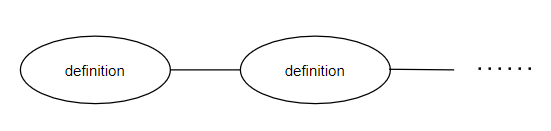
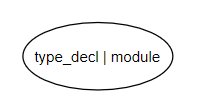
**抽象语法树设计：**

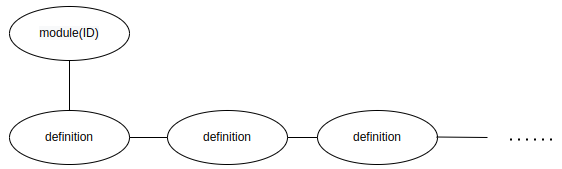
1、specification -> definition { definition}



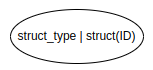
2、definiton -> type\_decl“**;**”| module “**;**”



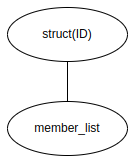
3、module -> “**module**”ID “**{**” definition { definition } “**}**”



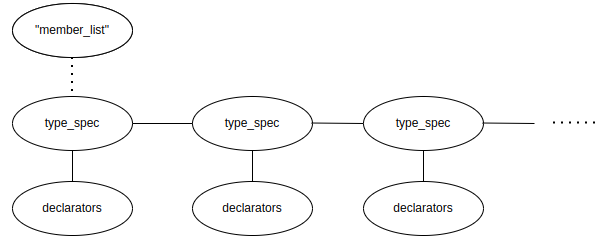
4、type\_decl -> struct\_type | “**struct**” ID



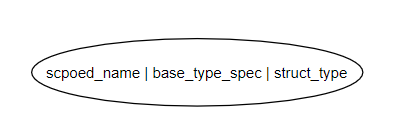
5、struct\_type->**“struct”** **ID “{”** member\_list **“}”**



6、member\_list-> { type\_spec declarators **“;”** }



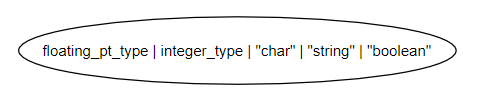
7、type\_spec -> scoped\_name | base\_type\_spec | struct\_type



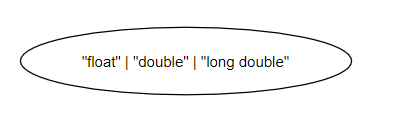
8、scoped\_name -> [**“::”**] ID {**“::”** ID}



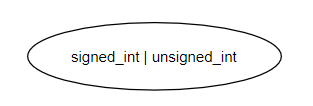
9、base\_type\_spec->floating\_pt\_type|integer\_type|**“char”**|**“string”**|**“boolean”**



10、floating\_pt\_type -> **“float”** | **“double”** | **“long double”**



11、integer\_type -> signed\_int | unsigned\_int

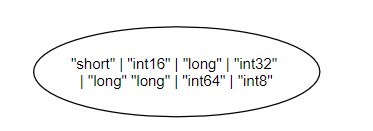


12、signed\_int->(**“short”**|**“int16”**)

|(**“long”**|**“int32”**)

|(**“long”** **“long”**|**“int64”**)

|**“int8”**

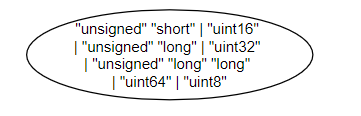


13、unsigned\_int -> (**“unsigned”“short”**| **“uint16”**)

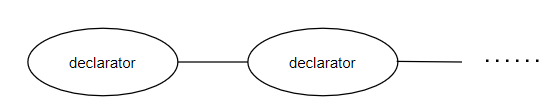
| (**“unsigned”“long”**| **“uint32”**)

| (**“unsigned” “long” “long”** | **“uint64”**)

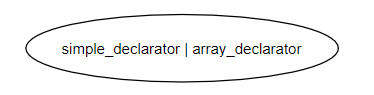
| **“uint8”**



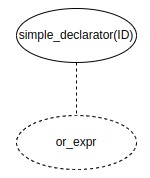
14、declarators -> declarator {**“,”** declarator }



