Применение суперкомпилятора языка Java для решения обратных задач (в стиле логического программирования)

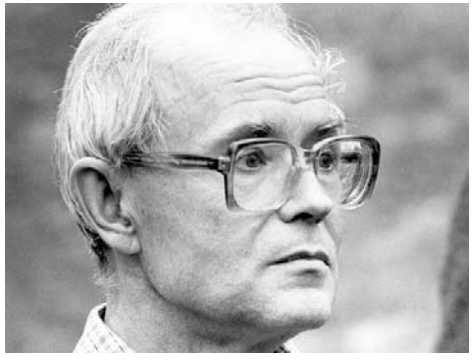
Андрей В. Климов
ИПМ им. М.В. Келдыша РАН, Москва
klimov@keldysh.ru

Введение в суперкомпиляцию на примере суперкомпилятора языка Java (JScp) и одной интересной задачи

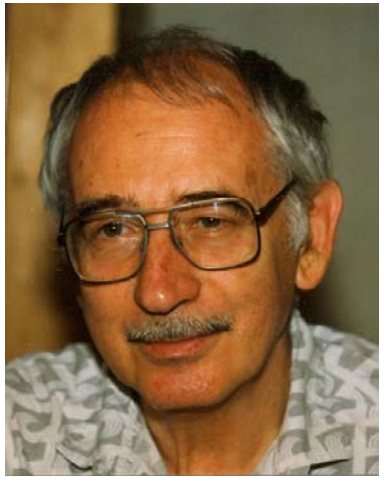
План

- Метавычисления:
  - основоположники, методы, задачи
- Что такое «суперкомпиляция» и «прогонка»?
- Задачи о расстановки ферзей на Java
  - и ее решение с помощью суперкомпилятора
- Краткая история суперкомпиляции
- Выводы

Метавычисления (metacomputation) = преобразования программ



Андрей П. Ершов
(смешанные вычисления)



Валентин Ф. Турчин
(суперкомпиляция)

Задачи

- Специализация программ
  - «по данным»

    $f(x,y) \rightarrow f_a(y) = f(a, y)$

  - «по управлению» (композиция)

    $f(x), g(x) \rightarrow f_g(x) = f(g(x))$

- Обратная задача
  - Решение уравнений

    $f(x) = a \rightarrow x$

    на примере JScp и задаче о ферзях
  - Инверсия программ

    $f(x) \rightarrow f^{-1}(y) = x$, если $y = f(x)$

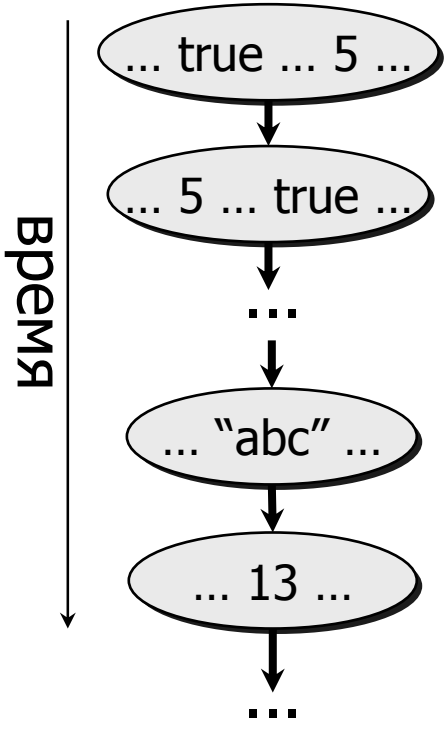
- Верификация программ
  - $f(x), P(y) \rightarrow P(f(x)) = \text{true}$

Neil D. Jones
(partial evaluation)

Yoshihiko Futamura
(partial computation)

