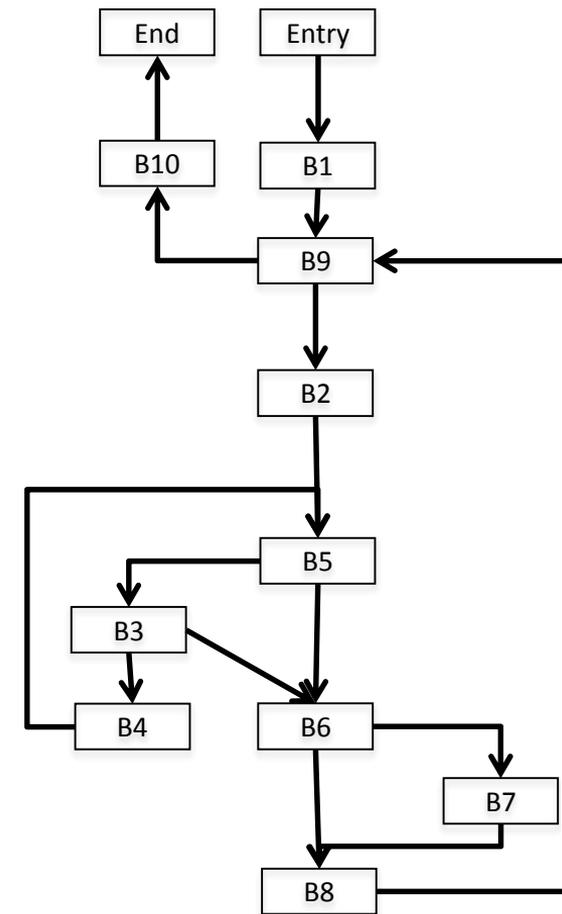


Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода

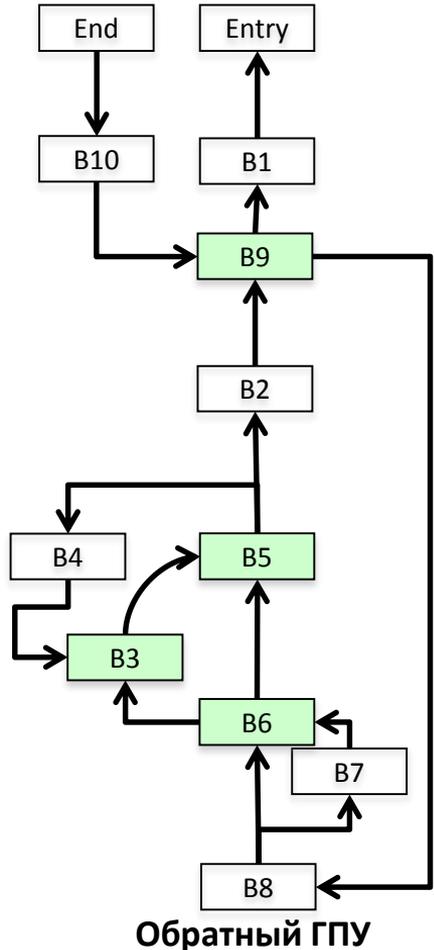
Бротиковская Данута, 420 группа

Исходный граф потока управления



Домашнее задание №2 по курсу “Конструирование компиляторов”

