



Сошников Дмитрий Валерьевич

к.ф.-м.н., доцент

dmitryso@microsoft.com

Логическое программирование



Факультет Прикладной математики и физики

Кафедра Вычислительной математики и программирования

Московский авиационный институт (государственный технический университет)

Что будем изучать



- Принципы логического программирования
- Математическая теория в основе логического программирования – логика предикатов, логический вывод, вывод типов
- Семантика языков логического программирования, вопросы реализации
- Языки логического программирования:
 - Prolog
 - Mercury

Что понадобится?

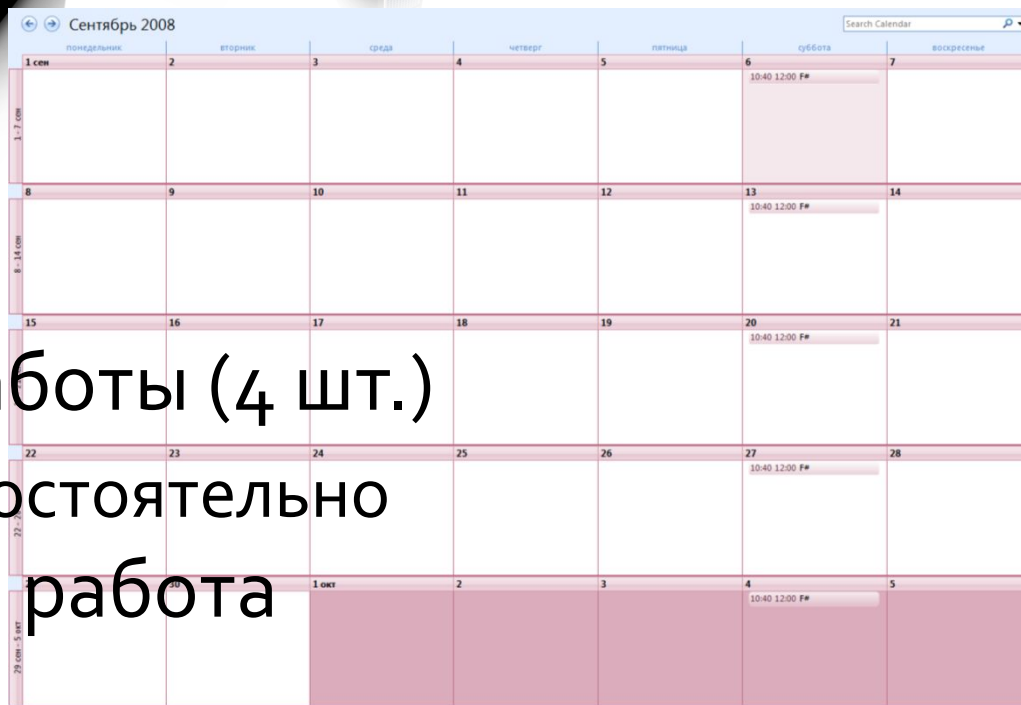


- Любая СП Prolog, поддерживающая «классический» Эдинбургский синтаксис:
 - GNU Prolog (<http://www.gprolog.org>)
- Система на базе .NET
 - P#
 - Prolog.NET (<http://prolog.hodroj.net>)
- Strawberry Prolog (<http://www.dobrev.com>)

Что нас ждет?



- Лекции
- Семинары
- Лабораторные работы (4 шт.)
 - выполняются самостоятельно
- Самостоятельная работа
 - Доклады
 - Обсуждения
- Экзамен



Критерии оценки



- Экзамен (письменный, 5 вопросов) – 80%
- Лабораторные работы – 20%
- Самостоятельная работа (доклады, выступления на семинарах, вопросы, дополнительная работа) – 20%

