

ЭФФЕКТИВНОЕ ИСПОЛЬЗОВАНИЕ ВИЗУАЛИЗАЦИИ В ПРОЦЕССЕ ОБУЧЕНИЯ

О роли визуального представления информации в процессе обучения написано достаточно много работ. С приходом компьютерных технологий появились новые, интерактивные возможности для визуального представления знаний.

В первой части статьи будет представлена работа двух преподавателей и ученых Джона Бэрвайза (Jon Barwise) и Джона Этчименди (John Etchemendy) «Компьютеры, визуализация и природа мышления» (Computers, visualization and the nature of reasoning), в которой авторы не только делятся опытом использования визуальных средств в процессе обучения, но еще делают важные теоретические выводы об эффективности и познавательной силе таких представлений.

Во второй части статьи будет рассмотрена более современная работа «Эффективные черты визуализации алгоритма» («Effective Features of Algorithm Visualizations» Purvi Saraiya, Clifford A. Shaffer, D. Scott McCrickard and Chris North), где авторы представляют практические результаты эффективности использования различных возможностей визуализации.

КОМПЬЮТЕРЫ, ВИЗУАЛИЗАЦИЯ И ПРИРОДА МЫШЛЕНИЯ

Джон Бэрвайз и Джон Этчименди – известные ученые в области логики, мышления и образования. В своей работе авторы поделились опытом обучения элемен-

тарной логики в Стенфордском университете, начиная с 1983 года. К тому времени Стенфордский университет уже имел определенные традиции преподавания логики, которые восходили еще к 60-м годам XX столетия. Тогда для преподавания логики использовалась программа Valid. Однако, наблюдая за работой студентов с этой программой, авторы заметили, что, хотя она и повышает эффективность изучения логики, некоторые аспекты теории остаются для студентов трудными для понимания.

Тогда у авторов появилось желание создать программное обеспечение, которое бы помогло студентам развить способность наглядно представлять абстрактные логические понятия. Следующие 4 года работы с различными группами студентов привели к созданию таких программ, как Мир Тьюринга и Мир Тарского.

МИР ТЬЮРИНГА

Изобретенная Аланом Тьюрингом в 1936, машина Тьюринга является ключевым понятием в современной вычислительной теории, определяющей, что компьютер может делать, а чего – нет. Машина Тьюринга представляет собой бесконечную в обе стороны ленту, разделенную на ячейки, и управляющее устройство, способное записывать или читать символы на ленте, а также двигаться по ленте влево или вправо. В каждую ячейку можно записать не больше одного символа. В каждый конк-

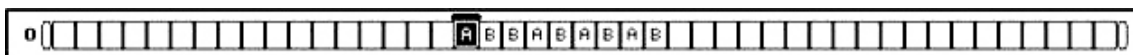


Рисунок 1.

ретный момент управляющее устройство может считать/записать символ в ячейку, над которой оно расположено, и продолжить свой путь влево или вправо.

В Мире Тьюринга бесконечная лента представлена узким окошком, расположенным внизу экрана (см. рисунок 1).

Машина Тьюринга имеет конечное число состояний. Связанные с каждым состоянием инструкции определяют, какое действие должна совершить машина Тьюринга и в какое состояние она должна перейти по завершению этих действий. Состояния машины Тьюринга описываются с помощью диаграмм.

Например, на рисунке 2 изображена диаграмма, соответствующая машине Тьюринга только с двумя состояниями:

Лингвистическое описание такой машины будет следующим.

В состоянии 0: если мы видим A – движемся направо и переходим в состояние 0.

В состоянии 0: если мы видим B – движемся направо и переходим в состояние 1.

В состоянии 1: если мы видим B – движемся направо и переходим в состояние 1.

Такая машина может просматривать ленту с записанными в ее ячейки символами A и B, но она остановится, как только увидит символ A сразу после B, так как в случае если машина увидела B она переходит в состояние 1, а для состояния 1 не описано инструкций, если машина увидит символ A.

В Мире Тьюринга можно создавать такие диаграммы с помощью набора графических средств. Когда машина запускается, следить за изменениями можно как по ленте, так и по диаграмме. По ленте головка будет путешествовать, делая заданные изменения, а на диаграмме будут подсвечиваться узлы и дуги, показывая изменения в вычислительном процессе.

