**Мог ли на самом деле Пролог быть первым языком программирования, придуманным человечеством?**

В нашей реальности Пролог был создан намного позднее, чем первые языки программирования. Во времена появления первых вычислительных машин языки программирования создавались с учетом архитектуры ЭВМ тех лет и были рассчитаны на написание программ с явно заданной последовательностью элементарных действий. Создание машин, способных выполнять самые простые операции, было сопряжено с огромными затратами финансов и рабочей силы. Они были сложны в использовании и занимали огромные пространства. Создание ЭВМ, способной выполнять любую программу, написанную на Прологе, не представлялось бы возможным. И даже если бы каким-то чудом такие ЭВМ были бы доступны для производства, они были бы еще менее доступными для использования за счет дороговизны производства, сложности внутреннего устройства и размеров, которые, вероятнее всего, были бы еще больше, чем привычные нам ЭВМ того времени. Такие машины имелись бы только в главных научных центрах самых богатых и развитых стран, а количество людей, способных обслуживать и использовать такие машины, было бы очень мало, и их обучение было бы связано с огромными затратами и сложностями. Возможно, широкое распространение таких ЭВМ в научных лабораториях и университетах произошло бы намного позже или, что еще хуже, не произошло бы никогда. Пролог был бы тем инструментом, которым было бы невозможно эффективно воспользоваться на текущем уровне развития аппаратного обеспечения, который негде применить практически, кроме как для построения какой-то теоретической модели декларативного программирования. Он не решал бы текущих проблем и, что более вероятно, добавлял бы новых. Также сложно представить, как в этом случае были бы устроены операционные системы, каким образом они бы функционировали и выполняли свою роль по управлению ресурсами компьютера и предоставлению интерфейса между компьютером и человеком. Такие ее функции как работа с файловыми системами и структурами, обеспечение системных прерываний, многозадачности и другие важные функции трудно реализуемы или вовсе нереализуемы на Прологе. Появление графического интерфейса также представляется маловероятным, так как неизвестно, как логически описать графику. Пришлось бы использовать более привычные подходы. Пролог был бы непригоден для использования в военных целях, космонавтике или на стратегических объектах из-за возможности зацикливания и малой отказоустойчивости при возникновении непредвиденных ситуаций, которые не описаны в базе фактов Пролога. Проблема в том, что все ситуации предугадать и описать невозможно. Для этих задач понадобился бы привычный нам императивный язык программирования. В конце концов, язык императивного типа лучше подходил для решения текущих задач, стоящих перед человечеством. В связи с вышеперечисленным, появление Пролога в роли первого языка программирования, придуманного человечеством, можно было бы считать чудом. Но как говорил Цицерон « То, что не может произойти, никогда не происходит; то, что может, — не чудо. Следовательно, чуда вовсе не бывает ».

