**EBNF of MUA**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **name** | = | ‘"’, **name\_wo\_quo** |
| **name\_wo\_quo** | = | **letter**, { **letter** | **digit** | “\_”};  **(\* name without quotation mark \*)** |
| **literal** | = | **number** | **literal\_word** | **bool** | **list**; |
| **number** | = | [“-”], **digit**, {**digit**}, [“.”, **digit**, {**digit**}]; |
| **const\_word** | = | ‘"’, **word**; |
| **word** | = | {**?all non-spacing character?**}; |
| **bool** | = | “true” | “false”; |
| **list** | = | “[”, [**literal**], {“ ”, **literal**}, “]”; |
| **digit** | = | **?0..9?**; |
| **letter** | = | **?A..Za..z?**; |
| **program** | = | {**sentence**}; |
| **sentence** | = | **expression** | **make\_func** | **return\_stmt**; |
| **expression** | = | “make”, **name**, **expression** | “thing”, **name** |
|  | | | “:”, **name\_wo\_quo** | “print”, **expression** |
|  | | | “erase”, **name** | “run”, **list** |
|  | | | “if”, **bool\_expr**, **list**, **list** |
|  | | | **name**, **arglist** **(\* func call \*)** | “export”, **name** |
|  | | | **num\_expr** | **bool\_expr** | **list\_expr** | **word\_expr**; |
| **arglist** | = | [**expression**], {“ ”, **expression**}; |
| **num\_expr** | = | **number** | “thing”, **name** | “:”, **name\_wo\_quo** |
|  | | | **basic\_op**, **num\_expr**, **num\_expr** |
|  | | | “random”, **num\_expr** | “int”, **num\_expr** |
|  | | | “sqrt”, **num\_expr** | “read”; |
| **basic\_op** | = | “add” | “sub” | “mul” | “div” | “mod”; |
| **bool\_expr** | = | **bool** | “thing”, **name** | “:”, **name\_wo\_quo** |
|  | | | “load”, **word\_expr** | “erall” |
|  | | | (“eq”|“gt”|“lt”), **num\_expr**, **num\_expr** |
|  | | | (“eq”|“gt”|“lt”), **word\_expr**, **word\_expr** |
|  | | | (“and”|“or”), **bool\_expr**, **bool\_expr** |
|  | | | (“isnumber”|“isword”|“islist”|“isbool”), **expression** |
|  | | | “isempty”, (**word\_expr**|**list\_expr**) |
|  | | | “isname”, **expression** | “not”, **bool\_expr;** |
| **list\_expr** | = | **list** | “thing”, **name** | “:”, **name\_wo\_quo** |
|  | | | “sentence”, **expression**, **expression** |
|  | | | “list”, **expression**, **expression** |
|  | | | “join”, **list\_expr**, **expression** |
|  | | | **extract\_op**, **list\_expr** | “readlist” | “poall”; |
| **extract\_op** | = | “first” | “last” | “butfirst” | “butlast”; |
| **word\_expr** | = | **word** | “thing”, **name** | “:”, **name\_wo\_quo** |“read” |
|  | | | “word”, **word\_expr**, (**word\_expr**|**num\_expr**|**bool\_expr**) |
|  | | | “save”, **word\_expr** | **extract\_op**, **word\_expr**; |
| **make\_func** | = | “make”, **name**, “[”, **list**, **list**, “]”; |
| **return\_stmt** | = | “return”, **value**; |

尝试写了一个EBNF 😋 但是还有很多地方没搞清楚，本身对EBNF也没什么接触，所以还有很多地方需要修正 😜

为了简洁，EBNF中许多地方省去了空格。

还没有搞清楚的语法问题用 黄底 标注，语义问题用 灰底 标注（目前暂时没有了）。

Update 1 (2021.9.24):

* **name**并不被视作一个表达式，取其值需要使用**thing**或 **:** 操作。据此更新了**expression**和 **\*\_expr**的部分规则
* 解决了**make**和**erase**的语义问题

Update 2 (2021.9.24):

* **isname**的操作数可以是操作的返回值
* 确认了**thing**和 **:** 的语法（建议规则改为thing <word> 和 :<name>）
* 确认了一些关于list的语法。核心是“数据与程序不区分”（参见issue#4中范学长的解释）。据此更改了相关规则，删除了**stmt\_list**

Update 3 (2021.9.24):

* 确认了函数调用的语法