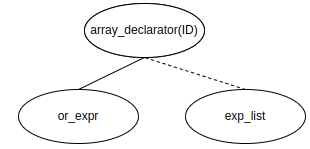
15、declarator -> simple\_declarator | array\_declarator



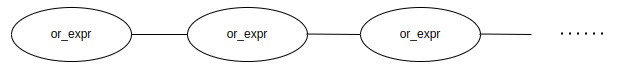
16、simple\_declarator -> ID [**“=”** or\_expr]



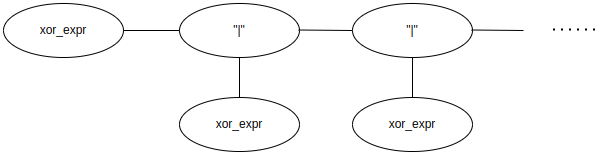
17、array\_declarator -> ID “[” or\_expr “]” [**“=”** exp\_list ]



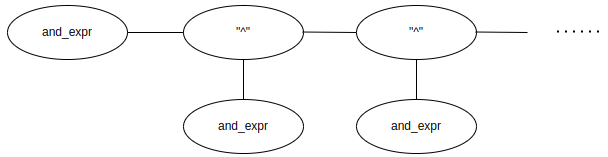
18、exp\_list -> **“[”** or\_expr { **“,”**or\_expr } **“]”**



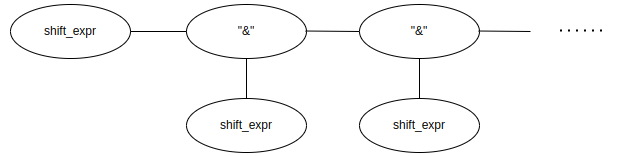
19、or\_expr -> xor\_expr {**“|”** xor\_expr }



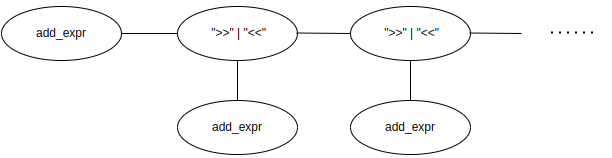
20、xor\_expr -> and\_expr {**“^”** and\_expr }



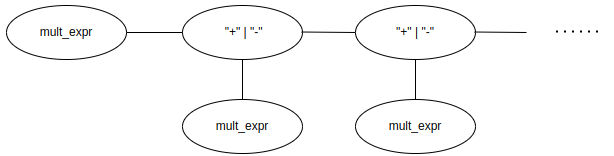
21、and\_expr -> shift\_expr {**“&”**shift\_expr }



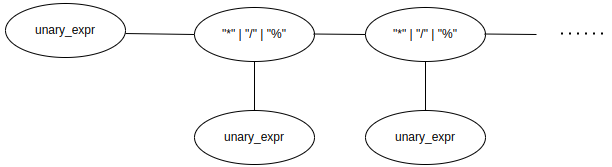
22、shift\_expr -> add\_expr { (**“≥≥”** | **“<<”**) add\_expr }



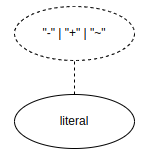
23、add\_expr -> mult\_expr { (**“+”** | **“-”**) mult\_expr }



24、mult\_expr -> unary\_expr { (**“\*”** |**“/”**|**“%”**) unary\_expr }

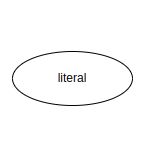


25、unary\_expr -> [**“-”**| **“+”** | **“~”**] literal

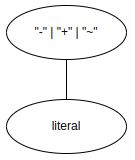


**或者改成**：

不存在 “- | + | ~” 时为：

，

存在“- | + | ~”时为：



26、literal -> **INTEGER** | **FLOATING\_PT** | **CHAR** | **STRING** | **BOOLEAN**

