

АББАКУМОВ А. А., ПЫРЯКИНА К. А., БАЙНЕВ В. В.
АЛГОРИТМ СОСТАВЛЕНИЯ РАСПИСАНИЯ ЗАНЯТИЙ
ДЛЯ ВЫСШИХ УЧЕБНЫХ ЗАВЕДЕНИЙ

Аннотация. Рассматривается алгоритм составления расписания занятий, который на выходе выдает близкое к оптимальному решение. Данное решение может быть использовано для дальнейшего корректирования расписания вручную.

Ключевые слова: расписание, Visual Prolog, база фактов, предикаты.

ABBAKUMOV A. A., PYRYAKINA K. A., BAYNEV V. V.
ALGORITHM FOR SCHEDULING CLASSES FOR HIGHER EDUCATION

Abstract. The article presents an algorithm for scheduling classes considering near-optimal solution outlet. The algorithm can be used for schedule manual adjustment.

Keywords: schedule, Visual Prolog, facts, predicates.

Как известно, процесс составления расписания является наиболее важным в деятельности вуза в целом и каждого факультета в частности. Хорошо разработанное расписание может стать основой для реализации научно-педагогического потенциала учебных курсов и для лучшей усвояемости материала студентами. Следует заметить, что данная тема не является новой, но оптимального решения для автоматизации процесса составления расписания до сих пор не найдено. Основной целью исследования является создание принципиально нового алгоритма составления расписания занятий на объектно-ориентированном расширении языка логического программирования PDC Prolog Visual Prolog.

В процессе реализации проекта были поставлены следующие основные задачи:

- анализ входных данных и формирование базы фактов;
- разработка алгоритма составления расписания в разрезе групп и преподавателей;
- создание приятного и простого пользовательского интерфейса.

Задача по составлению расписания занятий не так проста, какой кажется на первый взгляд. Как известно, было сделано огромное количество попыток ее решения. Некоторые программные продукты вышли в свет, например, Ректор-ВУЗ, БИТ.ВУЗ.Расписание, 1С: Автоматизированное составление расписаний. Университет, Расписание Про v2.3 и т.д. К сожалению, ни одна из этих программ не обладает необходимой универсальностью и не удовлетворяет потребностей всех вузов, либо имеет слишком высокую стоимость. Большинство уже существующих решений для составления расписания используют большое

количество входной информации и хранят ее в виде базы данных. Затем с помощью сложных алгоритмов, разработка которых потребовала не малых сил, входная информация анализируется и составляется расписание, которое не всегда является идеальным. Затрачиваемые ресурсы, время и деньги могут быть весьма внушительными.

Из всего вышесказанного, можно сделать вывод о том, что для решения данной задачи необходим принципиально новый подход, который поможет составить основу для расписания занятий и в дальнейшем доработать ее с учетом человеческого фактора. Исходя из требований к проекту и специфики предметной области, было принято решение разработать систему на Visual Prolog. Выбор именно этого языка для разработки обусловлен тем, что он отлично подходит для реализации данной проблемы [1].

Для каждого языка существует свой класс задач, для которых он подходит лучше других языков программирования. Соответственно, для решения любой задачи есть оптимальный язык (языки). Многие из них, хорошо решаемые императивными языками, например, Pascal и C, плохо решаются на Prolog, и наоборот. Одной из основных областей применения языка Prolog является полуавтоматическое составление расписаний, что весьма подходит для нашей проблемы [2].

Процесс создания расписания с помощью разрабатываемого алгоритма должен начинаться с формирования из входной информации базы фактов. В языке Prolog ключевое слово «facts» (это синоним устаревшего слова «database») определяет начало объявления раздела facts. Раздел facts состоит из последовательности объявлений предикатов, описывающих соответствующую внутреннюю базу фактов. В данном проекте существуют три базы фактов, описывающих входную информацию, результирующее расписание и отчет о процессе генерации расписания [3].

Перейдем к принципу работы алгоритма составления расписания. В системе существует база фактов cells, которая хранит информацию о составленном расписании. Для каждого элемента этой базы заполняются такие поля, как время в формате (Неделя, День, Урок), аудитория, группа, преподаватель, предмет, час и подгруппа. Процесс заполнения базы представлен несколькими циклами. Ниже обозначены основные шаги алгоритма.

Для каждого элемента времени (Неделя, День, Урок) выполняем следующие действия:

1. Перебираем все имеющиеся группы и проверяем наличие ограничений, т.е. невозможность проведения занятий для группы по данному элементу времени. Если ограничений нет, следуем дальше;

2. Перебираем все имеющиеся предметы, проверяем, есть ли конкретный предмет у этой группы, проверяем ограничения по предмету, если предмет у группы есть, а ограничений нет, то следуем дальше;

3. Для каждого преподавателя проверяем, ведет ли он этот предмет, и нет ли у него ограничений, если предмет ведет он, а ограничений на проведение занятий нет, то следуем дальше;

4. Для каждой аудитории считываем ее идентификатор, проверяем, есть ли ограничения, если ограничений нет, а предмет проводится в этой аудитории или ID=0, т.е. можно проводить занятие в любом свободном кабинете, следуем дальше;

5. Проверяем соответствие вместимости аудитории и количества занимающихся, если все студенты и преподаватель уместаются, следуем дальше;

6. Считываем количество подгрупп и часов занятий для каждой из них и выполняем следующие действия:

6.1. Для каждого из часов смотрим, не проставлено ли уже расписание по времени в факте cells для данного преподавателя, аудитории, предмета и часа, если не по одному из показателей не проставлено, следуем дальше;

6.2. Добавляем новый факт в cells;

6.3. Считываем информацию о проведении нескольких занятий подряд, если true, то следуем дальше;

6.4. Проверяем, является ли урок последним в дне, если нет, то устанавливаем значение времени (Неделя, День, Урок+1), проверяем, нет ли по этому времени ограничений для данных аудитории, предмета, группы и преподавателя, если нет, то следуем дальше;

6.5. Возвращаемся к пункту 6, но вместо первоначального времени передаем только что установленное.

После выполнения всех итераций цикла, все факты cells заполнены и можно будет сформировать по ним расписание в виде приемлемом для пользователя.

В заключение стоит отметить, что разработанный алгоритм учитывает основные ограничения и в результате выдает близкое к оптимальному решение, которое может быть использовано в дальнейшем для ручной доработки. Конечно, все нюансы работы вуза алгоритм использовать не в состоянии, но иногда лучше затратить меньшие ресурсы и получить приближенный результат, чем задавать колоссальный объем входной информации и потратить большое количество времени и денег.

ЛИТЕРАТУРА

1. Коста Э. Visual Prolog 7.1 для начинающих. – М., ФЛИНТА, 2008. – 210 с.
2. Гаврилова И. В., Масленникова О. Е. Основы искусственного интеллекта. — М.: ФЛИНТА, 2013. – 282 с.
3. Внутренняя база фактов [Электронный ресурс]. – Режим доступа: http://asu.bgu.mogilev.by/кафедра/Учебнометодические%20материалы/Дисциплины%20кафедры/Экспертные%20системы/Prolog_5/lr07%20%20Внутренняя%20база%20фактов.pdf.