**Что было бы, если бы Пролог был первым языком программирования, придуманным человечеством?**

Если бы человечество придумало Пролог, как первый язык программирования, то возможно это сильно отличалось бы от того, что мы имеем в реальной жизни. Привычные для нас вещи могли бы выглядеть совершенно иначе. Само программирование могло бы пойти по совершенно другому пути, где основной упор делался бы на декларативные и функциональные языки программирования. Устройство ЭВМ и принципы их использования также могли бы быть совершенно другими. Модель компьютера Фон-Неймана не подходит для построения компьютера, с языком программирования Пролог в качестве основного. Вероятно, была бы придумана совершенно другая модель ЭВМ, способная работать напрямую, без использования в качестве посредника между Прологом и машиной другого языка программирования, с языками сверхвысокого уровня, такими как Пролог. Такой компьютер мог бы решить любую задачу, которую можно решить, используя языки императивного или функционального типа. Мне кажется, что такие ЭВМ были бы намного сложнее в производстве и обслуживании, чем существовавшие в реальности, и, как следствие, менее распространенными в научном сообществе. С другой стороны, Пролог позволяет отделить логику от внутренней реализации, и программисту не надо было бы задумываться о внутреннем устройстве такого компьютера для решения задачи. Это намного бы упростило бы и сам процесс программирования, и процесс его обучению. Совершенно другим мог бы быть и перечень основных задач. Скорее всего, в приоритете стояли бы такие задачи, как решение логических задач, проверка гипотез на истинность или ложность на основе известных науке фактов об окружающем мире, разработка автоматических систем перевода с одного языка на другой, составление достаточно точных прогнозов в различных сферах деятельности человека на основе собранных данных и некоторых фактов о взаимодействии различных компонент между собой, их влиянии друг на друга и на состояние системы в целом, символьные вычисления для решения уравнений, дифференцирования и интегрирования. Намного раньше были бы предприняты попытки по созданию некого подобия искусственного интеллекта и естественно-языкового интерфейса для обеспечения простого и понятного взаимодействия человека и машины. Можно было бы общаться с компьютером на понятном для человека языке. Достаточно сформулировать задачу или запрос, и машина сама решит поставленную задачу или выдаст рекомендации, которые могут сильно помочь при решении данной задачи. Это намного упростило бы процесс взаимодействия человека и компьютера и позволило бы увеличить продуктивность в тех областях, где широко применяются компьютеры для решения различных задач. Даже необученный программированию или математической логике человек мог бы воспользоваться таким компьютером, сформулировав запрос и получив ответ в понятном для него виде. Это позволило бы существенно увеличить сферу применения таких компьютеров и избавиться от потребности в специально обученных работниках для использования этих компьютеров. Работы в сфере искусственного интеллекта позволили бы изобрести достаточно умных роботов, способных самостоятельно выполнять различные задачи на производствах вредных или тяжелых для человека, таких как горнодобывающая промышленность или химическая промышленность, для которых нет необходимости досконально описывать каждое действие. Достаточно лишь описать цель, а машина сама выполнит всю работу. Они хорошо подходили бы для физического или иного труда, который сложно выполнить человеку, но, по моему мнению, задачи, в которых требуются логические размышления, были бы им не под силу. Может быть, тогда мы бы встречали такого рода роботов намного чаще и они бы влияли на нашу жизнь намного больше, чем в реальности. С другой стороны, не требующие отдыха, еды и зарплаты роботы могли бы лишить работы огромное количество людей, что вызвало бы требования об ограничении или запрете их применения на производствах. Использование языка Пролог для разработки систем автоматического доказательства теорем могло бы сильно расширить нашу область знаний в различных точных науках, таких как математика, физика и во многих других. Это, несомненно, оказало бы огромное влияние на весь научный мир, и помогло бы ученым легче делать новые открытия и проверять новые гипотезы. Предпринимались бы попытки автоматизировать некоторые мыслительные процессы человека, что крайне сложно или невозможно при использовании императивных языков программирования. Имея достаточно точные прогнозы о том, в каком состоянии система будет находиться через определенный промежуток времени на основе имеющейся на данный момент информации, мы могли бы своевременно узнать о возможных проблемах или неприемлемых результатах нашей деятельности и скорректировать наше поведение для достижения поставленных целей и для избегания нежелательных ситуаций. Это можно использовать при составлении прогноза погоды, для управления ресурсами компании и для множества других различных целей. Таких сюрпризов, как плохая погода во время важного мероприятия на открытом воздухе или банкротство компании в связи с неправильным распределением ресурсом или неверной интерпретацией ценовых сигналов на