

定值到达分析用户手册

撰写人：张书博

前期准备：

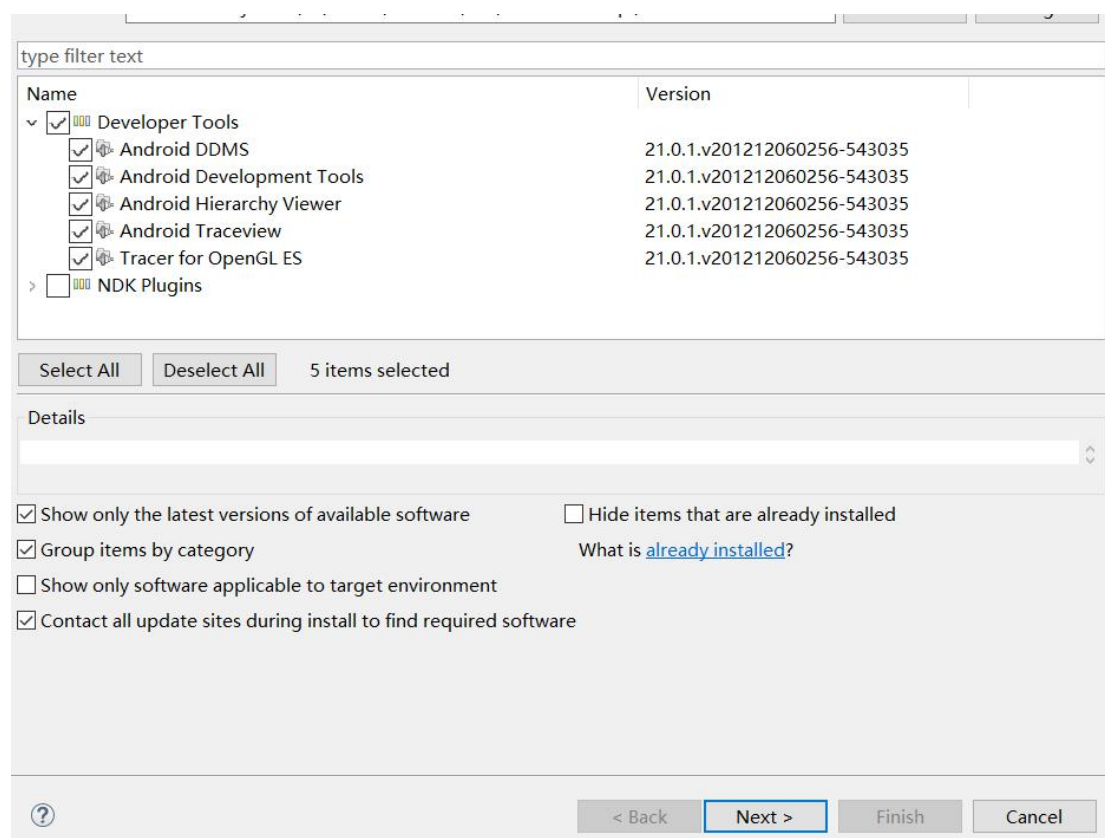
1. 下载一个 Eclipse JAVA EE 并配置 Eclipse JDT

关于 Eclipse JAVA EE 的下载，直接去 Eclipse 的官网下载最新版本的即可。

下载链接：

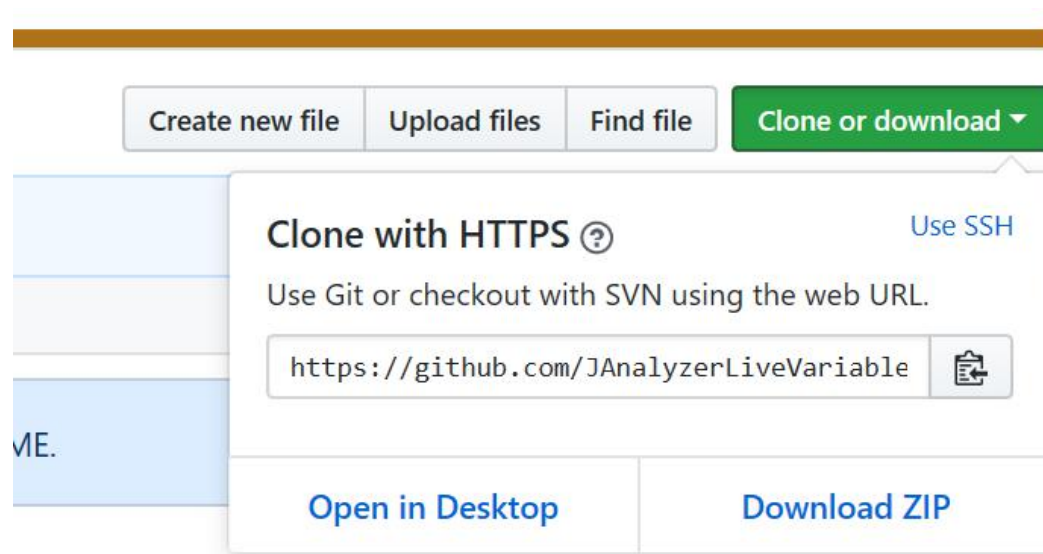
想要启动 JAnalyser，是一定要先完成 JDT 的配置，其实 JDT 很少有单独配置的，都是完成 ADT 配置的同时会自动完成 JDT 的配置，所以可以下载一份 ADT-21.0.1.zip，然后按照网上的教程去做，这里要注意的是添加的时候要添加 zip 文件（不用添加解压出来的文件），然后名字就命名为 ADT-21.0.1。