Таким образом, возможность создавать машины Тьюринга простым рисованием

диаграмм состояний вкупе с возможностью прогонять их и видеть все происходящие изменения позволяет сделать процесс обучения более увлекательным и наглядным, а следовательно, более продуктивным.

МИР ТАРСКОГО

Целью программного обеспечения Мир Тарского – научить студентов символическому языку на основе современной логики первого порядка. Программа позволяет создавать простые трехмерные миры, населяя их геометрическими объектами различных форм (тетраэдры, кубы и додекаэдры) и различных размеров (маленькие, средние, большие), а затем проверять, верны ли в этом мире определенные высказывания первого порядка.

На рисунке 3 приведен пример одного из таких миров, а также примеры высказываний, которые проверяли студенты для этого мира.

Как показано на рисунке, три из пяти приведенных высказываний (11, 13, 14) – верны, остальные два (12, 15) – нет. Например, 12 высказывание неверно, т.к. утверждает, что в созданном мире существует только один тетраэдр среднего размера, в то время как мы видим, что их там – три.

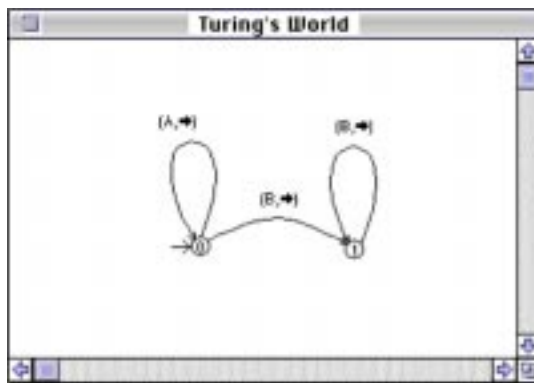


Рисунок 2.

Несмотря на кажущуюся простоту, авторы придумали разнообразные способы применения этой программы. Например, авторы предлагали студентам оценить определенные высказывания в данном мире. Кроме того, студентам предлагалось построить мир, удовлетворяющий определенным высказываниям. Или еще, исходя из данного мира, студентам предлагалось вывести некоторые высказывания, удовлетворяющие этому миру пользуясь языком высказываний первого порядка.

ВИЗУАЛИЗАЦИЯ И МЫШЛЕНИЕ (/ВЫВОД)

Наблюдая за работой студентов с программой Мир Тьюринга, авторы были поражены познавательной силой графического интерфейса. Несмотря на то, что в программном продукте были доступны как графическое представление машины с возможностью ее запуска, так и лингвистическое и формальное описание, студенты в первую очередь отдавали предпочтение именно графическим возможностям, а к

лингвистическим описаниям обращались только тогда, когда их просили об этом.

Опыт работы с программой Мир Тарского только еще больше убедил авторов в правоте своих выводов.

Приведем простейший пример.

Возьмем мир, изображенный на рисунке 4.

Мир представляет собой набор фигур расположенных в пространстве на шахматной доске. Фигуры имеют формы тетраэдра (пирамида), куба и додекаэдра и делятся на три размера – большие, средние маленькие.

Нашей целью будет выяснить как можно больше о двух фигурах *b* и *c*. Может быть, нам даже удастся отметить их на рисунке.

Предположим, что изначально мы получили следующую информацию:

«У фигуры *b* нет соседей, а фигура *c* – не куб».

Исходя из такой информации мы не можем однозначно определить ни фигуру *b*, ни фигуру *c*. Однако мы можем с уве-

