимум по всем возможным политикам величины матожидания суммы ряда из $R(s_t)$ при t от единицы до бесконечности с уменьшающими коэффициентами γ^t (для того чтобы ряд сошелся).

Матожидание используется, так как мир не детерминирован, иначе нам бы были однозначно известны все перехода мира между состояниями после какого-то нашего действия, и мы бы поиском в графе переходов нашли оптимальный путь.

Определение 10. Эти R(s) называются *reinforcements*.

Простым и эффективным способом вычисления наибольшей полезности является метод *итерации присваивания*. Например, используя его в лабиринте, где ямы обозначены малой полезностью -10,все клетки полезностью $-1/30$ и финишные клетки (которых нужно достичь) полезностью 10, получим оптимальную полезность и оптимальную политику.

Чтобы посчитать $R(s)$ и $U^*(\pi)$, нужно брать в качестве начальных какие-то(более-менее разумные) значения R(s) и по ним считать $U(s)$ - полезности состояний - в процессе деятельности агента. Эти $U(s)$ считаются пошаговым присваиванием:

$$U(s) := R(s) + \gamma \max \sum P(s'|s,a)U(s')$$

, где максимум берется по всем a, а сумма- по всем s'. Присваивание происходит до тех пор, пока с некоторой точностью они не сойдутся к чему-то. Тогда и получится ответ - наилучшая политика.

9.1 Q-learning- алгоритм

Q-learning- алгоритм считает вместо $U(s)$ (полезностей состояния для агента) величины $Q(a,s)$(полезности действия a в состоянии мира s). Эти величины тоже вначале полагаются случайными, а потом сходятся к некоторым значениям в ходе итераций присваивания после каждого действия агента в процессе обучения.

$$Q(s,a) := (1-\lambda)Q(s,a) + \lambda(R + \gamma \max Q(s',a'))$$

, где $\lambda \in (0,1)$ а смысл величины λ в том, что это доля, на которую мы позволяем каждому присваиванию изменять значение $Q(s,a)$.