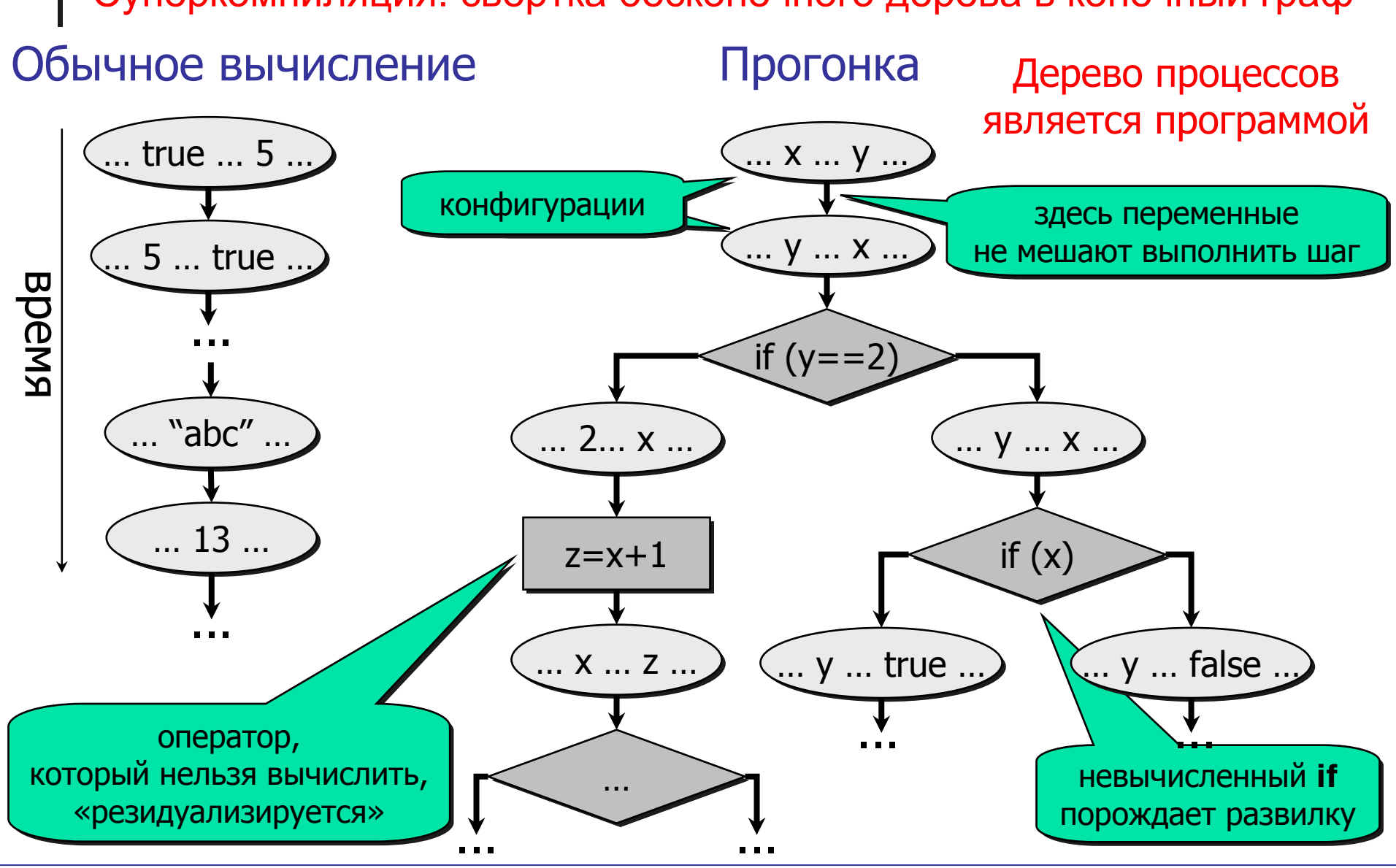
Прогонка: построение дерева процессов

Обычное вычисление

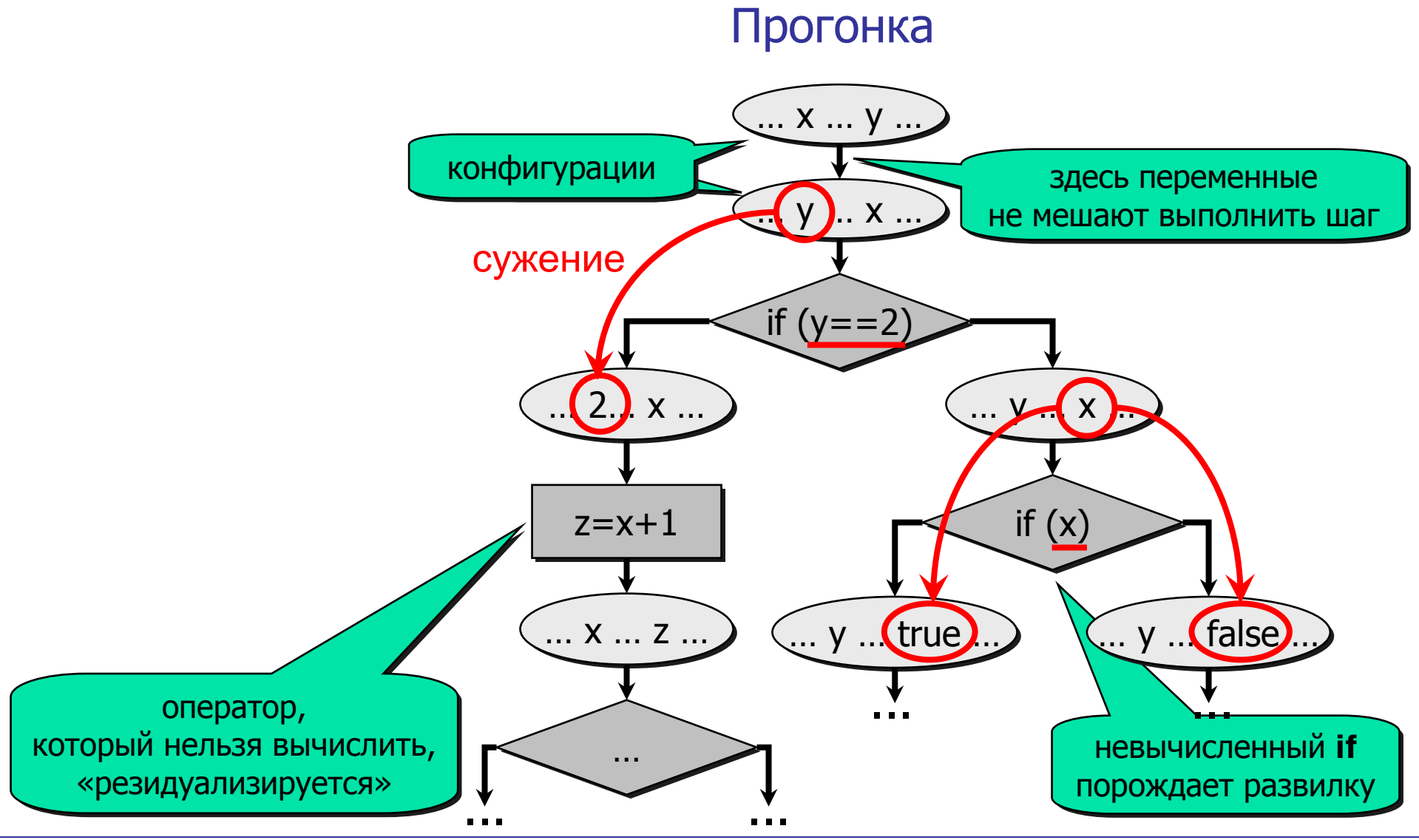


Прогонка: построение дерева процессов

Суперкомпиляция: свертка бесконечного дерева в конечный граф



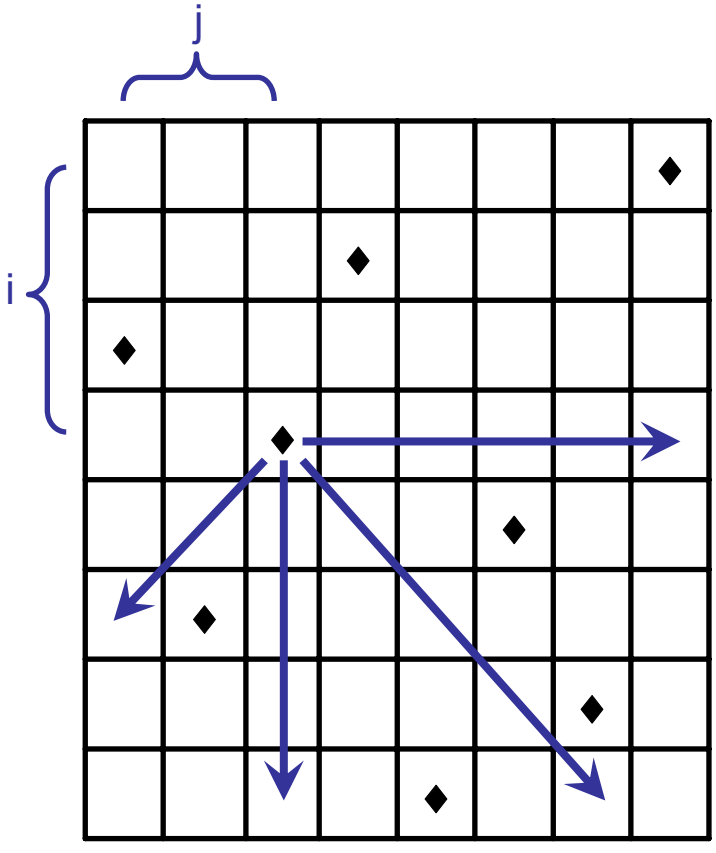
Прогонка: построение дерева процессов



Задача о ферзях на языке Java

Проверка, что ферзи стоят правильно

```
public class NQueens {
  final int n;
  final boolean[][] board;
...
  void checkNQueens() {
    for (int i = 0; i < n; i++) {
      int j = findQueenInRow(i);
      noQueensInRowToTheRight(i, j);
      noQueensInColumnBelow(i, j);
      noQueensInLeftDiagonal(i, j);
      noQueensInRightDiagonal(i, j);
  }   }
  int findQueenInRow(int i)
    for (int j = 0; j < n; j++) {
      if(board[i][j]) return j;
    throw new Error();
  }
  void noQueensInRowToTheRight(int i, int j) {
    for (int k = j+1; k < n; k++)
      if (board[i][k]) throw new Error();
  }
  void noQueensInColumnBelow(int i, int j) {
    for (int k = i+1; k < n; k++)
      if (board[k][j]) throw new Error();
  }
  void noQueensInLeftDiagonal(int i, int j) {
    for (int k = 1; i+k < n && 0 <= j-k && j-k < n; k++)
      if (board[i+k][j-k]) throw new Error();
  }
  void noQueensInRightDiagonal(int i, int j) {