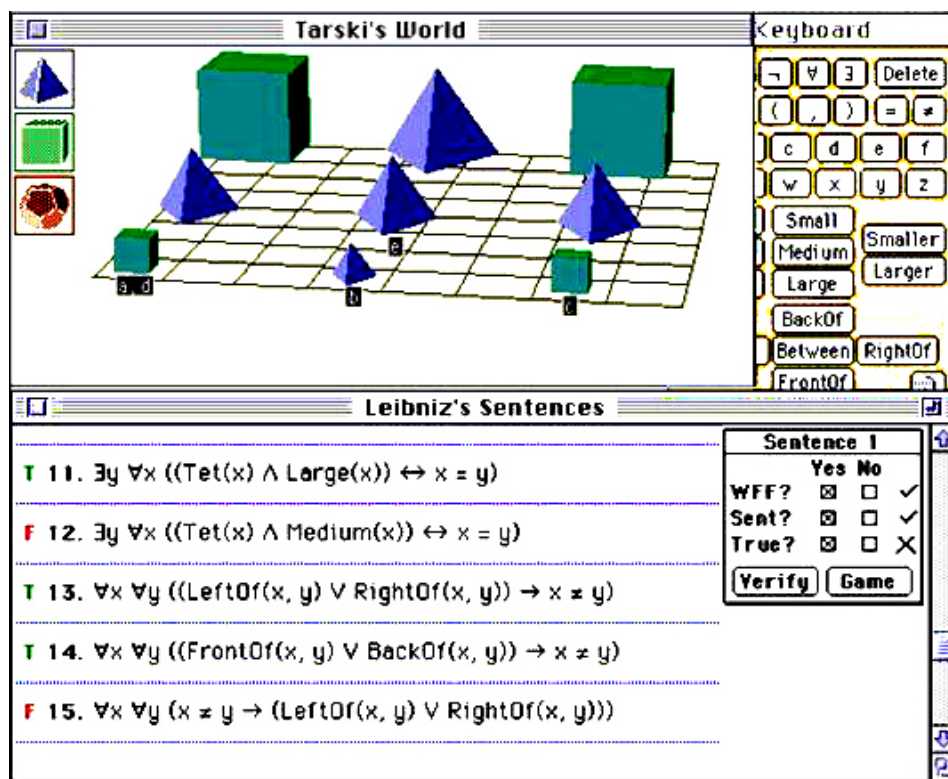


Рисунок 3.

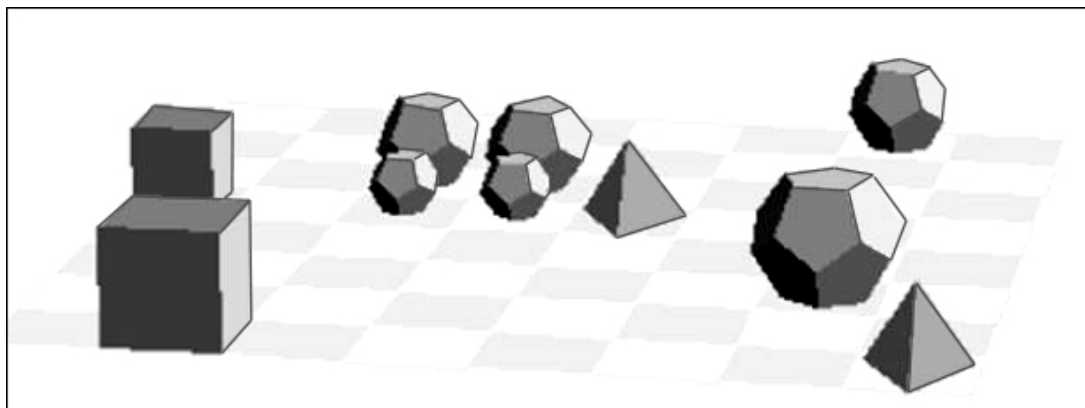


Рисунок 4.

ренностью сказать, что фигура c не может быть ни одним из двух кубов, а фигура b не принадлежит центральной группе, так как каждая из них имеет соседа.

Тогда добавим следующее утверждение:

«Фигура c – либо маленькая, либо большая» (не средней величины).

Мы, конечно, не узнаем ничего нового о фигуре b , однако, в значительной степени сужаем область возможных претендентов на фигуру c . Фигура c может быть либо одним из двух маленьких додекаэдров в центре, либо большим додекаэдром. Таким образом, мы точно знаем, что c – додекаэдр.

Продолжим. Добавим следующую информацию:

«Фигура b расположена дальше, чем оба тетраэдра».

Исходя из этой информации, мы можем сказать, что фигура b может быть одним из трех додекаэдров на заднем плане. Однако еще раньше мы выяснили, что фигура b не расположена в центральной группе. То есть мы можем однозначно определить, что фигура b – додекаэдр среднего размера, расположенный на заднем плане.

Рассмотрим последнюю подсказку, которую мы получаем:

«Фигура b имеет больший размер, чем фигура c ».