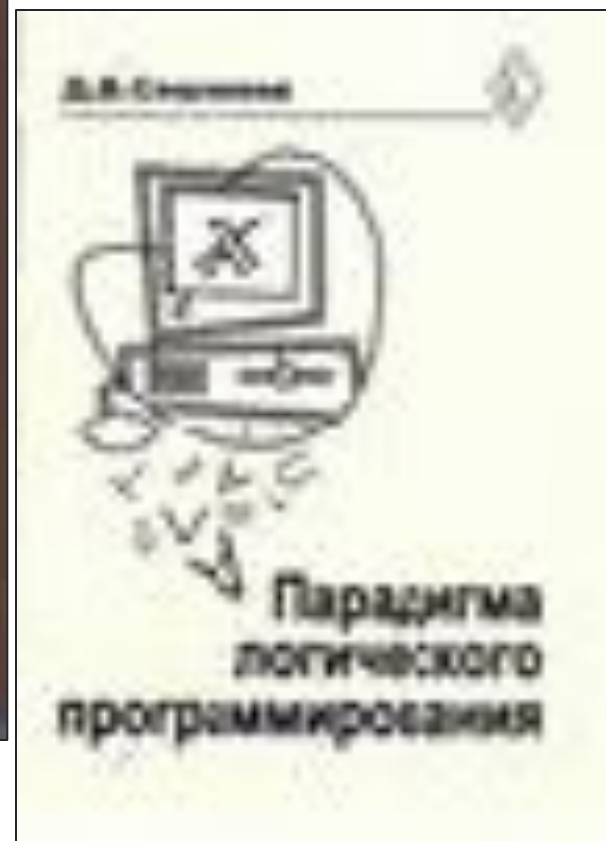


Варианты самостоятельной работы



- Научно-исследовательская работа
- Выполнение полу-исследовательского проекта
- Выступление с докладом (15-20 мин.)

Источники



Источники



- **Сошников Д.В., Парадигма логического программирования**
- **Братко И.** Программирование на языке Пролог для искусственного интеллекта. пер. с англ. – М.: Мир, 1990.
- Bratko I. Programming in Prolog for Artificial Intelligence (3rd edition), Addison-Wesley Publishers, 2001.
- **Клоксин У., Меллиш К.** Программирование на языке Пролог. – М.: Мир, 1987.
- Хоггер К. Введение в логическое программирование: Пер. с англ. -М.: Мир, 1988.
- Набебин А.А. Логика и Пролог в дискретной математике. – М.: Изд-во МЭИ, 1996.
- Малпас Дж. Реляционный язык Пролог и его применение: Пер. с англ. -М.: Наука, 1990.
- Стерлинг Х., Шапиро Э. Искусство программирования на языке Пролог: Пер. с англ. - М.: Мир, 1990.

Введение в Пролог и логическое программирование

Рассмотрим пример



```
speciality(X,tech_translator) :-
studied_languages(X),studied_technical(X).
speciality(X,programmer) :-
    studied(X,mathematics),studied(X, compscience).
speciality(X,lit_translator) :-
studied_languages(X),studied(X,literature).

studied_technical(X) :- studied(X,mathematics).
studied_technical(X) :- studied(X,compscience).
studied_languages(X) :- studied(X,english).
studied_languages(X) :- studied(X,german).

studied(petya,mathematics).    studied(vasya,german).
studied(petya,compscience).    studied(vasya,literature).
studied(petya,english).
```