添加完 ADT-21.0.1.zip 文件以后，直接点 OK，然后选定 Developer Tools 然后直接一直点 Next 就可以自动配置完毕 ADT。



2. 从 github 上面获取源码

直接进行 `git clone https://github.com/JAnalyzerLiveVariableAnalysis/JA-ver2` 操作，
或者是在 <https://github.com/JAnalyzerLiveVariableAnalysis/JA-ver2> 点击 Download 获取源码。



3. windows 下安装 GVEDiv

可以进入官网下载 windows 的版本然后安装。

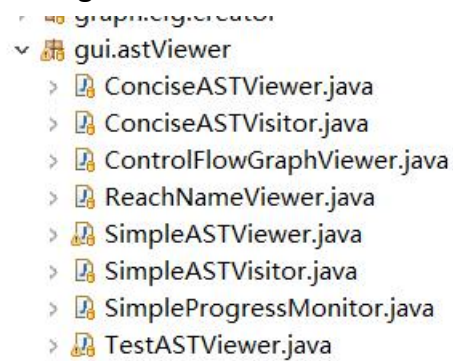
<https://graphviz.gitlab.io/download/>

程序运行步骤：(windows 下)

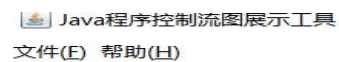
1. 打开 Eclipse，然后选择对应项目。对应项目的结构如下：

- > JRE System Library [jdk1.8.0_45]
- > (default package) [JA-ver2 master]
- > analyzer.dataTable
- > analyzer.nullCheck
- > analyzer.qualitas
- > graph.basic
- > graph.cfg
- > graph.cfg.analyzer
- > graph.cfg.creator
- > gui.astViewer
- > gui.toolkit
- > nameTable
- > nameTable.creator
- > nameTable.filter
- > nameTable.nameDefinition
- > nameTable.nameReference
- > nameTable.nameReference.referenceGroup
- > nameTable.nameScope
- > nameTable.visitor
- > sourceCodeAST
- > util
- > Referenced Libraries

2. 点击 gui.astViewer



右键 `TestASTViewer.java`，然后选定运行。
出现如下 GUI。



3. 点击文件，选择定值到达分析，然后选定一个.java 文件进行定值到达分析。

效果图如下：

右边配备了详细的文字说明，从字面意思去理解分析的结果就可以。（分析是以可执行点为单位，每个可执行点的名字定义都会被分析（显示定义名字的定值表达式），还涉及到名字定义的根源定值的分析等）



4.关于定值到达分析还有一个选项是最大定值到达分析。

注意，最大定值到达分析的含义就是，选定一个.java 文件，分析的是这个.java 文件的父目录下的所有.java 文件中的最长（大）的一个方法定义（从父目录的所有的.java 文件中选择一个最大的方法定义，最大的方法定义属于一个.java 文件，一个.java 文件可能有很多方法定义）。所以选定的.java 文件有可能不会被分析。

最大定值到达分析的效果图如下：



关于可视化分析

1.dot 文件产生的位置

可视化分析是要借助 GVEdit 工具来完成的，可以通过让 GVEdit 要读取.dot 文件从而产生出对应.java 文件的定值到达分析的图片。

注意：运行一次定值到达分析，就会产生一个.dot 文件在电脑中。至于产生的位置在哪里，取决于 TestASTViewer.java 中的 OutputStream os 的配置，如下图所示：

