3 address

1.0

Generated by Doxygen 1.8.8

Mon Apr 13 2015 00:24:38

Contents

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 2

File Index

2.1 File List

Here is a list of all documented files with brief descriptions:

Chapter 3

Class Documentation

3.1 YYSTYPE::BackpatchList Struct Reference

Collaboration diagram for YYSTYPE::BackpatchList:

The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.2 BackpatchList Struct Reference

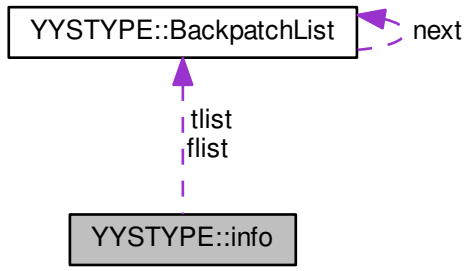
Collaboration diagram for BackpatchList:

The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.3 YYSTYPE::info Struct Reference

Collaboration diagram for YYSTYPE::info:

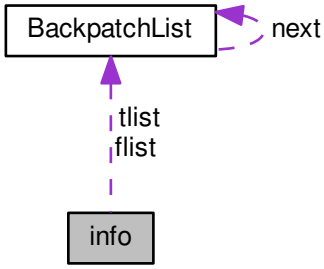


The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.4 info Struct Reference

Collaboration diagram for info:



The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.5 YYSTYPE::marks Struct Reference

The documentation for this struct was generated from the following file:

- yaccrule.cpp

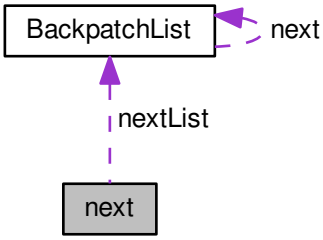
3.6 marks Struct Reference

The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.7 next Struct Reference

Collaboration diagram for next:

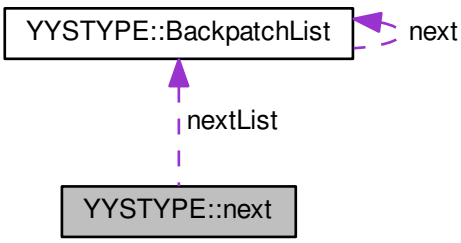


The documentation for this struct was generated from the following file:

- yaccrule.cpp

3.8 YYSTYPE::next Struct Reference

Collaboration diagram for YYSTYPE::next:



The documentation for this struct was generated from the following file:

- yaccrule.cpp

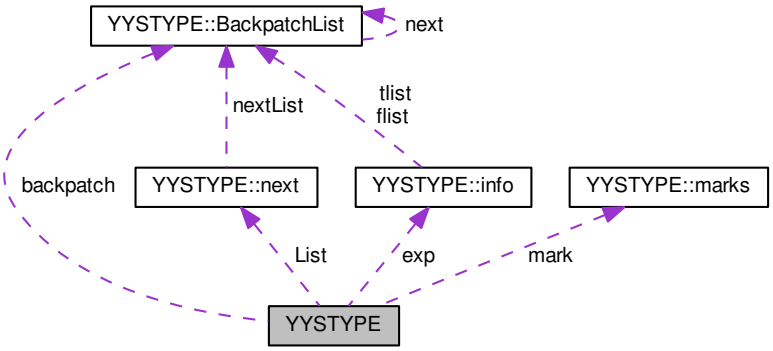
3.9 Quadruple Struct Reference

The documentation for this struct was generated from the following files:

- yaccrule.cpp
- codegener.cpp

3.10 YYSTYPE Union Reference

Collaboration diagram for YYSTYPE:



Classes

- struct BackpatchList
- struct info
- struct marks
- struct next

The documentation for this union was generated from the following file:

- yaccrule.cpp

Chapter 4

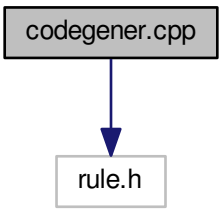
File Documentation

4.1 codegener.cpp File Reference

This file includes grammar rule and their semantic action(s).

```
#include "rule.h"
```

Include dependency graph for codegener.cpp:



Classes

- struct Quadruple

Typedefs

- typedef YYSTYPE::BackpatchList patchList

  generate new temporary variable name

Functions

- void printCode ()

  print code to the console after completion of parsing and store in file name 'output.txt' for future use
- void gencode (const char ∗result, const char ∗addr1, const char ∗op, const char ∗addr2)

  Generate 3 address code and store in Quadrule table for binary expression e.g.
- void gencode (const char ∗result, const char ∗unop, const char ∗addr1)

Generate 3 address code and store in Quadrule table for unary expression e.g.

- void gencode (const char ∗result, const char ∗addr1, const char ∗op, const char ∗addr2, const char ∗addr3, int label)

  Generate 3 address code and store in Quadrule table for conditional jump.
- void gencode (const char ∗result, const char ∗addr1)

  Generate 3 address code and store in Quadrule table for assignement e.g.
- void gencode (const char ∗result, int label)

  Generate 3 address code and store in Quadrule table for goto target e.g.

4.1.1 Detailed Description

This file includes grammar rule and their semantic action(s).

Version

1.0

4.1.2 Typedef Documentation

4.1.2.1 typedef YYSTYPE::BackpatchList patchList

generate new temporary variable name

Parameters

void	None

Returns

newtemp char∗

4.1.3 Function Documentation

4.1.3.1 void printCode ( )

print code to the console after completion of parsing and store in file name 'output.txt' for future use

Parameters

void	None

Returns

void None

4.1.3.2 void gencode (const char ∗ *result,* const char ∗ *addr1,* const char ∗ *op,* const char ∗ *addr2*)

Generate 3 address code and store in Quadrule table for binary expression e.g.

a=t0+b;

Parameters

result	char*
address1	char*
binary_operator	char*
address2	char*

Returns

void None

4.1.3.3 void gencode (const char * *result*, const char * *unop*, const char * *addr1*)

Generate 3 address code and store in Quadrule table for unary expression e.g.

a = -b;

Parameters

result	char*
address1	char*
unary_operator	char*

Returns

void None

4.1.3.4 void gencode (const char * *result*, const char * *addr1*, const char * *op*, const char * *addr2*, const char * *addr3*, int *label*)

Generate 3 address code and store in Quadrule table for conditional jump.

Parameters

if	char*
address1	char*
relational_operator	char*
address2	char*
goto	char*
jump_instruction_number	int

Returns

void None

4.1.3.5 void gencode (const char * *result*, const char * *addr1*)

Generate 3 address code and store in Quadrule table for assignement e.g.

a = t0;

Parameters

result	char*
address2	char*

Returns

void None

4.1.3.6 void gencode (const char * *result*, int *label*)

Generate 3 address code and store in Quadrule table for goto target e.g.

'goto' -1

Parameters

goto	char*
jump_↩ instruction_↩ number	int

Returns

void None

Index