**АНАЛИЗ СНИЗУ ВВЕРХ И СВЕРХУ ВНИЗ**

“Сверху вниз” vs. “снизу вверх”, “прямой” vs. “обратный”, “управляемый данными” vs. “движимый целью” - три пары определений для таких терминов, как “цепной анализ”, “парсинг”, “синтаксический разбор”, “логический анализ” и “поиск”. В принципе, все эти термины отражают сходные отношения, и различие между ними состоит лишь в том, что они взяты из различных подобластей компьютерной науки и искусственного интеллекта (парсинг, системы с заложенными в них правилами, поисковые системы и системы, направленные на решение проблем и т.д.)

Суть этих противопоставлений можно проиллюстрировать на примере парадигмы поиска. Основная задача любого поиска состоит в том, чтобы определить маршрут, по которому вы будете перемещаться с настоящей позиции к вашей цели. Если вы начнете поиск с текущей позиции и будете продолжать его, пока не наткнетесь на желаемый результат, - это так называемый прямой поиск или поиск снизу вверх. Если вы мысленно ставите себя в то место, где вы хотите очутиться в результате поиска и определяете маршрут, двигаясь в обратном направлении, т.е. туда, где вы действительно находитесь в настоящий момент, - это поиск в обратном направлении или поиск сверху вниз. Обратите внимание на то, что, определив маршрут в результате обратного поиска, вам все же предстоит добраться до своей цели. Несмотря на то, что сейчас вы движетесь вперед, это не является прямым поиском, т.к. поиск уже был осуществлен ранее, причем в обратном направлении.

Эти же противопоставления можно рассмотреть на примере систем с встроенными правилами. Представим себе, что правило состоит из набора антецедентов и набора следствий. Когда система определяет, что все антецеденты определенного правила удовлетворены, это правило вызывается и выполняется (выполняется ли каждое вызванное правило зависит от специфики конкретной системы). После этого в базу знаний заносятся утверждения, полученные в результате выполнения правила, и выполняются соответствующие операции. Данный процесс происходит вышеописанным образом, независимо от того, применяет ли система прямой или обратный логический анализ. Чтобы проиллюстрировать различия между ними, следует отдельно рассмотреть процедуру активации правила. Вызываются только активированные правила. При прямом логическом анализе (снизу вверх), когда в систему добавляются новые данные, они сравниваются со всеми антецедентами всех правил. Если данные соответствуют антецеденту правила, то это правило активируется (если оно еще не является активированным), и если подобраны все антецеденты определенного правила, то оно вызывается. Утверждения, полученные в результате выполнения правила, заносятся в базу знаний и рассматриваются в качестве новых данных, сравниваются с антецедентами и могут вызвать активацию и вызов дополнительных правил. При обратном логическом анализе (сверху вниз) при добавлении данных правила не активируются. Когда система получает запрос, он сравнивается со всеми следствиями всех правил. Если запрос совпадает со следствием, то это правило активируется, а все его антецеденты рассматриваются в качестве вторичных запросов и могут вызвать активацию дополнительных правил. Когда запрос соответствует не ограниченному условием утверждению базы знаний, на него поступает ответ, и если этот запрос исходил от антецедента, считается, что он удовлетворяет последнему. Когда все антецеденты некоторого правила будут удовлетворены, правило вызывается и выполняется. При выполнении правила осуществляется ответ на запросы, которые его активировали, и теперь другие антецеденты считаются удовлетворенными и могут вызываться соответствующие им правила. Обратите внимание на то, что вызов и выполнение правила всегда происходит в прямой последовательности, а отличие прямого цепного анализа от обратного состоит в том, когда активируется правило.

**Примеры**

Парсинг. Попытаемся проиллюстрировать и объяснить разницу между синтаксическим анализом сверху вниз и снизу вверх на примере предложения “They are flying planes” и простой грамматики, представленной в виде пронумерованных правил:

1. S NP VP

2. NP N

3. NP PRO

4. NP ADJ N

5. VP VT NP

6. VT V

7. VT AUX V

8. N planes

9. PRO they

10. ADJ flying

11. AUX are

12. V are

13. V flying

Антецеденты указаны с правой стороны, а следствия - с левой. Например, правило 1 читается следующим образом: “Если последовательность состоит из именной группы (NP), за которой следует глагольная группа (VP), то эта последовательность является предложением (S).”