    for (int k = 1; i+k < n && j+k < n; k++)
      if (board[i+k][j+k]) throw new Error();
  }
}
```



Задача о ферзях на языке Java

```
public class NQueens {
  final int n;
  final boolean[][] board;
...
  void checkNQueens() {
    for (int i = 0; i < n; i++) {
      int j = findQueenInRow(i);
      noQueensInRowToTheRight(i, j);
      noQueensInColumnBelow(i, j);
      noQueensInLeftDiagonal(i, j);
      noQueensInRightDiagonal(i, j);
  }   }
  int findQueenInRow(int i)
    for (int j = 0; j < n; j++) {
      if(board[i][j]) return j;
    throw new Error();
  }
  void noQueensInRowToTheRight(int i, int j) {
    for (int k = j+1; k < n; k++)
      if (board[i][k]) throw new Error();
  }
  void noQueensInColumnBelow(int i, int j) {
    for (int k = i+1; k < n; k++)
      if (board[k][j]) throw new Error();
  }
  void noQueensInLeftDiagonal(int i, int j) {
    for (int k = 1; i+k < n && 0 <= j-k && j-k < n; k++)
      if (board[i+k][j-k]) throw new Error();
  }
  void noQueensInRightDiagonal(int i, int j) {
    for (int k = 1; i+k < n && j+k < n; k++)
      if (board[i+k][j+k]) throw new Error();
  }
}
```

```
static void checkAndPrintNQueens
  (int n, boolean[][] board)
{
  NQueens b = new NQueens(n, board);
  b.checkNQueens();
  b.printBoard();
}
```