Мы уже знаем, что фигура c – либо один из маленьких додекаэдров в центре, либо большой додекаэдр справа. Так как фигура b имеет средний размер, то мы

можем сделать вывод, что фигура c имеет маленький размер. То есть фигура c – один из маленьких додекаэдров в центре. Но мы не знаем, какой именно! Поэтому, в конце концов, у нас есть два варианта, изображенных на рисунке 5.

НЕОДНОРОДНОЕ МЫШЛЕНИЕ

С педагогической точки зрения, подобные упражнения очень эффективны как для изучения логики первого порядка, так и для получения навыков логического мышления. Однако была и обратная сторона медали. Стиль мышления, которым пользовались студенты при решении поставленных задач, отличался от стандартной модели дедуктивных размышлений. Другими словами классическая теория вывода, для обучения которой и были придуманы эти программы, казалось, не соответствует той логике, которую студенты уже использовали для выполнения упражнений.

Авторы утверждают, что в большинстве случаев, когда студенты выполняют упражнения, проявляется так называемая неоднородная логика. Студенты естественным образом комбинируют информацию из двух видов ее представления: диаграмм «мира Тарского» и высказываний первого порядка. Традиционная теория вывода кажется неудобной в данном случае из-за возможного несоответствия между традиционной теорией логики и эмпирически наблюдаемой разнород-

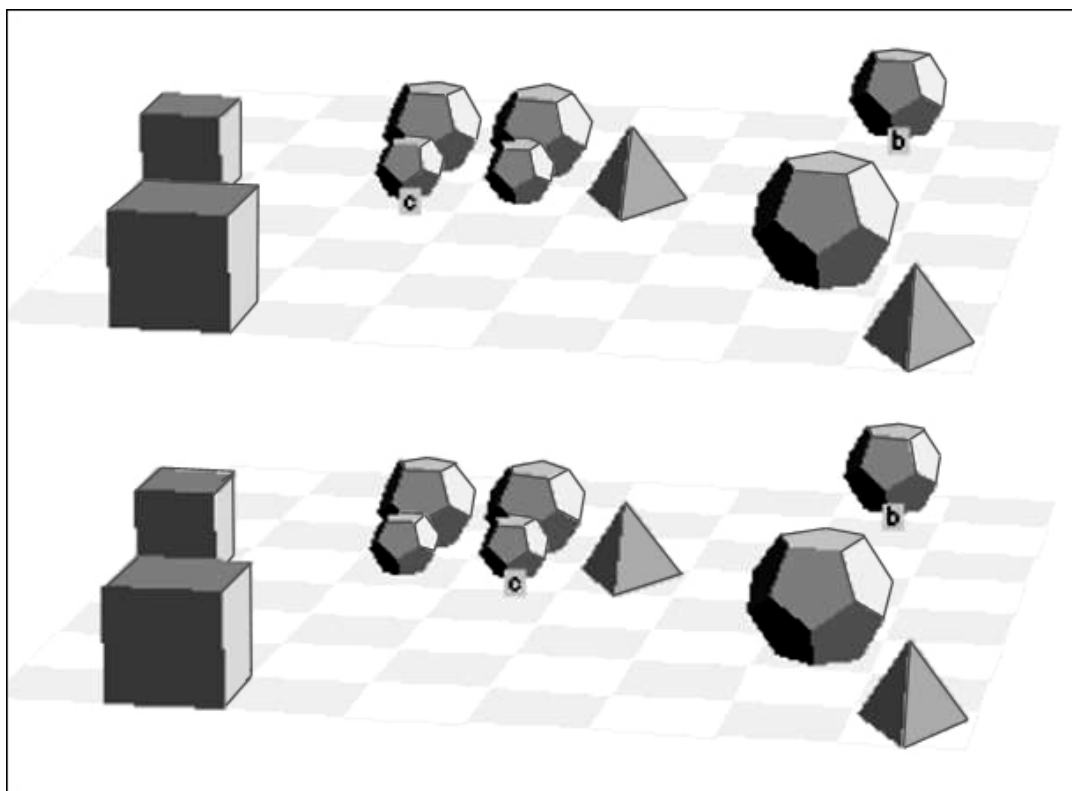


Рисунок 5.

ной логикой. Практическое решение этой проблемы – последнее авторское программное обеспечение – «Nureproof», которое позволяет объединить визуальную и пропозициональную информацию для решения многих типов задач вывода.