Найти бесполезный код. Построение обратной границы доминирования



Точки сбора – B3, B5, B6, B9

$Pred(B3) = \{4, 6\}$

$IDOM(B3) = \{6\}$

$Pred(B5) = \{3, 6\}$

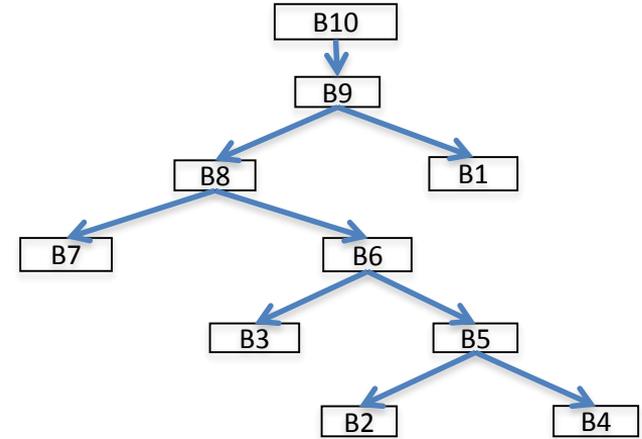
$IDOM(B5) = \{6\}$

$Pred(B6) = \{7, 8\}$

$IDOM(B6) = \{8\}$

$Pred(B9) = \{2, 10\}$

$IDOM(B9) = \{10\}$



Обратное дерево доминаторов

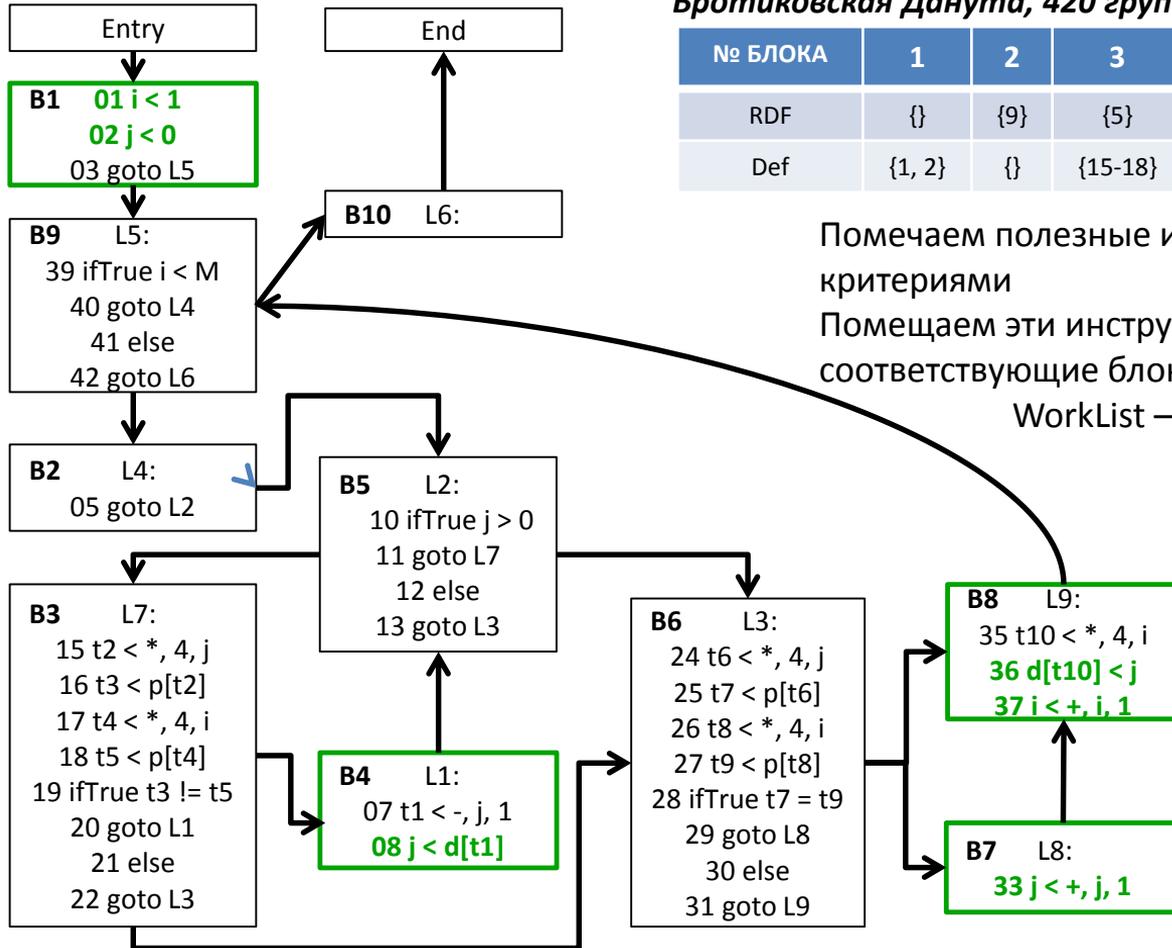
№ БЛОКА	1	2	3	4	5	6	7	8	9	10
PRED	{9}	{4, 5}	{4, 6}	{5}	{3, 6}	{7, 8}	{8}	{9}	{2, 10}	{}
DOM	{1, 9, 10}	{2, 5, 6, 8, 9, 10}	{3, 6, 8, 9, 10}	{4, 5, 6, 8, 9, 10}	{5, 6, 8, 9, 10}	{6, 8, 9, 10}	{7, 8, 9, 10}	{8, 9, 10}	{9, 10}	{}
IDOM	9	5	6	5	6	8	8	9	10	-
RDF	{}	{9}	{5}	{3}	{3, 9}	{9}	{6}	{9}	{9}	{}