Задача о ферзях на языке Java (N = 4)

```
public class NQueens {
  final int n;
  final boolean[][] board;
...
  void checkNQueens() {
    for (int i = 0; i < n; i++) {
      int j = findQueenInRow(i);
      noQueensInRowToTheRight(i, j);
      noQueensInColumnBelow(i, j);
      noQueensInLeftDiagonal(i, j);
      noQueensInRightDiagonal(i, j);
  }   }
  int findQueenInRow(int i)
    for (int j = 0; j < n; j++) {
      if(board[i][j]) return j;
    throw new Error();
  }
  void noQueensInRowToTheRight(int i, int j) {
    for (int k = j+1; k < n; k++)
      if (board[i][k]) throw new Error();
  }
  void noQueensInColumnBelow(int i, int j) {
    for (int k = i+1; k < n; k++)
      if (board[k][j]) throw new Error();
  }
  void noQueensInLeftDiagonal(int i, int j) {
    for (int k = 1; i+k < n && 0 <= j-k && j-k < n; k++)
      if (board[i+k][j-k]) throw new Error();
  }
  void noQueensInRightDiagonal(int i, int j) {
    for (int k = 1; i+k < n && j+k < n; k++)
      if (board[i+k][j+k]) throw new Error();
  }
}
```

```
static void checkAndPrintNQueens
  (int n, boolean[][] board)
{
  NQueens b = new NQueens(n, board);
  b.checkNQueens();
  b.printBoard();
}

static void check4Queens(boolean[][] b) {
  checkAndPrintNQueens(4, b);
}
```

Головная программа с примером вызова

```
public static void main(String[] args) {
  check4Queens(
    new boolean[][]{
      new boolean[]{false,true ,false,false},
      new boolean[]{false,false,false,true },
      new boolean[]{true ,false,false,false},
      new boolean[]{false,false,true ,false}});
}
```

Результат печати

```
. * . .
. . . *
* . . .
. . * .
```

Задача о ферзях на языке Java (N = 4)

```
public class NQueens {
  final int n;
  final boolean[][] board;
...
  void checkNQueens() {
    for (int i = 0; i < n; i++) {
      int j = findQueenInRow(i);
      noQueensInRowToTheRight(i, j);
      noQueensInColumnBelow(i, j);
      noQueensInLeftDiagonal(i, j);
      noQueensInRightDiagonal(i, j);
  }   }
  int findQueenInRow(int i)
    for (int j = 0; j < n; j++) {
      if(board[i][j]) return j;
    throw new Error();
  }
  void noQueensInRowToTheRight(int i, int j) {
    for (int k = j+1; k < n; k++)
      if (board[i][k]) throw new Error();
  }
  void noQueensInColumnBelow(int i, int j) {
    for (int k = i+1; k < n; k++)
      if (board[k][j]) throw new Error();
  }
  void noQueensInLeftDiagonal(int i, int j) {
    for (int k = 1; i+k < n && 0 <= j-k && j-k < n; k++)
      if (board[i+k][j-k]) throw new Error();
  }
  void noQueensInRightDiagonal(int i, int j) {
    for (int k = 1; i+k < n && j+k < n; k++)
      if (board[i+k][j+k]) throw new Error();
  }
}
```

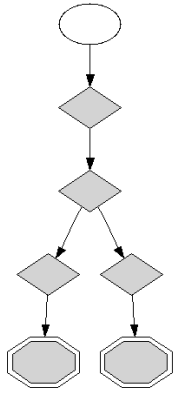
```
static void checkAndPrintNQueens
  (int n, boolean[][] board)
{
  NQueens b = new NQueens(n, board);
  b.checkNQueens();
  b.printBoard();
}

static void check4Queens(boolean[][] b) {
  checkAndPrintNQueens(4, b);
}
```

Суперкомпилировать!

Задача о ферзях на языке Java (N = 4)

```
public class NQueens {
  final int n;
  final boolean[][] board;
...
  void checkNQueens() {
    for (int i = 0; i < n; i++) {
      int j = findQueenInRow(i);
      noQueensInRowToTheRight(i, j);
      noQueensInColumnBelow(i, j);
      noQueensInLeftDiagonal(i, j);
      noQueensInRightDiagonal(i, j);
  }   }
  int findQueenInRow(int i)
    for (int j = 0; j < n; j++) {
      if(board[i][j]) return j;
    throw new Error();
  }
  void noQueensInRowToTheRight(int i, int j) {
    for (int k = j+1; k < n; k++)
      if (board[i][k]) throw new Error();
  }
  void noQueensInColumnBelow(int i, int j) {
    for (int k = i+1; k < n; k++)
      if (board[k][j]) throw new Error();
  }
  void noQueensInLeftDiagonal(int i, int j) {
    for (int k = 1; i+k < n && 0 <= i-k && i-k < n; k++)
      if (board[i+k][j
  }
  void noQueensInRight
    for (int k = 1; i+
      if (board[i+k][j-
  }
}
```