Правила

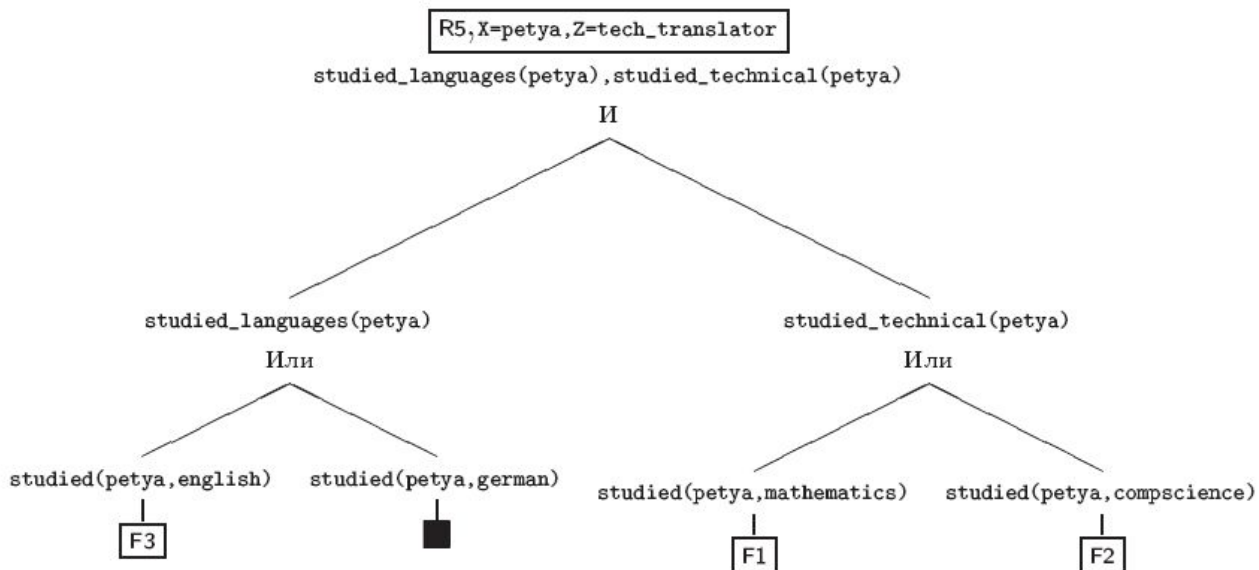
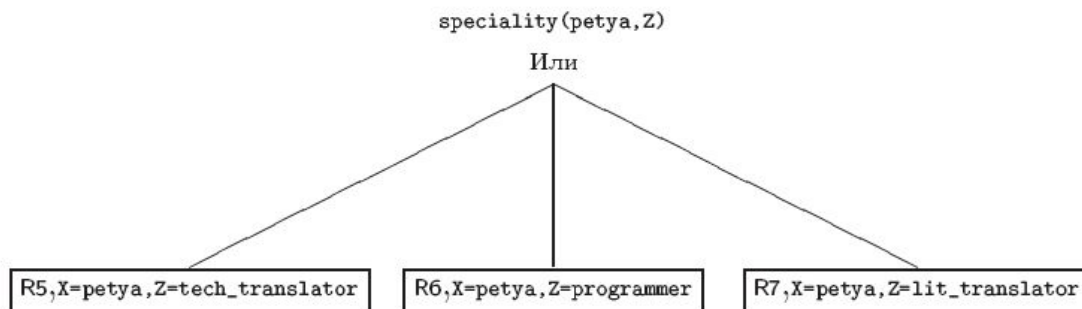
Факты

Как это выглядит?



```
GNU Prolog console
File Edit Terminal Help
GNU Prolog 1.3.1
By Daniel Diaz
Copyright (C) 1999-2009 Daniel Diaz
| ?- consult(spec).
compiling C:/Users/ ит /Desktop/Text/LP/Examples/Lect2/spec.pl for byte code...
C:/Users/ ит /Desktop/Text/LP/Examples/Lect2/spec.pl compiled, 15 lines read - 2320 by
yes
| ?- speciality(petya,X).
X = tech_translator ? ;
X = tech_translator ? ;
X = programmer ? ;
no
| ?- speciality(X,tech_translator).
X = petya ?
yes
| ?-
```


Дерево И-ИЛИ

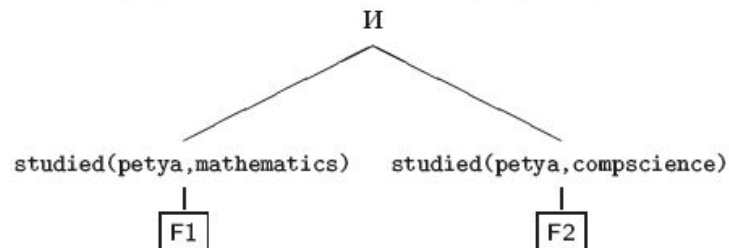


Дерево И-ИЛИ (продолжение)