Домашнее задание №4 по курсу "Конструирование компиляторов"

Удаление бесполезного кода. Проход Mark

Бротиковская Данута, 420 группа

№ БЛОКА	1	2	3	4	5	6	7	8	9	10
RDF	{}	{9}	{5}	{3}	{3, 9}	{9}	{6}	{9}	{9}	{}
Def	{1, 2}	{}	{15-18}	{7}	{}	{24-27}	{33}	{35-37}	{}	{}



Помечаем полезные инструкции и ветви в соответствии с критериями

Помещаем эти инструкции в WorkList, отмечаем соответствующие блоки

WorkList – 1, 2, 8, 36, 37, 33

Globals	i	j	p	d
Blocks	{1, 8}	{1, 4, 7}	{}	{8}

Критерии полезности кода

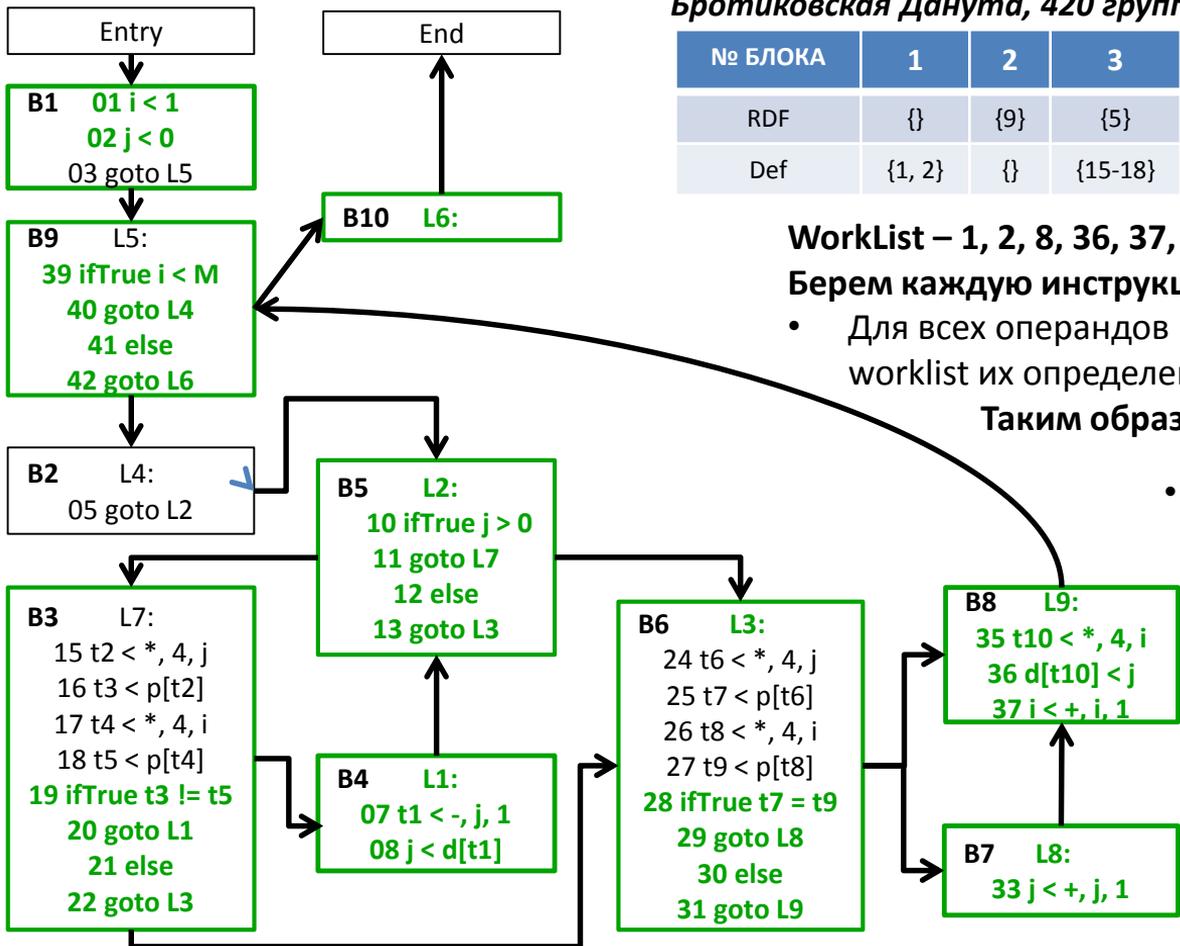
- вычисляет возвращаемое значение процедуры (return)
- является обращением к функции ввода-вывода
- вычисляет значение глобальной переменной (**Globals**)

Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода. Проход Mark

Бротиковская Данута, 420 группа

№ БЛОКА	1	2	3	4	5	6	7	8	9	10
RDF	{}	{9}	{5}	{3}	{3, 9}	{9}	{6}	{9}	{9}	{}
Def	{1, 2}	{}	{15-18}	{7}	{}	{24-27}	{33}	{35-37}	{}	{}



WorkList – 1, 2, 8, 36, 37, 33

Берем каждую инструкцию из WorkList

- Для всех операндов инструкции помечаются и добавляются в worklist их определения, те, что ещё не были помечены.

Таким образом, еще помечаются инструкции 7 и 35

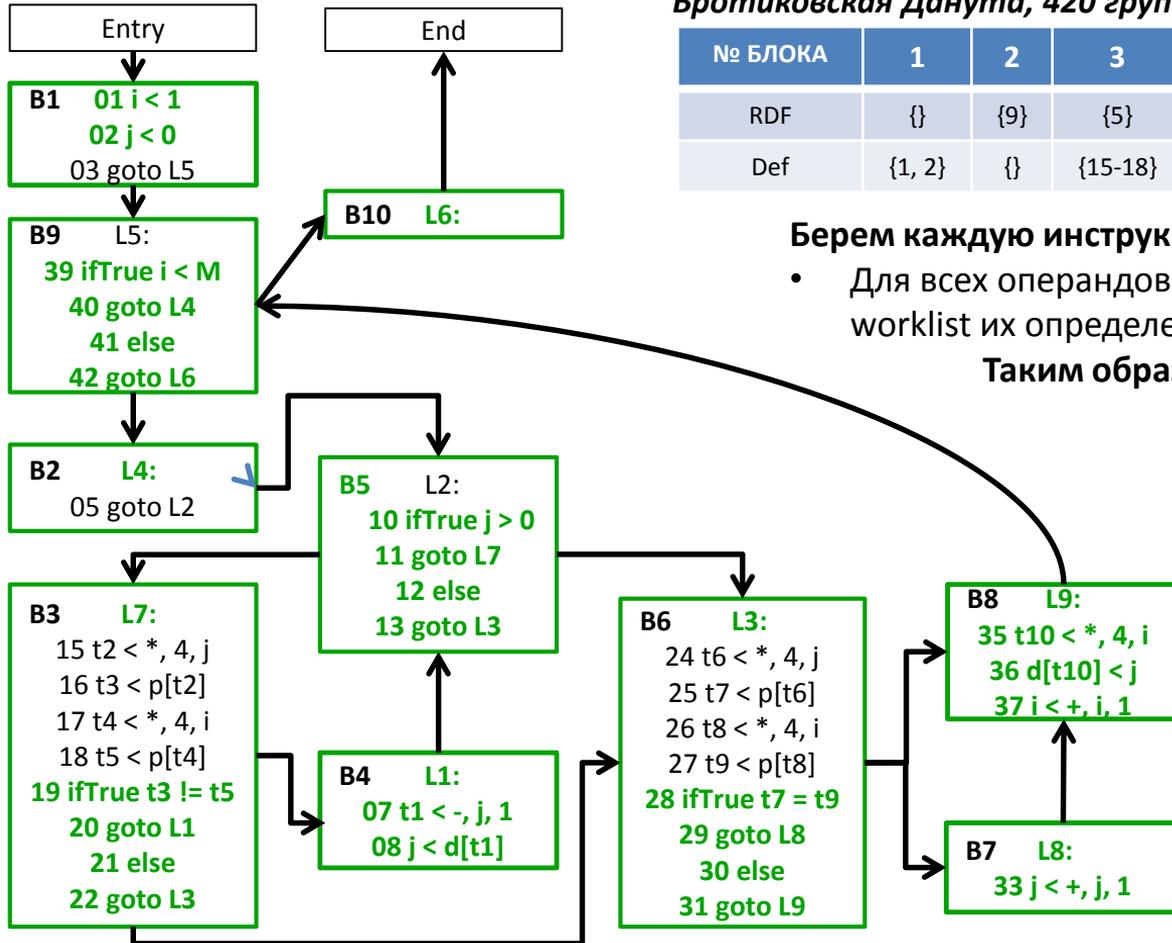
- Помечаем инструкции ветвления в блоках и вносим их в WorkList, входящих в обратную границу доминирования выбранной инструкции

Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода. Проход Mark

Бротиковская Данута, 420 группа

№ БЛОКА	1	2	3	4	5	6	7	8	9	10
RDF	{}	{9}	{5}	{3}	{3, 9}	{9}	{6}	{9}	{9}	{}
Def	{1, 2}	{}	{15-18}	{7}	{}	{24-27}	{33}	{35-37}	{}	{}



Берем каждую инструкцию из WorkList

- Для всех операндов инструкции помечаются и добавляются в worklist их определения, те, что ещё не были помечены.

Таким образом, ещё помечаются инструкции 7 и 35

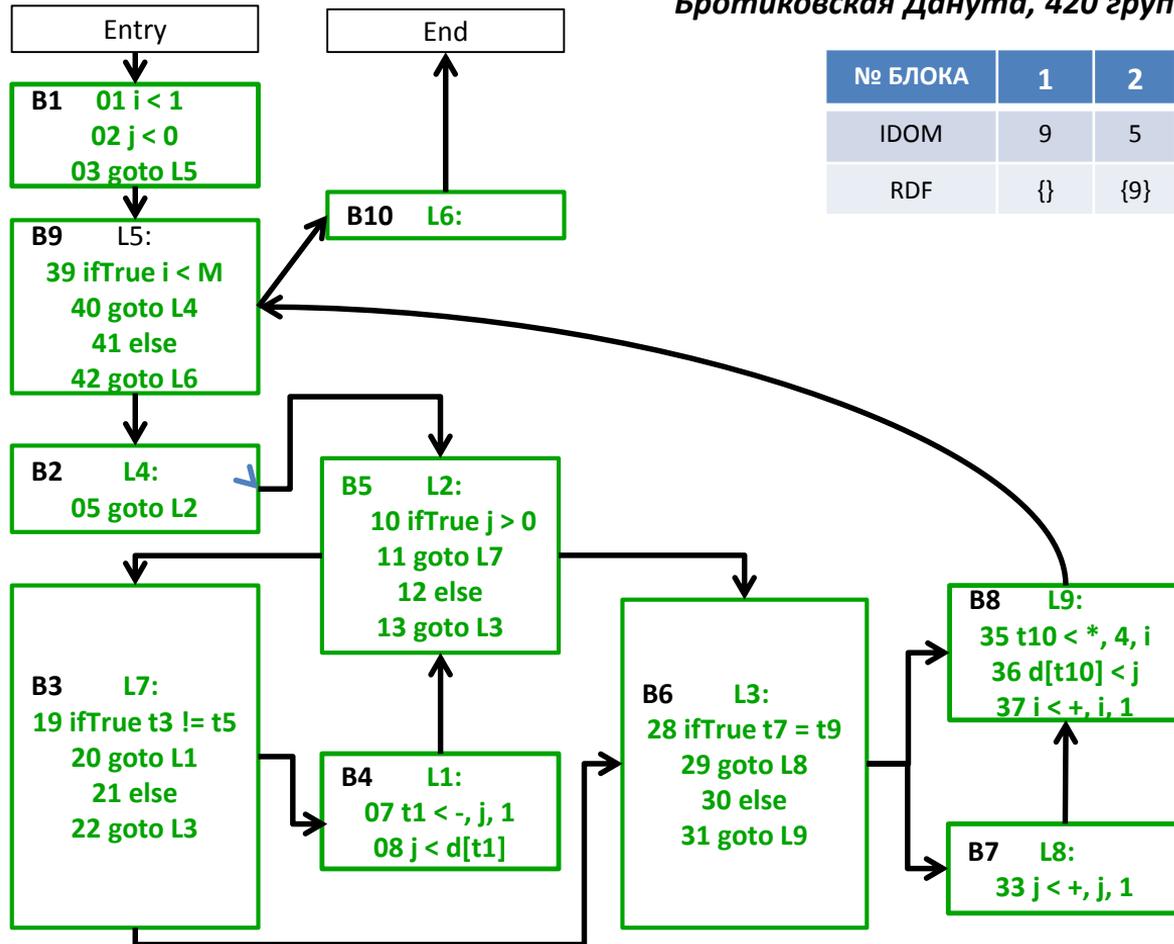
- Помечаем инструкции ветвления в блоках и вносим их в WorkList, входящих в обратную границу доминирования выбранной инструкции