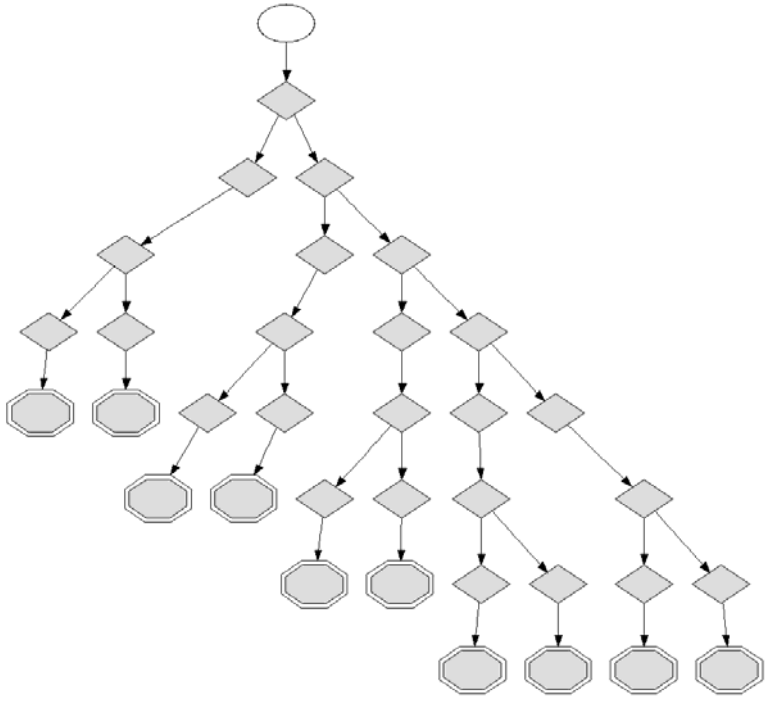
```
static void check4Queens (final boolean[][] b_1)
{
  final boolean b_0_0_13 = b_1[0][0];
  final boolean b_0_1_17 = b_1[0][1];
  final boolean b_0_2_21 = b_1[0][2];
  final boolean b_0_3_25 = b_1[0][3];
  final boolean b_1_0_37 = b_1[1][0];
  final boolean b_1_1_41 = b_1[1][1];
  final boolean b_1_2_45 = b_1[1][2];
  final boolean b_1_3_49 = b_1[1][3];
  final boolean b_2_0_61 = b_1[2][0];
  final boolean b_2_1_65 = b_1[2][1];
  final boolean b_2_2_69 = b_1[2][2];
  final boolean b_2_3_73 = b_1[2][3];
  final boolean b_3_0_85 = b_1[3][0];
  final boolean b_3_1_89 = b_1[3][1];
  final boolean b_3_2_93 = b_1[3][2];
  final boolean b_3_3_97 = b_1[3][3];
  if (b_0_0_13) {
    throw new java.lang.Error();}
  if (b_0_1_17) {
    if (b_0_2_21  ||  b_0_3_25  ||  b_1_1_41
    ||  b_2_1_65  ||  b_3_1_89  ||  b_1_0_37
    ||  b_1_2_45  ||  b_2_3_73  || !b_1_3_49
    ||  b_3_3_97  ||  b_2_2_69  || !b_2_0_61
    ||  b_3_0_85  || !b_3_2_93) {
      throw new java.lang.Error();}
    java.lang.System.out.println(".*..");
    java.lang.System.out.println("...*");
    java.lang.System.out.println("*...");
    java.lang.System.out.println("..*.");
    return;}
  else {
    if(!b_0_2_21  ||  b_0_3_25  ||  b_1_2_45
    ||  b_2_2_69  ||  b_3_2_93  ||  b_1_1_41
    ||  b_2_0_61  ||  b_1_3_49  || !b_1_0_37
    ||  b_3_0_85  ||  b_2_1_65  || !b_2_3_73
    ||  b_3_3_97  || !b_3_1_89) {
      hrow new java.lang.Error();}
      a.lang.System.out.println("..*.");
      a.lang.System.out.println("*...");
      a.lang.System.out.println("...*");
      a.lang.System.out.println(".*..");
      urn;}
```

Размер доски и число ферзей N = 4

Число решений: 2

Время суперкомпиляции: 1 сек

Задача о ферзях на языке Java (N = 5)

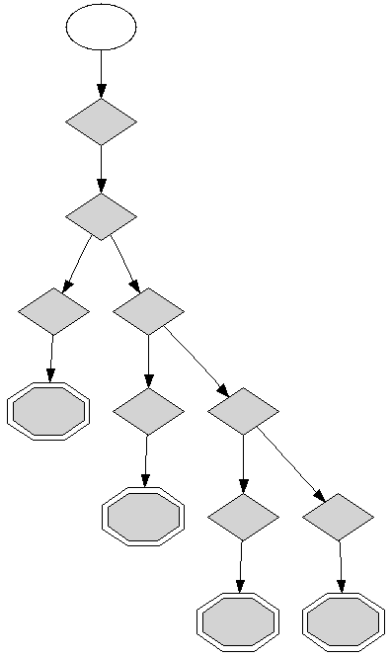


Размер доски и число ферзей N = 5

Число решений: 10

Время суперкомпиляции: 2 сек

Задача о ферзях на языке Java (N = 6)