R6, X=petya, Z=programmer

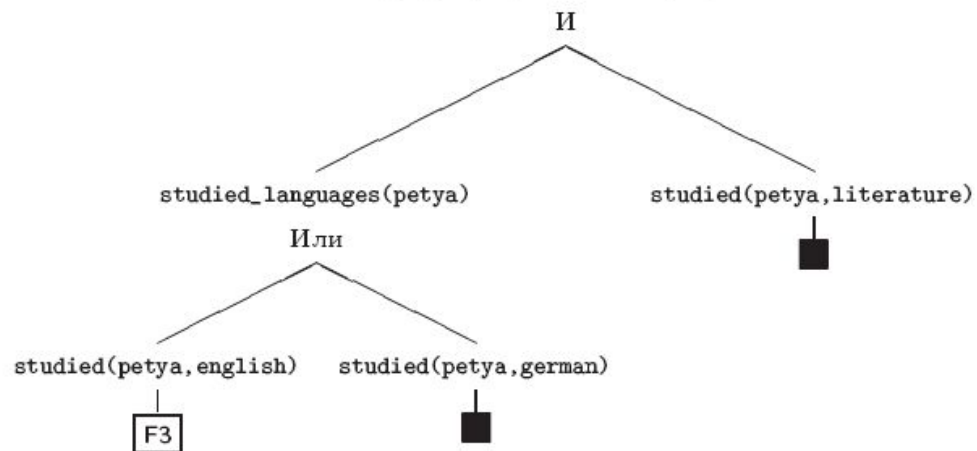
studied(petya, mathematics), studied(petya, compscience).



C

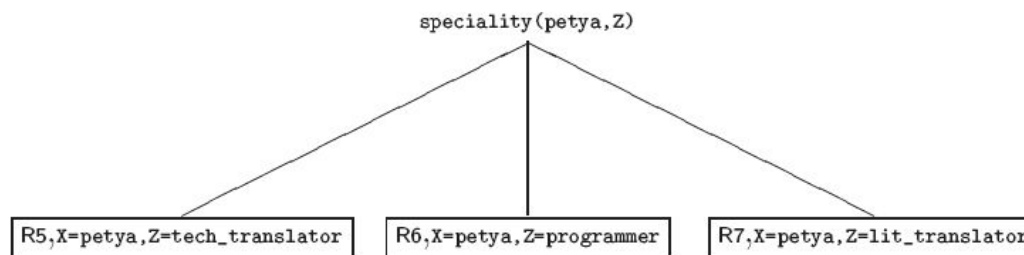
R7, X=petya, Z=lit_translator

studied_languages(petya), studied(petya, literature)