Результат прохода Mark продемонстрирован на слайде

Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода. Проход Sweep

Бротиковская Данута, 420 группа



№ БЛОКА	1	2	3	4	5	6	7	8	9	10
IDOM	9	5	6	5	6	8	8	9	10	-
RDF	{}	{9}	{5}	{3}	{3, 9}	{9}	{6}	{9}	{9}	{}

Рассматриваем непомеченные инструкции

Для инструкций ветвления заменяем переходы на переходы на ближайшие постдоминаторы
Непомеченные инструкции присваивания удаляем

Результат продемонстрирован на слайде – удалены инструкции 15-18 и 24-27

Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода

Бротиковская Данута, 420 группа

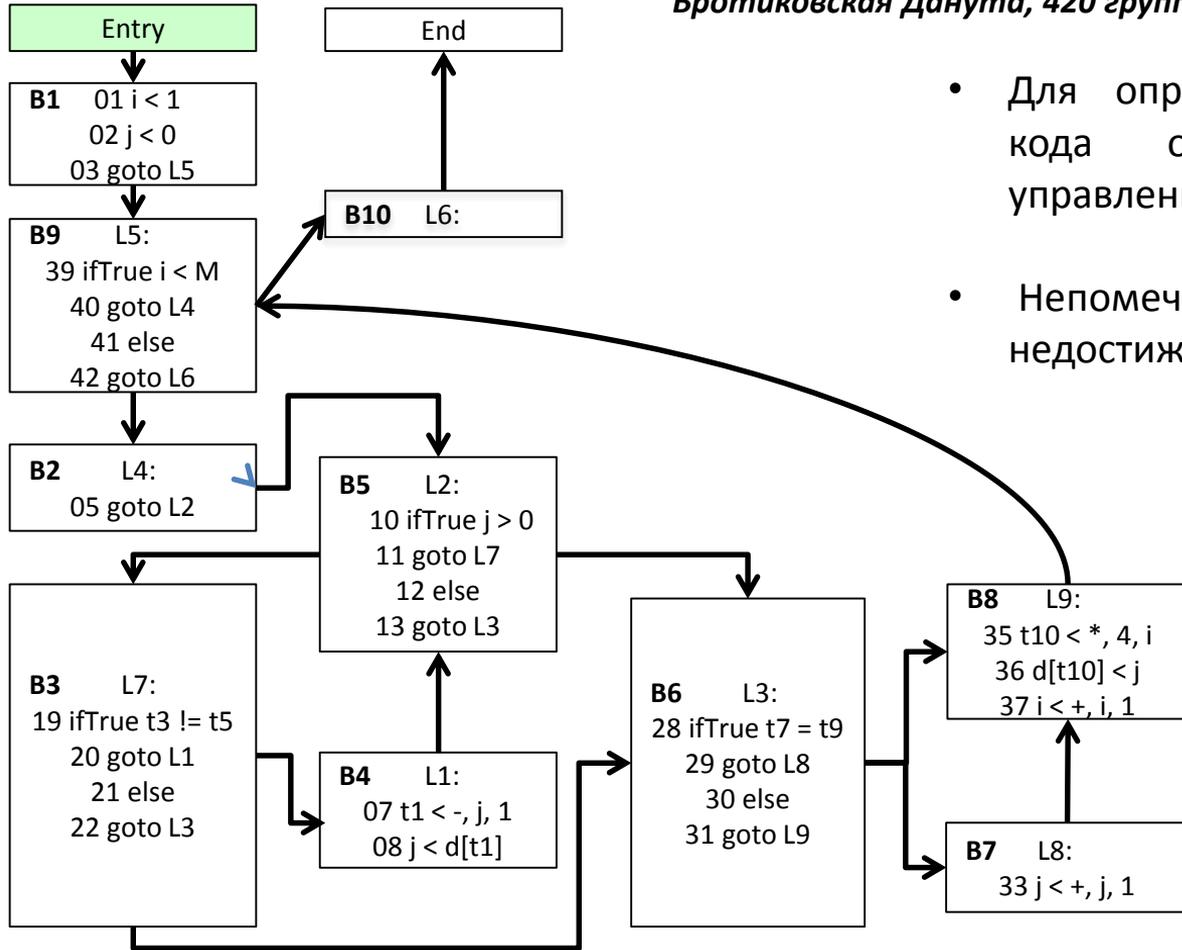
Результат удаления бесполезного кода



Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода. Удаление недостижимого кода