Синтаксический разбор сверху вниз начинается с символа S, который и будет являться вершиной дерева разбора. Эта процедура эквивалентна процедуре постановки задачи, которая заключается в том, чтобы определить, является ли последовательность слов предложением. Правило 1 гласит, что каждое предложение состоит из именной группы (NP), за которой следует глагольная группа (VP). При наличии нескольких правил, сперва применяется правило с наименьшим номером, а затем оно расширяется слева направо. Таким образом следующим шагом является нахождение первой связи, т.е. NP. Сперва активируется правило 2, а затем правило 8 (рис. 2а). Т.к. “planes” не соответствует ”they”, алгоритм срабатывает вновь, и теперь сперва активируется правило 3, а затем правило 9. Затем алгоритм возвращается к правилу 1 и следующей целью ставит определение VP. Сперва активируются правила 5, 6, а затем 12 (рис. 2b). Дальнейший ход разбора отржен на рисунке 2 (с, d, e).

Синтаксический разбор снизу вверх начинается со слов в предложении. Опять же разбор ведется слева направо, и сперва применяется правило с наименьшим номером. Итак, сначала первое слово предложения “they” соотносится с антецедентом правила 9, которое после выполнения выдает утверждение, что “they” является местоимением (PRO). Затем выполняется правило 3 и выдает, что “they” является NP. NP соответствует антецедентам правил 1 и 5, но ни одно из этих правил еще не вызвано, поэтому разбор переходит к “are”. Выполняется правило 11 (несмотря на то, что правило 12 также вызвано, оно не выполняется в соответствии с правилом о последовательности выполнения правил). Затем выполняются правила 10, 8 и 2 (рис. 3а). На данной стадии дальнейший разбор последовательности NP+AUX+ADJ+NP невозможен, поэтому мы возвращаемся к последнему вызванному, но еще не выполненному правилу, т.е. к правилу 4. Разбор последовательности NP+AUX+NP так же невозможен, поэтому опять выполняется последнее вызванное невыполненное правило. Сейчас это правило 13, которое выдает, что “flying” является V. Затем выполняются правила 6 и 5 (рис. 3с). Разбор последователльности NP+AUX+VP невозможен, поэтому выполняется правило 7 и выдает утверждение, что “are flying” является VT. Затем снова выполняются правила 5 и 1, на чем и заканчивается синтаксический разбор (рис. 3d).

Данный пример был приведен с целью сравнения механизмов синтаксического разбора снизу вверх и сверху вниз. Установление строгого порядка разбора слева направо и нумерация правил обусловлены стремлением к применению в наибольшей степени сходного алгоритма, несмотря на то, что результаты разбора оказались различными.

Системы со встроенными правилами. Рассмотрим прямой и обратный цепной анализ на примере выдуманного набора правил о том, как следует провести вечер. Правила расположены в обычном порядке, антецедент располагается слева, а следствие - справа, все вызванные правила выполняются, а разбор ведется параллельно.

1. Хороший фильм по ТВ + Рано утром встреч нет Позднее кино

2. Рано утром встреч нет + Нужно поработать Работа допоздна

3. Нужно поработать + Необходимы документы Работа в офисе

4. Позднее кино Не спать допоздна

5. Работа допоздна Не спать допоздна

6. Работа допоздна Возвращение в офис

7. Работа в офисе Возвращение в офис

Например, правило 1 гласит, что если по ТВ идет хорошее кино и у меня завтра рано утром встреч нет, тогда я следую режиму “Позднее кино”.

Рассмотрим сперва пример прямого цепного анализа. Допустим, система получила начальную информацию о том, что завтра рано уторм у меня нет встреч. Активируются правила 1 и 2. Допустим, что далее система получила сообщение о том, что мне нужно поработать. Активируется правило 3, а правило 2 вызывается и выполняетя, откуда следует вывод, что я нахожусь в режиме “Работа допоздна”, в результате чего вызываются и выполняются правила 5 и 6. В итоге система заключает, что я должна вернуться в офис и не спать допоздна.