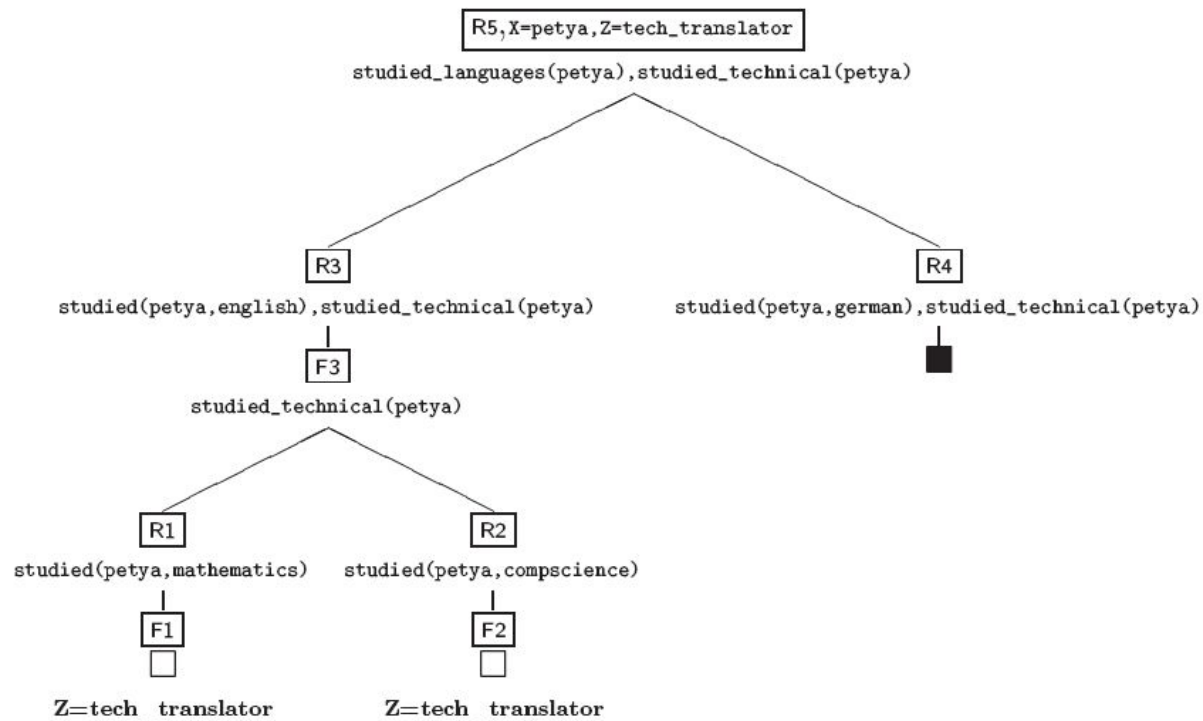


D

Дерево вывода

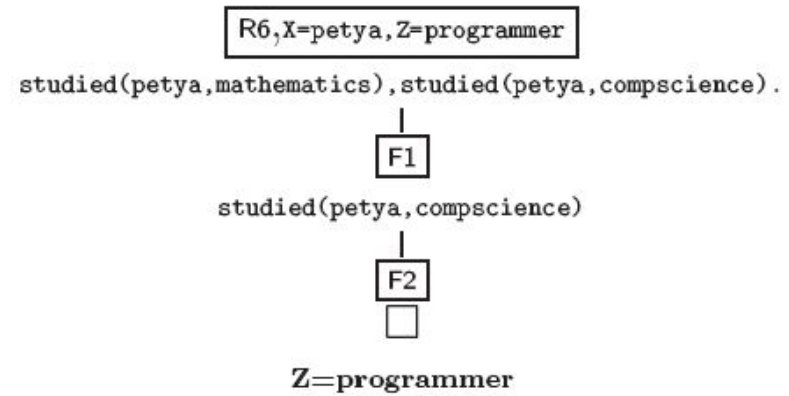


A

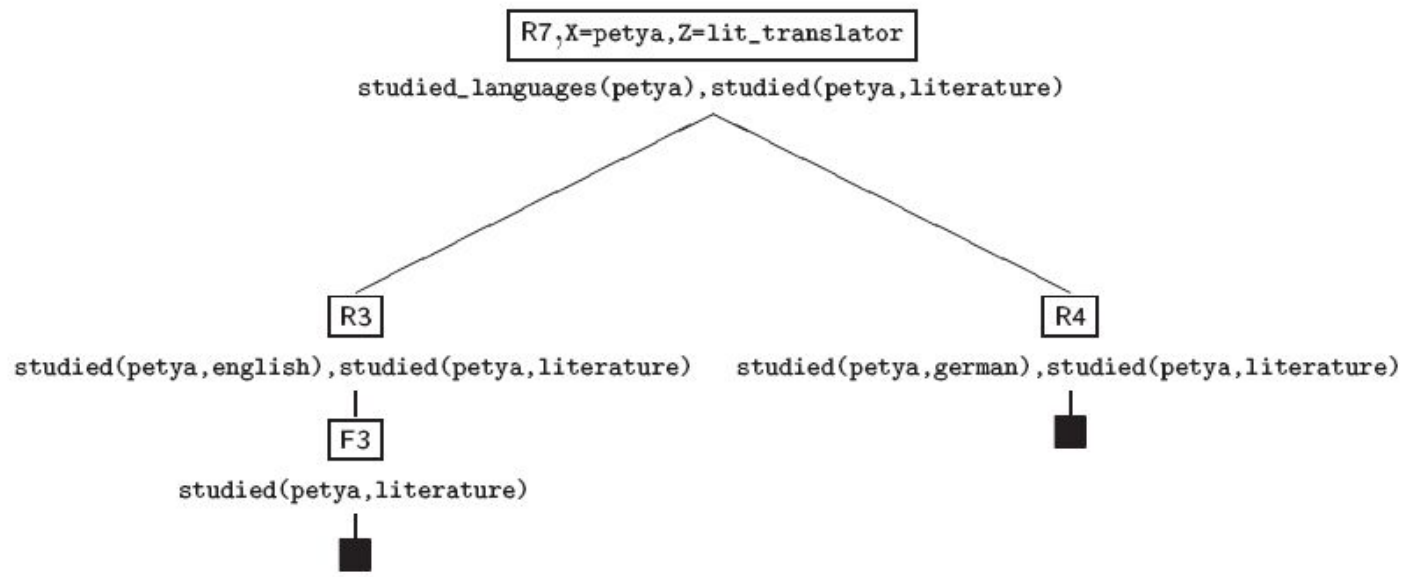


B

Дерево вывода (продолжение)



C



D

Механизм работы логического интерпретатора



- Запрос (целевое утверждение) сопоставляется (унифицируется) с головами имеющихся в программе правил и фактов.
- Начиная с первого найденного правила, целевое утверждение подменяется правой частью правила (с учетом замены переменных)
- Если встречается неуспех (правило не находится), то происходит откат (backtracking)

Какое отношение это имеет к логике?



```
speciality(X,tech_translator) :-  
studied_languages(X),studied_technical(X).
```

$(\forall X)$ $\text{specialty}(X, \text{tech_translator}) \subset$
 $\subset \text{studied_languages}(X) \wedge \text{studied_technical}(X)$

Дедуктивные базы данных



- В приведенном выше примере можно условно выделить базу фактов (кто какой предмет изучал) и базу правил
- Дедуктивные базы данных – это базы данных, снабженные средствами логического программирования для вывода дополнительных фактов
- Примеры: Mercury, Datalog

Более сложный пример



- Автоматическое построение учебных планов
- Опишем зависимости между дисциплинами:
 - `depends(lin_alg, math_logic)`
 - `depends(logic_prog, math_logic)`.
 - `depends(compscience, lin_alg)`.
- Опишем требуемые для специальности дисциплины:
 - `requires(programmer, compscience)`.
- Как понять, что должен изучать программист?

Более сложный пример (продолжение)