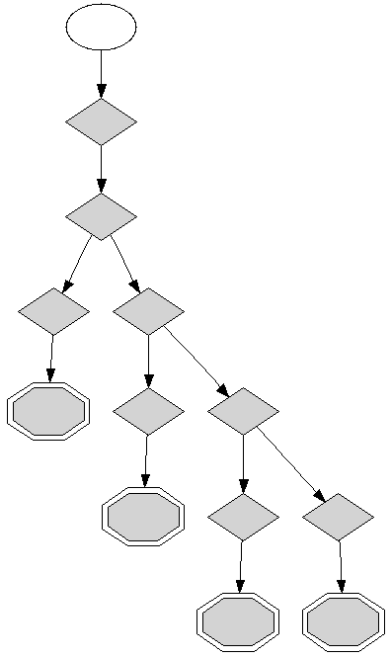
Размер доски и число ферзей N = 8

Число решений: 4

Время суперкомпиляции: 6 сек

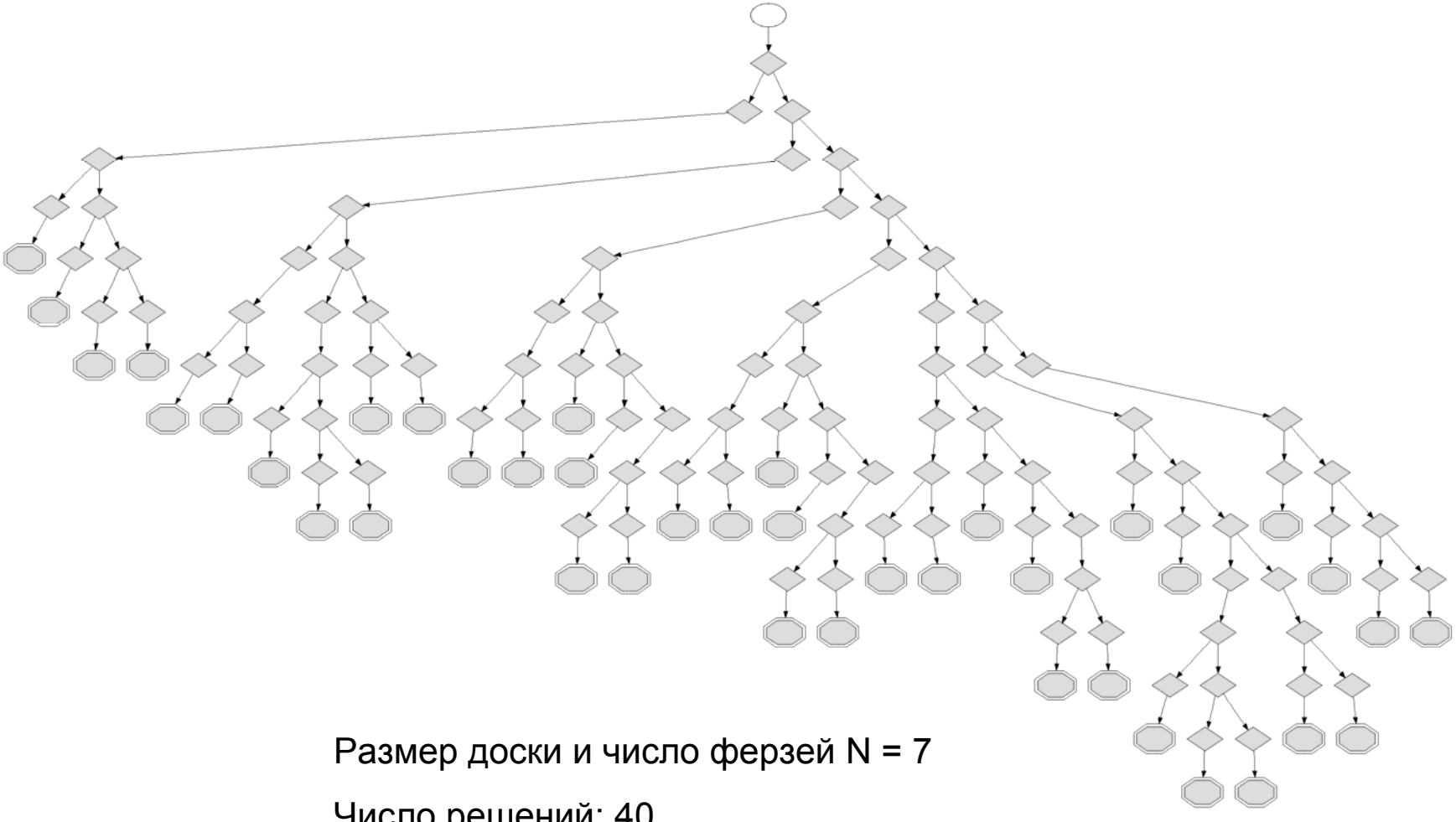
Задача о ферзях на языке Java (N = 6)

```
static void check6Queens (final boolean[][] b_1)
{
  final boolean b_0_0_15 = b_1[0][0];
    ...
  final boolean b_5_5_205 = b_1[5][5];
  if (b_0_0_15) {
    throw new java.lang.Error();}
  if (b_0_1_19) {
    if (b_0_2_23  ||  b_0_3_27  ||  b_0_4_31
    ||  b_0_5_35  ||  b_1_1_53  ||  b_2_1_87
    ||  b_3_1_121 ||  b_4_1_155 ||  b_5_1_189
    ||  b_1_0_49  ||  b_1_2_57  ||  b_2_3_95
    ||  b_3_4_133 ||  b_4_5_171 || !b_1_3_61
    ||  b_1_4_65  ||  b_1_5_69  ||  b_3_3_129
    ||  b_4_3_163 ||  b_5_3_197 ||  b_2_2_91
    ||  b_4_0_151 ||  b_2_4_99  ||  b_3_5_137
    ||  b_2_0_83  || !b_2_5_103 ||  b_5_5_205
    ||  b_5_2_193 || !b_3_0_117 ||  b_3_2_125
    ||  b_5_0_185 || !b_4_2_159 ||  b_4_4_167
    || !b_5_4_201) {
      throw new java.lang.Error();}
    java.lang.System.out.println(".*....");
    java.lang.System.out.println("...*..");
    java.lang.System.out.println(".....*");
    java.lang.System.out.println("*.....");
    java.lang.System.out.println("..*...");
    java.lang.System.out.println("....*.");
    return;}
  else {
    if (b_0_2_23) {
      if (b_0_3_27  ||  b_0_4_31  ||  b_0_5_35
      ||  b_1_2_57  ||  b_2_2_91  ||  b_3_2_125
      ||  b_4_2_159 ||  b_5_2_193 ||  b_1_1_53
      ||  b_2_0_83  ||  b_1_3_61  ||  b_2_4_99
      ||  b_3_5_137 ||  b_1_0_49  ||  b_1_4_65
      || !b_1_5_69  ||  b_2_5_103 ||  b_4_5_171
      ||  b_5_5_205 ||  b_3_3_129 ||  b_5_1_189
      || !b_2_1_87  ||  b_2_3_95  ||  b_3_1_121
      ||  b_4_1_155 ||  b_3_0_117 ||  b_4_3_163
      ||  b_5_4_201 || !b_3_4_133 ||  b_4_4_167
      || !b_4_0_151 ||  b_5_0_185 || !b_5_3_197) {
        throw new java.lang.Error();}
      java.lang.System.out.println("..*...");
      java.lang.System.out.println(".....*");
      java.lang.System.out.println(".*....");
      java.lang.System.out.println("....*.");
      java.lang.System.out.println("*.....");
      java.lang.System.out.println("...*..");
      return;}
```



```
    else {
      if (b_0_3_27) {
        if (b_0_4_31  ||  b_0_5_35  ||  b_1_3_61
        ||  b_2_3_95  ||  b_3_3_129 ||  b_4_3_163
        ||  b_5_3_197 ||  b_1_2_57  ||  b_2_1_87
        ||  b_3_0_117 ||  b_1_4_65  ||  b_2_5_103
        || !b_1_0_49  ||  b_1_1_53  ||  b_1_5_69
        ||  b_2_0_83  ||  b_4_0_151 ||  b_5_0_185
        ||  b_3_2_125 ||  b_5_4_201 ||  b_2_2_91
        || !b_2_4_99  ||  b_3_4_133 ||  b_4_4_167
        ||  b_4_2_159 ||  b_5_1_189 ||  b_3_5_137
        || !b_3_1_121 ||  b_4_1_155 || !b_4_5_171
        ||  b_5_5_205 || !b_5_2_193) {
          throw new java.lang.Error();}
        java.lang.System.out.println("...*..");
        java.lang.System.out.println("*.....");
        java.lang.System.out.println("....*.");
        java.lang.System.out.println(".*....");
        java.lang.System.out.println(".....*");
        java.lang.System.out.println("..*...");
        return;}
      else {
        if(!b_0_4_31  ||  b_0_5_35  ||  b_1_4_65
        ||  b_2_4_99  ||  b_3_4_133 ||  b_4_4_167
        ||  b_5_4_201 ||  b_1_3_61  ||  b_2_2_91
        ||  b_3_1_121 ||  b_4_0_151 ||  b_1_5_69
        ||  b_1_0_49  ||  b_1_1_53  || !b_1_2_57
        ||  b_3_2_125 ||  b_4_2_159 ||  b_5_2_193
        ||  b_2_1_87  ||  b_3_0_117 ||  b_2_3_95
        ||  b_4_5_171 || !b_2_0_83  ||  b_2_5_103
        ||  b_5_0_185 ||  b_5_3_197 ||  b_3_3_129
        || !b_3_5_137 ||  b_5_5_205 ||  b_4_1_155
        || !b_4_3_163 || !b_5_1_189) {
          throw new java.lang.Error();}
        java.lang.System.out.println("....*.");
        java.lang.System.out.println("..*...");
        java.lang.System.out.println("*.....");
        java.lang.System.out.println(".....*");
        java.lang.System.out.println("...*..");
        java.lang.System.out.println(".*....");
        return;}}}}
}
```

