NYACC C99 Munge Module

Matt Wette
March 2017

Introduction

The sxml parse tree can be used to provide autocoding via the `(nyacc lang c99 munge)` module. For example, start with the following C code

```
typedef const char *string_t;
extern string_t cmds[10];
```

The nyacc output (call it `the-tree`) for this will be

```
(trans-unit
  (decl (decl-spec-list
          (stor-spec (typedef))
          (type-qual "const")
          (type-spec (fixed-type "char")))
        (init-declr-list
          (init-declr
            (ptr-declr (pointer) (ident "string_t")))))
  (decl (decl-spec-list
          (stor-spec (extern))
          (type-spec (typename "string_t")))
        (init-declr-list
          (init-declr
           (array-of (ident "cmds") (p-expr (fixed "10")))))))
```

If we feed `the-tree` into `tree->udict` and use `assoc-ref` to lookup `"cmds"` we get

```
(udecl (decl-spec-list
         (stor-spec (extern))
         (type-spec (typename "string_t")))
       (init-declr
         (array-of (ident "cmds") (p-expr (fixed "10")))))
```

Now take this and feed into `expand-decl-typerefs` to get

```
(udecl (decl-spec-list
         (stor-spec (extern))
         (type-qual "const")
         (type-spec (fixed-type "char")))
       (init-declr
         (ptr-declr
           (pointer)
           (array-of (ident "cmds") (p-expr (fixed "10"))))))
```

which, when fed through the C99 pretty-printer, generates

```
extern const char *cmds[10];
```

Since the NYACC C99 parser captures some comments, these can be preserved in the above procedure.

The Util2 (or Munge) Module

Declarations must have one of

- declarators

  ```
  int foo;
  ```

- struct or union reference

  ```
  struct foo;
  ```

- enum value

  ```
  enum { FOO = 1 };
  ```

We provide the following:

`udict-ref udict name`
: Lookup name in udict. This is like assoc-ref but will check through enums also.

`udict-struct-ref udict name`
: Lookup `struct name`.

`udict-union-ref udict name`
: Lookup `union name`.

`udict-enum-ref udict name`
: Lookup `enum name`.

From Util2