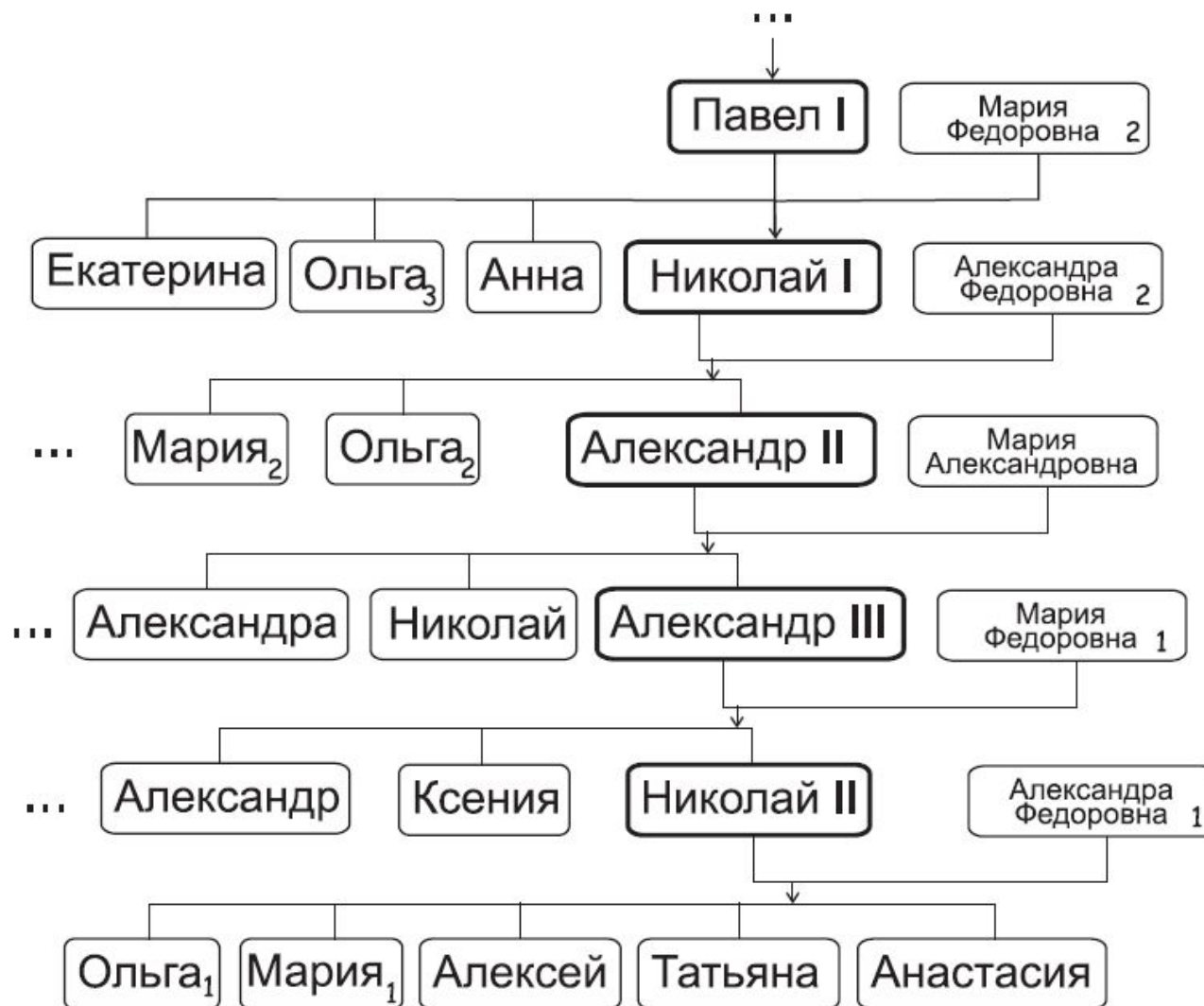


```
need_to_study(S,D) :- requires(S,D) .  
need_to_study(S,D) :- need_to_study(S,X) ,  
                        depends(X,D) .
```

Что видите интересного в этом примере?

- Рекурсия
- Та же предметная область (дисциплины, специальности) описана в других терминах
 - Напоминает нормальные формы БД?

Классический пример



Как можно описать предметную область?



```
father(nicholas_ii,olga_1).  
father(nicholas_ii,tatyana).  
mother(alexandra_fedorovna_1,olga_1).  
mother(alexandra_fedorovna_1,tatyana).
```

```
parent(nicholas_ii,olga_1).  
parent(alexandra_fedorovna_1,olga_1).  
parent(nicholas_ii,tatyana).  
parent(alexandra_fedorovna_1,tatyana).  
male(nicholas_ii).  
female(alexandra_fedorovna_1).  
father(X,Y) :- parent(X,Y), male(X).  
mother(X,Y) :- parent(X,Y), female(X).
```

```
parents(nicholas_ii,alexandra_fedorovna_1,olga_1).  
parents(nicholas_ii,alexandra_fedorovna_1,tatyana).
```

Какие достоинства и недостатки у каждого из таких способов описания?

Описание отношений родства



```
sibling(X,Y) :- parents(A,B,X) , parents(A,B,Y) .
```

$$(\forall X) (\forall Y) (((\exists A) (\exists B) P(A,B,X) \wedge P(A,B,Y)) \supset S(X,Y))$$