Теперь рассмотрим эту же проблему с применением обратного цепного анализа. Допустим, что система получила исходную информацию о том, что у меня нет завтра утром встреч, но мне нужно еще поработать, а затем ее (систему) спросили, следует ли мне вернуться в офис. Данный запрос активирует правила 6 и 7. В свою очередь возникнет вопрос “Работа допоздна” или “Работа в офисе”? При этом активируются правила 2 и 3, и возникает вопрос “Рано утром встреч нет”, “Нужно поработать” или “Нужны документы”? Первые два антецедента будут удовлетворены, таким образом правило 2 будет вызвано и выполнено, что повлечет за собой удовлетворение антецедента “Работа допоздна”, вызов и выполнение правила 6, в результате чего система придет к заключению, что мне следует вернуться в офис.

Обратите внимание на то, что при прямом разборе порождается больше следствий, а при обратном - запросов. Т.к. в обоих примерах использовались одни и те же данные, то в ходе анализа выполнялись одни и те же правила, но активировались различные.

**Сравнение**

Эффективность. Выбор вида анализа (сверху вниз или снизу вверх) зависит от конфигурации дерева, по которому осуществляется поиск. Если в среднем каждому элементу следует большее количество элементов, нежели предшествует, то анализ сверху вниз (или обратный анализ) будет более эффективным и наоборот. Рассмотрим крайний случай. Допустим, что поисковая область образует дерево с вершиной в начальном состоянии. Тогда при использовании прямого подхода нам придется осуществлять поиск практически по всему дереву, тогда как при обратном подходе - только в его линейной части.

Сравнение и унификация. В системах с заложенными правилами или системах логического анализа выбор прямого или обратного цепного анализа влияет на степень трудности процесса сравнения. При прямом цепном анализе системе постоянно предъявляются новые факты, не имеющие свободных переменных. Таким образом постоянно проводится сравнение антецедентов, вполне вероятно обладающих свободными переменными, с фактами, не обладающими таковыми.

С другой стороны, системам с обратным цепным анализом често задают специальные вопросы. Если правила изложены в логике предикатов, а не логике суждений, тогда производится сравнение вопроса с переменной со следствием с переменными. Вторичные запросы также могут содержать переменные, поэтому, в общем, системы с обратным цепным анализом должны быть разработаны таким образом, чтобы они могли сравнивать две символьные структуры, каждая из которых может содержать переменные, для чего потребуется создание алгоритма унификации.

**Смешанные стратегии**

Поиск в двух направлениях. Если не ясно, какой вид поиска - прямой или обратный - является наиболее приемлимым для конкретного приложения, следует осуществлять поиск в двух направлениях. В таком случае, отправными точками становятся начальное и конечное состояние, и поиск осуществляется по направлению к центру.

Вывод по двум направлениям. При данном подходе изначальные данные применяются для активирования правил, котоые перебирают другие антецеденты в обратном порядке. Вторичные запросы, которые не соответствуют ни следствиям, ни данным, сохраняются в качестве “демонов”, которые могут быть удовлетворены позднее за счет новых или позднее поступивших данных. Систему можно разработать таким образом, что данные, удовлетворяющие “демонам” (антецеденты активированных правил) не будут активировать дополнительные правила, что “заставит” систему при предстоящем прямом выводе сконцентрироваться на правилах, учитывающих предыдущий контекст.

Разбор с началом в левом углу. Применив вышеописанный метод к парсингу, мы получим так называемый разбор с началом в левом углу. В терминах примера, приведенного в разделе парсинг, система сначала рассмотрит “they”, найдет правило 9 - единственное правило, которое можно применить к этому слову, затем правило 3, объясняющее PRO, а затем правило 1, как единственное правило, следствие которого начинается с NP. Далее система попытается разобрать сверху вниз “are flying planes” как VP.

**Заключение**

Обычно в системах искусственного интеллекта применяется один из двух видов анализа. Первый - это анализ снизу вверх или прямой анализ, а второй- сверху вниз или обратный. Различие их определяется тем, в каком направлении ведется поиск (от начала в конец или наоборот) и какой элемент (следствие или антецедент) активирует правила.

Фактор эффективности и легкости внедрения может сыграть решающую роль при выборе вида анализа, который будет применяться в определенном приложении, но следует помнить, что использование смешанных стратегий также возможно.