

```

Project I
backend-x86.[h.c]
messages (message module)
error ( )
warning ( )
bug ( )

new-identifier :
  new-identifier-1 { $$ = st
                    ;
new-identifier-1 :
  LEX-ID { $$ = $1; }
  |
  ...
  ;
identifier :
  LEX-ID { $$ = stentid($1); }
  ;
typename :
  TYPE
  LEX-ID { check_if($1, stentid($1)); }
  ;
  // checks that this id
  // is installed in the
  // symbol as a typename

gramtypes.txt

type-definition-part
  LEX-STATE type-definition-list semi
  { resolve all pointer types }
  type module
  syntax module

type-defn
  newid 'a' TYPE
  { make a new STID
  { TYPENAME
  { install in symbol table
  { include TYPE info
  { $2 }

typedenoter :
  typename / a default x
  | typedenoter-d
  ;
typedenoter-d :
  new-or-type // all
  ; // default
  // actions

subrange-type
  construct LEX-STATE constants
  // e.g. 3..7
  { check_subrange($1, $3); }
  { checks that $1 ≤ $3
  { calls ty-build-subrange
  { to return the type. }

new_pointer_type :
  pointer-cha pointer-domain-type
  { $$ = $2; }

pointer-domain-type :
  new-identifier { ty-build_ptr($1, $2); }
  | new-proc-and-type { ty-build_ptr($1, $2); }
  ;

ty-build_ptr(STID, TYPE)

new-proc-type :
  LEX-PROC optional-param-list
  { ty-build_ptr($1, $2); }
  LEX-FCN optional-param-list function-type
  { ... }
  ;

f1 = function { Integer }
f2 = function { real } Integer.

optional-param-list :
  / empty x { $$ = NULL; }
  | ' (' proc-and-type ' ) '
  { $$ = construct-param($2); }
  ;

proc-and-type :
  proc-param-parameter // default
  | n-list // construct-param
  ;

proc-param-parameter :
  id-list : typename
  LEX-STATE id-list : typename
  { setup two procedures that
  { as semanticos
  function { ty-build_ptr($1, $2); }

```

Mar 13-11:02 AM