Бротиковская Данута, 420 группа



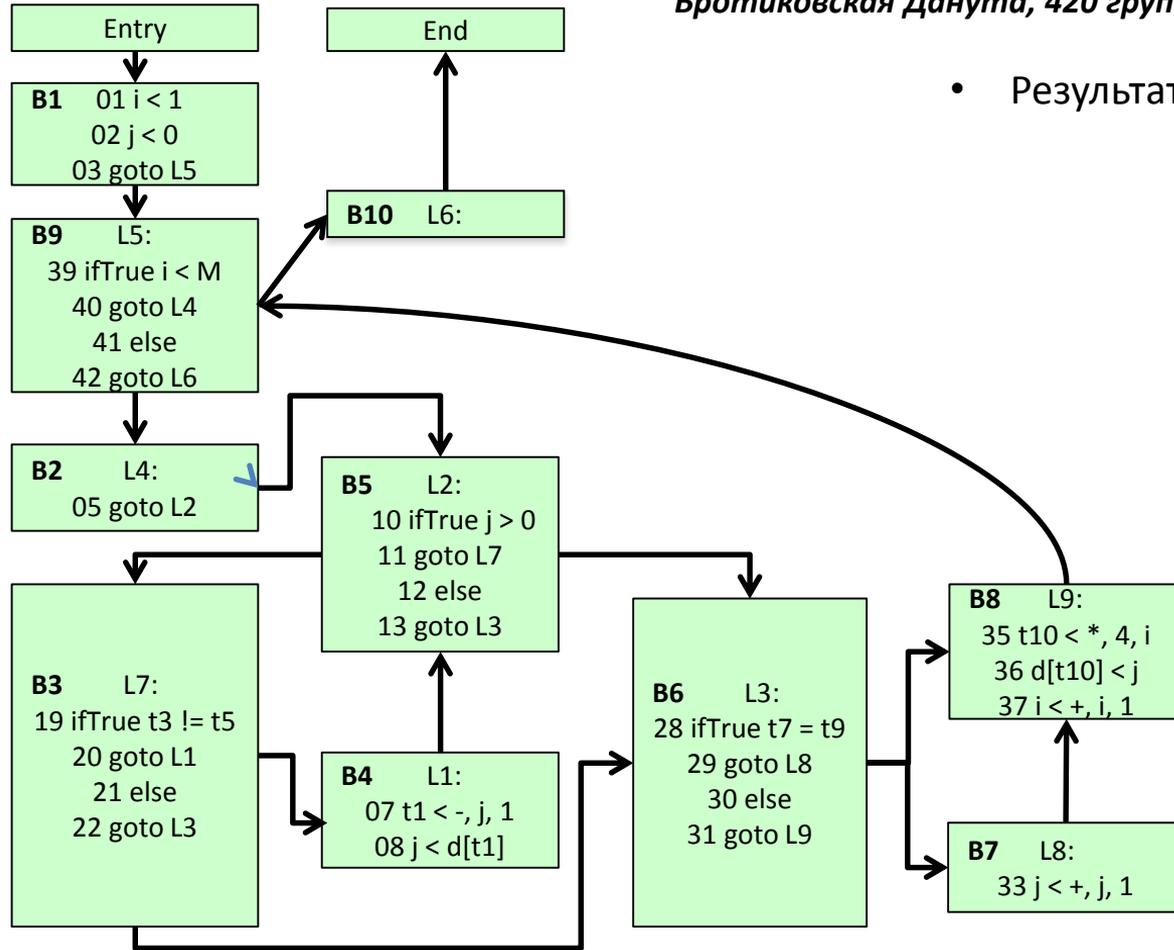
- Для определения недостижимого кода обходим граф потока управления в глубину

- Непомеченные блоки - недостижимы

Домашнее задание №4 по курсу “Конструирование компиляторов”

Удаление бесполезного кода. Удаление недостижимого кода

Бротиковская Данута, 420 группа



- Результат – все блоки достижимы