Задача о ферзях на языке Java (N = 7)

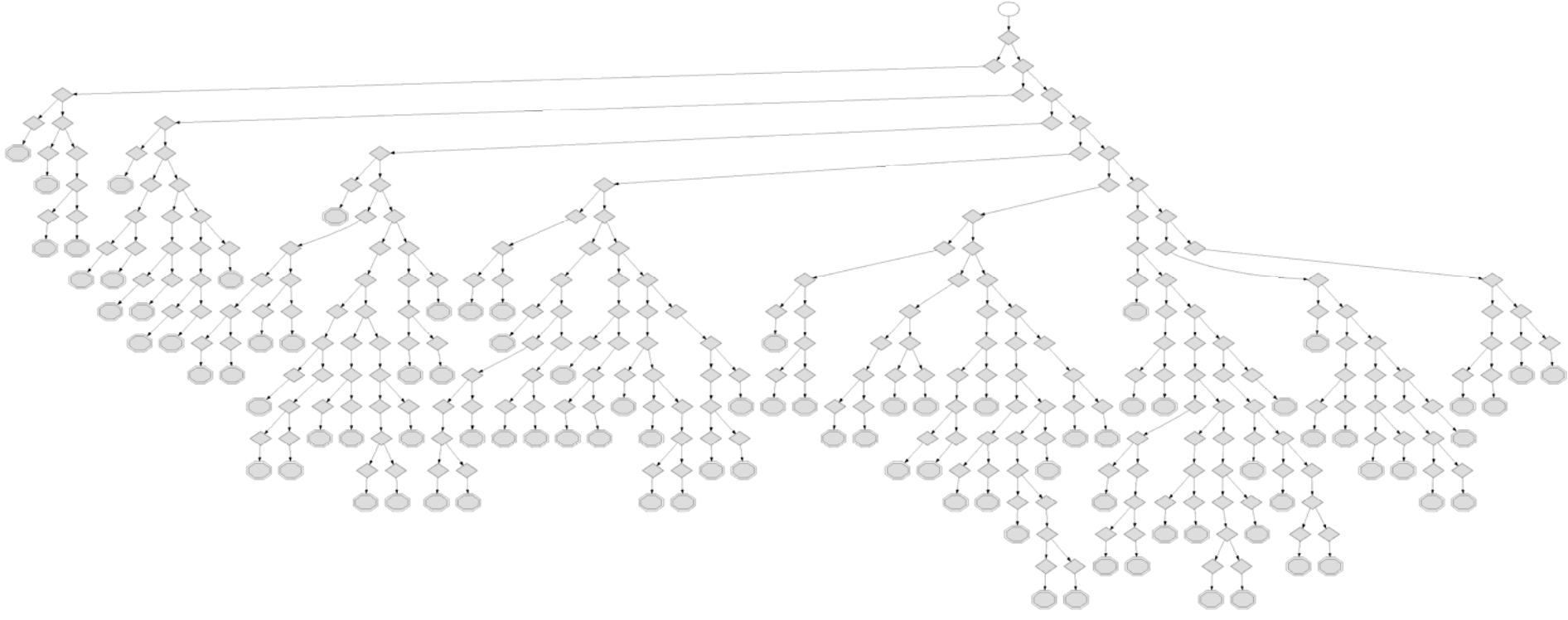


Размер доски и число ферзей N = 7

Число решений: 40

Время суперкомпиляции: 37 сек

Задача о ферзях на языке Java (N = 8)

