Collection Subset Demo

Let's look at a subset of the language, related to a collection:

collection – Defines a set of (**unique**) str strings

{ "….", "….", … }   -- Collection literal (**{} – Empty collection**)

collection   <var> *[, <var> …]*   -- Define one or more collection variables

<collection> + <collection>	Union	collection

<expression> :==   <literal>
                          <var>
output "<string>" *[<expression>]*; ➔

          output "<string>" *[<collection var> | <collection literal>]*;

Subset CFG for Demo:

Prog: SENTENCE_LIST

SENTENCE_LIST: SENTENCE

          | SENTENCE_LIST SENTENCE

SENTENCE: collection VAR ;

          | VAR = COLLECTION;

          | VAR = VAR + COLLECTION ;

          | output STRING COLLECTION ;

COLLECTION: VAR

          | {}

          | { STRING_LIST }

VAR: t_ID

STRING_LIST   STRING_LIST, t_STRING

          | t_STRING

Memory Representation for collection

'@' separated string (char*) preceded by "
For example: {"Hello","World"} ➔ "Hello@World\0

          In C language it is represented as: **"\"Hello@World"**

Symbol Table

```
typedef enum {Collection=1, Set} varType;

struct {
    char *name
    varType typ;
} SymbolTable[*];
```