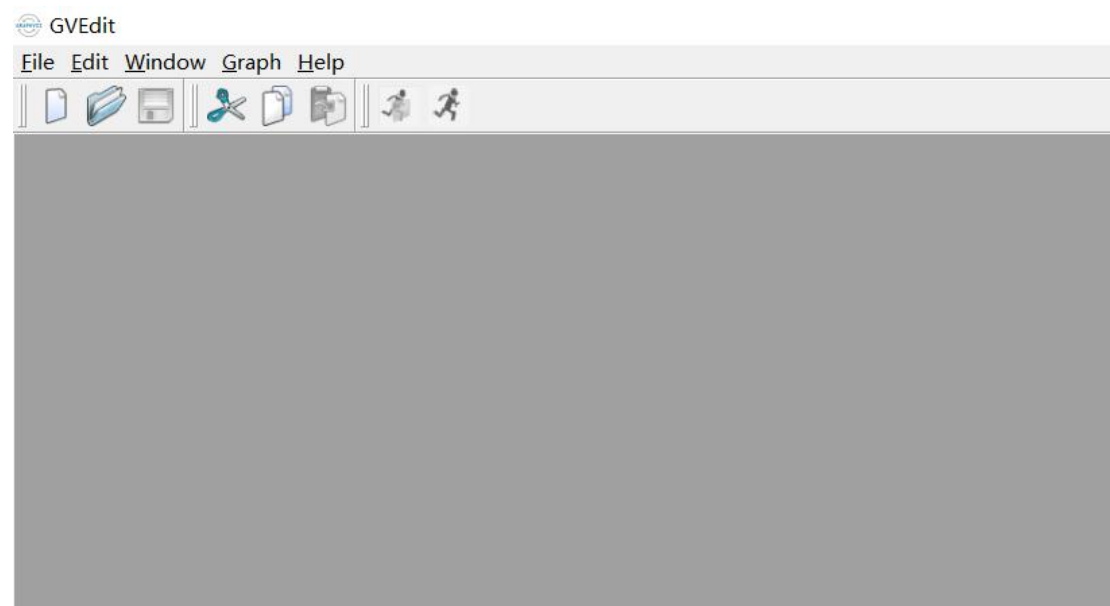
目前这么设置，点击运行一次定值到达分析产生的.dot 就是在 C:\\Java\\test2.dot 位置。

```
public DemoMenuCreator(Container place, JFrame topLevelFrame) {  
    >> this.place = place;  
    >> this.topLevelFrame = topLevelFrame;  
    >> fileOpener = new FileChooserAndOpener(topLevelFrame);  
    >> try {  
    >>     OutputStream os = new FileOutputStream("C:\\Java\\test2.dot");  
    >>     output = new PrintWriter(os);  
    >> } catch (Exception ex) {  
    >>     ex.printStackTrace();  
    >> }  
}
```