рынке стало бы на много меньше. Проблемы могли бы появиться при возникновении ситуаций, которые не могут быть предсказаны на основе имеющейся информации, но которые могли бы оказать такое сильное воздействие, которое нельзя игнорировать. В таком случае конечный результат может разительно отличаться от предполагаемого, что может привести к серьезным или даже непоправимым последствиям. Большая проблема заключается в том, что в жизни такие ситуации встречаются довольно часто. Проблемы возникновения непредвиденных ситуаций можно отнести к любым сферам, где использовался бы Пролог в качестве основного языка программирования. Поэтому, по моему мнению, оценкой полученных результатов должен заниматься специально обученный человек, который мог бы учесть возникновение таких ситуаций и вмешаться в случае возникновения непредвиденных ситуаций. Также сложность заключается в построении точной математической модели задачи, что может стать большим препятствием для достижения и реализации всех поставленных перед человечеством целей. Неправильно сформулированная задача и набор правил могут привести к неправильному результату или зацикливанию. Пролог мог бы быть использован для создания некой поисковой системы, которая искала бы информацию в большой общей базе данных, в которой хранились бы все доступные на данный момент человечеству знания, и вела бы диалог с пользователем на естественном для него языке, отвечая на его запросы и выдавая нужную информацию. Доступность информации, касающейся любых или, по крайней мере, значительной части отраслей деятельности человечества, при наличии такого аналога современной поисковой системы значительно возросла бы, что позволило бы существенно повысить эффективность в производстве, исследованиях и обучении. Помимо этого, в этом случае существенно возросла бы эффективность обработки больших объемов данных, увеличились бы возможности различных методов сбора информации. Это помогло бы добиться существенного прогресса и увеличить эффективность в различных сферах деятельности человека, например, в маркетинге, менеджменте или при проведении различных статистических исследований, в которых используются большие объемы данных. Изобретение и внедрение систем автоматического перевода с одного языка на другой могло бы помочь преодолеть языковой барьер, не используя при этом услуги живого переводчика, и существенно ускорить перевод книг, статей и других полезных материалов. Может быть, в будущем были бы изобретены компактные устройства, которые с большой точностью могли бы переводить в режиме реального времени с одного языка на другой, например, во время диалога или совещания. В таком случае значимость профессии переводчика была бы сильно снижена, а востребованность таких систем автоматического перевода увеличена. Преодоление языкового барьера помогло бы ускорить совместные исследования во многих науках. Возможно, такие системы могли бы также переводить программы с одного языка программирования на другой, что существенно повысило бы переносимость программ. Программист мог бы писать программы на своем любимом языке программирования и при возникновении необходимости, используя систему такого рода, перевести без особых трудностей программу на другой язык программирования, не тратя на это свои силы и время.

Изменились бы требования и к людям, которые используют компьютеры и пишут для них программы. Они должны обладать знанием математической логики и дискретной математики. Они должны уметь корректно сформулировать цель решения задачи, сформулировать известные сведения, относящиеся к задаче. Такие люди должны уметь думать на достаточно высоком уровне абстракции, намного большем, чем при использовании языков программирования императивного типа. Математические науки обрели бы еще большую важность и ценились бы еще больше. Потенциально любой хорошо обученный математик мог бы писать программы для компьютеров. Программисту не надо было бы задумываться о внутреннем устройстве компьютера для решения задачи, в отличие от ситуации, когда для решения задачи применяется язык программирования императивного типа. В связи с этим, именно математика расширила бы свое присутствие в учебных программах школ и технических университетов. Возросло бы число математических кружков и специализированных школ. Возможно, порог вхождения в сферу программирования был бы выше, за счет требований к знанию математической логики для написания эффективных программ для решения нетривиальных задач. С другой стороны основам логики и математической логики в частности можно обучать уже в школе. Также Пролог обладает простым и понятным синтаксисом, что позволяет легко понять, что делает программа, просто просмотрев ее код. Ученикам не нужно было бы изучать сложное внутреннее устройство компьютеров для обучения программированию и написания программ. Поэтому Пролог мог бы быть более прост для изучения и восприятия школьникам и студентам, и, как следствие, молодых и талантливых программистов могло бы быть больше. Возможно, намного раньше в школах появились бы предметы, связанные с программированием. Это позволило бы привить учащимся любовь к программированию, популяризовать программирование в обществе, увеличить знания населения о мире компьютеров и программ.