Авторы подчеркивают силу диаграмм и визуальных представлений для мышления. В своей работе они намекают на новую теорию вывода и ссылаются на то, что логика должна быть способна точно справляться с разнообразием представлений. В этой работе авторы утверждают, что традиционное понимание логики и вывода не в полной степени соответствует тому, как мы думаем и делаем выводы на самом деле. Таким образом, нам необходимо расширить и обогатить традиционные понятия синтаксиса, семантики, логического вывода и доказательства, чтобы удовлетворять новым (визуальным) формам представления.

Эта работа иллюстрирует, каким образом авторы пришли к осознанию и

оценке силы графических интерфейсов. Компьютеры со своими утонченными графическими возможностями являются важным инструментом для поддержки такого подхода. В этой статье теория вывода понимается как наиболее фундаментальное понятие во всей науке.

ЭФФЕКТИВНЫЕ ЧЕРТЫ ВИЗУАЛИЗАЦИИ АЛГОРИТМА

Изучение алгоритмов и структур данных является важной темой в процессе обучения. Сейчас все чаще преподаватели используют методы визуализации или анимации для лучшего усвоения студентами этого материала. Конечно, студент может освоить алгоритмы и без визуализации, однако, опыт многих преподавателей доказывает, что понимание алгоритмов улучшается, если их объяснение сопровождается наглядными изображениями или демонстрациями. В сети Интернет можно найти массу примеров визуализации

ции алгоритмов. Правда, не все из них работают с одинаковой эффективностью. Поэтому у авторов возникло желание выявить ключевые черты визуализации, которые способствуют улучшению понимания алгоритмов.

В своей работе авторы представили результаты двух экспериментов с визуализацией пирамидальной сортировки (сортировки методом «кучи»). Выбор именно этого алгоритма авторы объяснили тем, что это достаточно распространенный вид сортировки, различные визуализации ее можно без труда найти в сети Интернет, и для понимания она является умеренно сложной.

В соответствии с этим, чтобы выявить основные педагогически удачные черты визуализации сортировки авторами на основе экспертного изучения был составлен следующий список гипотетически возможных черт, улучшающих усвоение алгоритма:

- возможность вести свой собственный пример – таким образом, пользователь может глубже понять алгоритм сортировки;

- хороший предустановленный пример – такой пример может показать студенту лучший или худший результат исполнения алгоритма;

- отображение символического кода (псевдокода) – возможность отслеживать, какая именно часть алгоритма (то есть какие именно инструкции) исполняется в данный момент;

- кнопки «вперед»/«назад» для пошагового исполнения алгоритма и для возврата на определенное число операций назад – возможность контролировать процесс исполнения;

- руководство пользователя, представляющее собой серию вопросов для вовлечения пользователя в процесс сортировки.

Для экспериментов авторы создали свое собственное визуальное представление пирамидальной сортировки, общий вид которой со всеми внедренными возможностями из списка можно видеть на рисунке 6.

