

1 Actividade 0x08 - T.A.D. *Symbol Table*

Nesta actividade vamos desenvolver um *Tipo Abstracto de Dados* para a estrutura de base da Análise Semântica, que é a *Symbol Table* (ST). Vamos focar-nos apenas no necessário para a linguagem Yal.

2 Interface

A ST tem uma interface bastante simples, consistindo em apenas 5 funções:

```
1 typedef struct st_data_ *ST_Data;
2
3 /* Insere um novo nome (ID) na ST */
4 int ST_insert(char *id, ST_Data data);
5
6 /* Devolve os dados de um ID (ou NULL, se não existe) */
7 ST_Data ST_lookup(char *id);
8
9 /* Devolve os dados de um ID (apenas no scope actual) */
10 ST_Data ST_lookup_local(char *id);
11
12 /* Cria um novo âmbito (scope ou environment) local - à entrada
    na função */
13 int ST_new_scope();
14
15 /* Descarta o último scope - à saída da função */
16 int ST_discard();
```

3 Implementação

A ST é, acima de tudo, uma estrutura do tipo dicionário¹. Os alunos devem escolher uma estrutura de base adequada² e implementá-la.

A estrutura base para os valores a guardar na ST terá um formato do tipo:

```
1 struct st_data_ {
2     enum {ST_VAR, ST_FUNC, ST_TYPE} kind;
3
4     union {
5         struct {
6             _Type yatype; /* _Type pode ser o tipo definido na análise
7                sintáctica */
8             enum {VARloc, VARarg} kind;
9             /* . . . mais tarde, precisaremos de mais info */
10            } var;
11            struct {
12                _Type yatype; /* tipo de retorno */
13                _ArgTypes arg; /* "lista" de tipos dos argumentos, por
14                ordem */
15                /* . . . mais tarde, precisaremos de mais info */
16            } func;
17            _Type type; /* para este caso só precisamos do tipo destino */
18        } u;
19    };
20 }
```

Podemos sempre alterar esta `struct` ao nosso gosto, desde que permita:

- Guardar toda a informação *importante* sobre variáveis, funções e novos tipos;
- Ser aumentada com informações que só vamos “descobrir” mais à frente no processo de compilação.

4 Exercício

4.1 Código a implementar

1. Implementar a estrutura de base para a ST (*hash-table*, *b-tree*, *b+-tree*, etc.).
2. Implementar o tipo `struct st_data_`.
3. Implementar as funções necessárias ao funcionamento da ST.

¹associa **chaves** (IDs) a **valores**

²e.g., hash-table, b-tree, ...