Человечество также могло бы столкнуться с рядом проблем, используя Пролог в качестве своего первого языка программирования. Пролог не получилось бы эффективно использовать в военной сфере, на важных стратегических объектах, где при работе требуется большая степень отказоустойчивости, таких как атомные электростанции, гидроэлектростанции, в космической, отрасли и других не менее важных областях из-за неспособности Пролога справиться с непредвиденными ситуациями, отличающимися от описанных в базе данных Пролога. На такого рода предприятиях отказ системы категорически недопустим и может привести к очень серьезным или даже катастрофическим последствиям. Для этих целей пришлось бы отдельно разрабатывать некий императивный язык программирования и специальную аппаратуру, которая могла бы с ним работать. Из-за того, что Пролог основан на логике предикатов первого порядка, то не любую бы задачу представлялось возможным решить, используя только декларативные методы. Возникла бы необходимость в использовании императивно подобного кода. Для некоторых задач слишком сложно сформулировать решение и набор фактов, используя предикаты первого порядка. Их пришлось бы решать намного дольше или использовать язык программирования императивного или функционального типа. Пролог слабо подходит для сложных математических вычислений, поэтому области науки, зависимые от таких вычислений могли бы развиваться медленнее, чем в реальной жизни. Также Пролог не сильно подходит для непосредственной работы с аппаратурой и ее настройкой. Малая мощность процессоров тех лет и малый объем оперативной памяти в различных компактных устройствах требовали тонкой настройки и порождали необходимость работать напрямую с оборудованием. Код на Прологе получался бы слишком сложным и запутанным для понимания и, как следствие, сложно улучшаемым. В таком коде выше шанс появления ошибки, которую потом может быть очень трудно устранить. Для этого наверняка использовались бы привычные нам языки программирования императивного типа. Из-за потенциальной сложности и дороговизны производства компьютеров, использующих Пролог как основной язык программирования, компьютеры могли бы получить менее широкое распространение, а в распоряжение простых обывателей они могли бы вообще не попасть или попасть намного позже, чем это произошло в нашем мире. Также для реализации графического интерфейса между пользователем и компьютером пришлось бы использовать другие не декларативные средства, так как Пролог плохо подходит или вовсе не подходит для этой задачи. С другой стороны, изобретение и внедрение естественно-языкового интерфейса между компьютером и человеком могло бы частично решить эту проблему, но не полностью. Использование в Прологе поиска решения в глубину и операторы отсечения и ветвления препятствуют автоматическому распараллеливанию программ. Это могло бы отодвинуть время изобретения и широкого использования многопроцессорных систем и существенно сократить перечень сфер, где такие системы могли бы быть использованы. В конце концов, задачи становились бы все сложнее и требовали бы более совершенного и мощного оборудования для решения. Многопроцессорные системы стали бы необходимостью для дальнейшего развития, и Пролог мог бы быть заменен функциональными или императивными языками программирования, которые позволяют существенно легче решить проблему распараллеливания задач. При этом Пролог мог бы сохранить свои позиции в других сферах. Использование Пролога для реализации интернет технологий может быть сопряжено с трудностями из-за наличия таких ситуаций, как нестабильное подключение к сети, рассинхронизация клиента и сервера, потеря данных при передаче. Пришлось бы или использовать другой язык, или создавать специальные версии Пролога и дополнения к нему для работы с интернетом.