В рамках первого эксперимента авторы выработали пять вариантов програм-

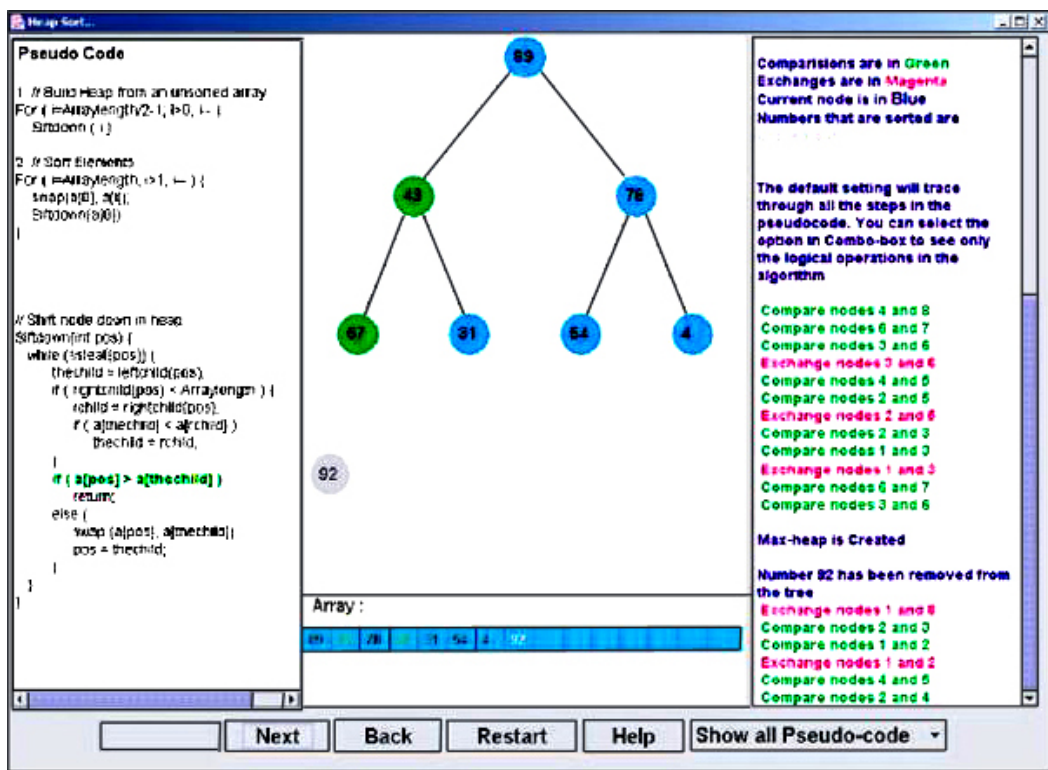


Рисунок 6.

Таблица 1.

Версия	Кнопка «Next» («вперед»)	Кнопка «Back» («назад»)	Предустановленный пример	Пример вводимый пользователем	Псевдокод	Руководство пользователя
1.1	+		+			
1.2	+	+		+		
1.3	+	+		+	+	
1.4	+	+		+		+
1.5	+	+		+	+	+

мы, каждая из которых содержала только некоторые из перечисленных выше возможностей. Список составленных версий можно увидеть в таблице 1.

В эксперименте приняли участие порядка 60 студентов, располагающих базовыми знаниями представления данных и навыками работы с простыми сортировками, однако еще не знакомые с понятиями деревьев и алгоритмом предложенной сортировки.

По итогам первого эксперимента авторы сделали вывод, что использование большого количества вспомогательных возможностей ухудшает, хотя и незначительно, усвоение алгоритма.

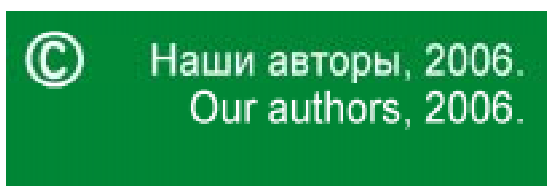
Таким образом, для второго эксперимента авторы сконцентрировались только на трех возможностях: пошаговое выполнение алгоритма, предустановленный программой пример и отображение символического кода. Возможность пошагового выполнения алгоритма и удачный предустановленный пример показали значительно лучшие результаты понимания студентами алгоритма. Отображаемый символичный

код не предоставил никаких значимых результатов.

В результате пошаговый контроль над алгоритмом и удачный предустановленный пример увеличили педагогическую значимость визуализации алгоритма. Но ни руководство пользователя, ни отображаемый символичный код не принесли ощутимых результатов. Однако время, которое студенты тратят на изучение алгоритма, зависит от визуальных особенностей средства. Студенты, использующие символичный код, тратили вдвое больше времени, чем остальные, и при этом показывали не лучшие результаты. Это позволяет сделать вывод, что дополнительные возможности могут снижать не только эффективность, но и оперативность обучения.

Таким образом, авторы показали, что некоторые возможности визуализации алгоритмов способствуют эффективности обучения больше, чем другие, и что предоставление слишком большого количества дополнительных возможностей может пагубно влиять на эффективность.

Исследование выполнено при финансовой поддержке РГНФ в рамках научно-исследовательского проекта РГНФ («Интерактивные формы представления информации в развитии научно-популярного жанра как способ формирования научного мировоззрения у молодёжи (от школьной медиатеки к виртуальному музею занимательной науки)», 08-06-00446а).



*Пухов Алексей Фёдорович,
студент 5 курса математико-механического факультета СПбГУ.*