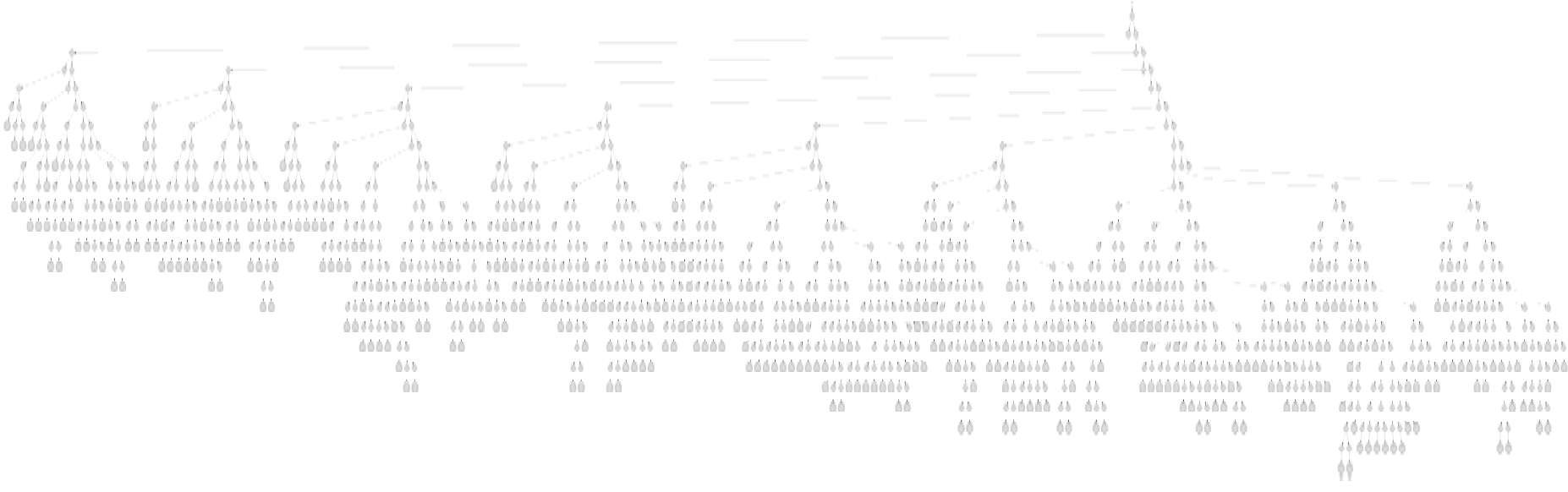


Размер доски и число ферзей N = 8

Число решений: 92

Время суперкомпиляции: 290 сек = ~6 мин

Задача о ферзях на языке Java (N = 9)



Размер доски и число ферзей N = 9

Число решений: 352

Время суперкомпиляции: 3840 сек = 64 мин

Время суперкомпиляции задачи об N ферзях

Число ферзей (N) (доска N*N)	Время суперкомпиляции (сек) (Pentium 4, 1.8 GHz)
1	<1
2	<1
3	<1
4	1
5	2
6	6
7	37
8	289
9	3833

Краткая история суперкомпиляции

1972 1980-е – для Рефала 2000-е – для Java	Статья В.Ф. Турчина «Эквивалентные преобразования рекурсивных функций, описанных на языке Рефал» (прогонка, «универсальный решающий алгоритм»)
1974	В.Ф. Турчин рассказал о суперкомпиляции группе студентов на серии семинаров в Москве
1980-е	В.Ф. Турчин разрабатывает первые суперкомпиляторы для функционального языка Рефал (CUNY, Нью-Йорк)
1980-е – 1990-е	Серия статей В.Ф. Турчина по суперкомпиляции Рефала
1990-е	Работы по теории суперкомпиляции в Копенгагенском университете (Р. Глюк, М. Сёренсен) в сотрудничестве с нами
1995	Книга С.М. Абрамова «Метавычисления и их приложения»
1993 – 2000-е	А.П. Немытых (ИПС, Переславль-Залесский) продолжил и завершил работу В.Ф. Турчина над суперкомпилятором для Рефала
2007	Книга А.П. Немытых «Суперкомпилятор SCP4: общая структура»
1998 – 2000-е	Суперкомпилятор языка Java (JScp) (Анд.В. Климов, Арк.В. Климов, А.Б. Шворин)
2008 – …	Суперкомпилятор языка с функциями высших порядков для верификации программ (И.Г. Ключников, С.А. Романенко)

Выводы

- Метод суперкомпиляции В.Ф. Турчина был расширен с функциональных языков (Рефал и др.) на объектно-ориентированный язык Java
  - Реализован экспериментальный суперкомпилятор JScp
- На примере была продемонстрирована часть функциональности JScp, используемая для решения обратной задачи – *прогонка*
  - это наиболее проработанная, алгоритмическая часть
- Суперкомпиляция для языка Java удалась благодаря хорошим свойствам языка («managed code»)
  - для C, C++ это нереально!
- Распространение «индустриальных» языков (Java, C#) с такими хорошими свойствами открывает возможность применять метавычисления на практике (специализация программ, решение обратной задачи, обращение программ, верификация программ)

Андрей В. Климов
klimov@keldysh.ru
ИПМ им. М.В. Келдыша РАН, Москва

http://pat.keldysh.ru
program analysis and transformation
Сектор анализа и преобразований программ