**Выводы.**

В заключение хочется сказать, что мир, в котором Пролог стал бы первым языком программирования, придуманный человечеством, мог бы сильно отличаться от того мира, в котором мы все живем и в котором главенствующая роль отводится императивным языкам программирования и их различным Фреймворкам. Многие отрасли человеческой деятельности, такие как наука, образование, маркетинг, менеджмент и многие другие могли бы быть на совершенно другом, более высоком уровне развития, чем в нашем мире. Возможно, было бы проверено и доказано больше научных гипотез и было бы сделано больше научных открытий, мы бы обладали более точными прогнозами и могли бы вовремя скорректировать наши действия или пересмотреть наши планы и, как следствие, избежать множества проблем. Искусственный интеллект намного больше влиял бы на нашу жизнь и сильнее задействовался в науке, обучении, производстве, в наших личных гаджетах. Он также учувствовал бы в нашей творческой деятельности, помогая в придумывании и изобретении новых вещей. Разработка естественно-языкового интерфейса, вероятно, позволило бы нам общаться с таким искусственным интеллектом практически так же, как и с живым человеком. Может быть, что многие профессии могли бы исчезнуть или потерять свою значимость, например переводчик с одного языка на другой, благодаря автоматическим системам перевода или профессии разнорабочих, носильщиков, фасовщиков, горняков и многие другие, связанные с физическим или вредным для здоровья трудом, за счет применения роботов для этих целей. Другие профессии, особенно связанные с математикой и математической логикой в частности, ценились и оплачивались бы в развитых странах мира намного больше. Языки программирования могли бы пойти по совершенно другому пути развития, и императивные языки не занимали бы лидирующей роли в сфере написания различных программ. Могли бы появиться современные более продвинутые версии Пролога или другие языки логического программирования, которые широко использовались бы в сфере написания программ. Вероятно, некоторые отрасли могли развиваться медленнее, чем это происходило в реальности. Пролог не может использоваться для решения абсолютно всех задач, для некоторых он или подходит плохо, либо не подходит совсем. В связи с этим, для решения целого ряда проблем пришлось бы отказаться от идеи логического программирования и использовать более привычные нам императивные языки программирования. Для создания графических интерфейсов, бесперебойного обмена данными в интернете, для использования в военной, космической, ядерной и других отраслях, где решающее значение играет надежность и устойчивость к непредвиденным ситуациям, больше подходит язык программирования императивного типа. Сами компьютеры могли бы быть не такими дешевыми и не такими распространенными, какими они являются сейчас. Может быть, они не были бы и настолько компактными, использовали бы совершенно другое по устройству аппаратное обеспечение и были бы недоступны большей части простых обывателей. Возможно, для домашнего пользования и обучения в школах и институтах были бы изобретены компьютеры, работающие и устроенные более привычным для нас образом и более дешевые и простые в производстве и обслуживании. В конечном итоге роль Пролога рано или поздно снизилась бы, а роль других языков программирования увеличилась бы. В любом случае, мы можем только предполагать. Были объективные причины, почему Пролог не стал и не мог стать первым языком программирования, придуманным человечеством. К сожалению, в нашем мире Пролог нельзя назвать очень популярным языком программирования, хотя он и используется для решения некоторого рода задач в программировании. Возможно, он был бы более популярным языком, чем сейчас, сложись история несколько иначе, но он все равно, наверняка бы, сильно проигрывал бы по популярности известным нам императивным языкам программирования и никогда не смог бы полностью или по большей части их заменить. Может быть так, что при возросшем количестве исследований в области искусственного интеллекта, Пролог и похожие на него логические языки смогут увеличить свою популярность и свое влияние в сфере программирования в будущем, но точно сказать не представляется возможным. В связи с этим нельзя с уверенностью сказать, что было бы, если бы Пролог был первым языком программирования, придуманный человечеством, как развивался бы тогда наш мир, и как бы развивалось программирование и языки программирования в частности. К сожалению или к счастью, невозможно повернуть время вспять и изменить события прошлого. Мы живем в мире, где главенствовали и главенствуют сейчас императивные языки программирования, и это вряд ли удастся изменить в ближайшем будущем. Такое состояние дел обусловлено историческим развитием сферы программирования. Можно сказать, что Пролог появился тогда, когда встала необходимость в подобного рода языке программирования. В любом случае, всегда интересно подумать об ином варианте развитии событий, о тех возможностях, которые мы бы тогда имели, о том мире, в котором бы жили и сравнить с тем, что мы имеем сейчас в нашем мире.