Token Object Garbage Collection (ANTLR Version 1.32)

oblem:

memory leak—token objects don’t get deleted.

itial solution:

`deleteTokens()` message to `ANTLRTokenBuffer` to delete
oken objects as they “fall off” the edge of the buffer.

user actions may point to deleted objects! (bad) onsidered:

ANTLRTokenPtr reference objects that deleted the token objects instead of the token buffer.

Requires serious C++ hacks (what’s new) and major ANTLR surgery.

ew solution (R. Quong):

Have separate token object “spaces”: token buffer space and user-action space.

DLG creates tokens and jams them into token buffer. Token buffer is responsible for deleting the token objects.

Any labeled token reference (e.g., a:ID) actually made a copy of the token object in the token buffer. The parser is responsible for deleting the duplicates; easily done now with an ANTLRTokenPtr reference object.

etails of new C++ interface

Use `deleteTokens()` message (token buffer will delete all tokens coming from lexer).

'`a:ID`" results in "`a = LT(1)->dup()`"; hence, all token bjects must respond to `dup()` if "token copying" is nabled.

use `ANTLRCommonRefCountToken` (or a similarly onstructed class)

Define `ANTLRTokenPtr` (use predefined `#include`)

Default

- `deleteTokens()` (in `ParserBlackBox`)
- copy tokens

Example 1—delete all tokens

```
 Delete all created tokens automagically

ragma dup-labeled-tokens // default

pedef ANTLRCommonRefCountToken ANTLRToken;
nclude "ATokPtr.h" // define ANTLRTokenPtr ref count
nclude "ATokPtr.C" // compile or link in once

nclude "PBlackBox.h"
nclude "DLGLexer.h"

in()

  ParserBlackBox<DLGLexer,Expr,ANTLRToken> p(stdin);
  p.parser()->deleteTokens(); // default
  p.parser()->start_rule();
```

Example 2—delete only dup’d tokens

```
 Delete all duplicated tokens, but not tokens
 created from scanner, however.

ragma dup-labeled-tokens // default

pedef ANTLRCommonRefCountToken ANTLRToken;
nclude "ATokPtr.h" // define ANTLRTokenPtr ref count
nclude "ATokPtr.C" // compile or link in once

nclude "PBlackBox.h"
nclude "DLGLexer.h"

in()

  ParserBlackBox<DLGLexer,Expr,ANTLRToken> p(stdin);
  p.parser()->noDeleteTokens();
  p.parser()->start_rule();
```

Example 3—delete no tokens

```
 Don’t delete tokens from scanner and don’t
 duplicate token objects in parser.

ragma no-dup-labeled-tokens

pedef ANTLRCommonToken ANTLRToken;

nclude "PBlackBox.h"
nclude "DLGLexer.h"

in()

  ParserBlackBox<DLGLexer,Expr,ANTLRToken> p(stdin);
  p.parser()->noDeleteTokens();
  p.parser()->start_rule();
```