valid\_program : graph\_decl

graph\_decl : type\_link graph\_stmt\_list

type\_link : **use** string **;**

graph\_stmt\_list : graph\_stmt **;**

| graph\_stmt\_list graph stmt **;**

graph\_stmt : label\_app

| node\_dec

| arc\_ dec

| list\_dec

|prim\_dec

| if\_stmt

| while\_stmt

| expr

label\_app : id **:** node\_dec

node\_dec : **@** id attr\_list

| **@** id

arc\_dec : id **->** id attr\_list

| id **->** id

list\_dec : **list of** type id

| **list of** type id **= [** attr\_list **]**

type : ptype

prim\_dec : type id

| type id **=** expr

while\_stmt : **while** **(** expr **) {** graph\_stmt\_list **}**

if\_stmt: **if** **(** expr **) {** graph\_stmt\_list **}**

attr\_list : expr

| attr\_list **,** expr

expr : **(** expr **)**

| **(** type **)** expr

| - expr

| expr **>** expr

| expr **>=** expr

| expr **<** expr

| expr **<=** expr

| expr **!=** expr

| expr **==** expr

| expr **+** expr

| expr **-** expr

| expr **\*** expr

| expr **/** expr

| expr **%** expr

| id **.** id

| **Graph** **.** id

| assignment

| access

| id

| func\_call

| pvalue

access : id **[** expr **]**

assignment : access = expr

| id = expr

id : ID

ptype : **int**

| **float**

| **string**

| Arc

| Node

pvalue : int

| float

| string