2.GVEdit 的运行

下面讲解如何运行：

打开 GVEdit 以后有如下界面：

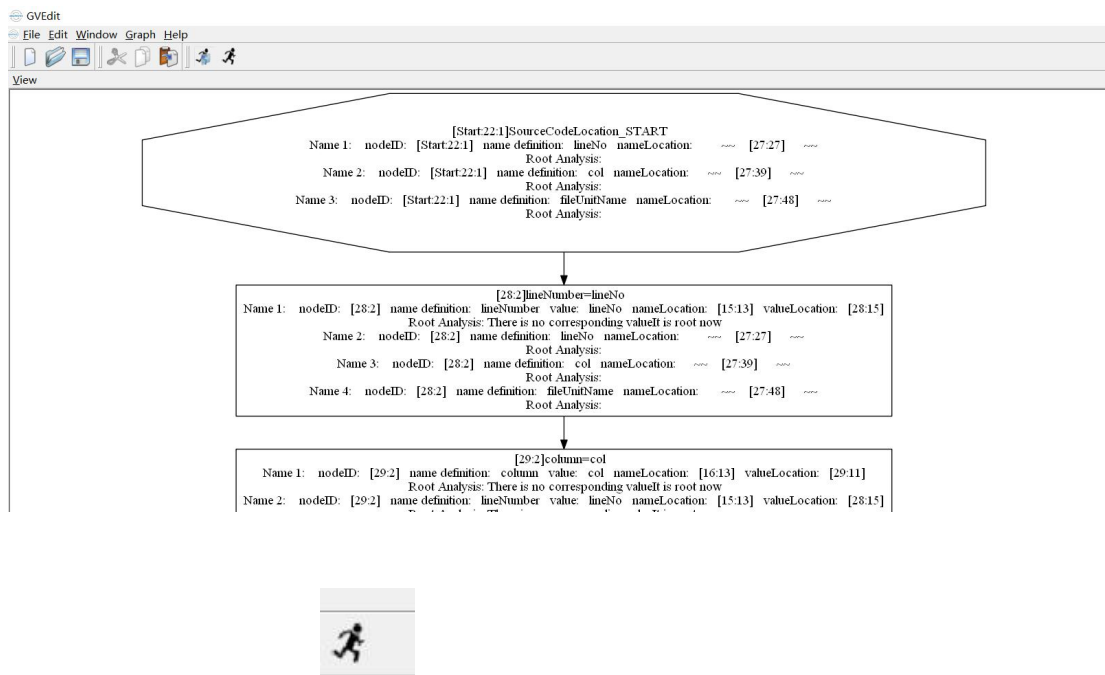



点击 File，点击 open，选择之前定值到达分析生成的.dot。

```
GVEdit - [test2.dot]
File Edit Window Graph Help

digraph SourceCodeLocation_22_1_SourceCodeLocation_java {
    SourceCodeLocation_START[label = "[Start:22:1]SourceCodeLocation_START
Name 1: nodeID: [Start:22:1] name definition: lineNo nameLocation: ~ [27:27] ~
Root Analysis:
Name 2: nodeID: [Start:22:1] name definition: col nameLocation: ~ [27:39] ~
Root Analysis:
Name 3: nodeID: [Start:22:1] name definition: fileUnitName nameLocation: ~ [27:48] ~
Root Analysis:
    ", shape = octagon]
    node28_2[label = "[28:2]lineNumber=lineNo
Name 1: nodeID: [28:2] name definition: lineNumber value: lineNo nameLocation: [15:13] valueLocation: [28:15]
Root Analysis: There is no corresponding valueIt is root now
Name 2: nodeID: [28:2] name definition: lineNo nameLocation: ~ [27:27] ~
Root Analysis:
Name 3: nodeID: [28:2] name definition: col nameLocation: ~ [27:39] ~
Root Analysis:
Name 4: nodeID: [28:2] name definition: fileUnitName nameLocation: ~ [27:48] ~
Root Analysis:
    ", shape = box]
    node29_2[label = "[29:2]column=col
Name 1: nodeID: [29:2] name definition: column value: col nameLocation: [16:13] valueLocation: [29:11]
Root Analysis: There is no corresponding valueIt is root now
Name 2: nodeID: [29:2] name definition: lineNumber value: lineNo nameLocation: [15:13] valueLocation: [28:15]
Root Analysis: There is no corresponding valueIt is root now
Name 3: nodeID: [29:2] name definition: lineNo nameLocation: ~ [27:27] ~
Root Analysis:
Name 4: nodeID: [29:2] name definition: col nameLocation: ~ [27:39] ~
Root Analysis:
Name 5: nodeID: [29:2] name definition: fileUnitName nameLocation: ~ [27:48] ~
Root Analysis:
    ", shape = box]
    node30_2[label = "[30:2]this.fileUnitName=fileUnitName
Name 1: nodeID: [30:2] name definition: fileUnitName value: fileUnitName nameLocation: [17:16] valueLocation: [30:22]
Root Analysis: There is no corresponding valueIt is root now
Name 2: nodeID: [30:2] name definition: column value: col nameLocation: [16:13] valueLocation: [29:11]
Root Analysis: There is no corresponding valueIt is root now
Name 3: nodeID: [30:2] name definition: lineNumber value: lineNo nameLocation: [15:13] valueLocation: [28:15]
Root Analysis: There is no corresponding valueIt is root now
    }
```

这是一般会自动弹出.dot 对应的 view。



如果没有 view，手动点击  这里即可。

要注意的是，定值到达分析生成的.dot 包括了那个.java 文件中的所有可执行点，生成的 view 只是第一个可执行点的 view。如果想看接下来的，要手动删除第一个 digraph，然后再点击



，这时可以看到第二个可执行点的 view。（每个可执行点以 digraph{}为单位）


```

digraph getColumn_37_1_SourceCodeLocation_java {
    getColumn_START[label = "[Start:37:1]getColumn_START ", shape = octagon]
    node38_2[label = "[38:2]return column ", shape = box]
    getColumn_END[label = "[End:37:1]getColumn_END ", shape = octagon]
    getColumn_START->node38_2
    node38_2->getColumn_END
};

digraph getFileUnitName_41_1_SourceCodeLocation_java {
    getFileUnitName_START[label = "[Start:41:1]getFileUnitName_START ", shape = octagon]
    node42_2[label = "[42:2]return fileUnitName ", shape = box]
    getFileUnitName_END[label = "[End:41:1]getFileUnitName_END ", shape = octagon]
    getFileUnitName_START->node42_2
    node42_2->getFileUnitName_END
};

digraph toString_45_1_SourceCodeLocation_java {
    toString_START[label = "[Start:45:1]toString_START ", shape = octagon]
    node47_2[label = "[47:2]return ' ' + lineNumber + LINE_COLUMN_SPLITTER+ column ", shape = box]
    toString_END[label = "[End:45:1]toString_END ", shape = octagon]
    toString_START->node47_2
    node47_2->toString_END
};

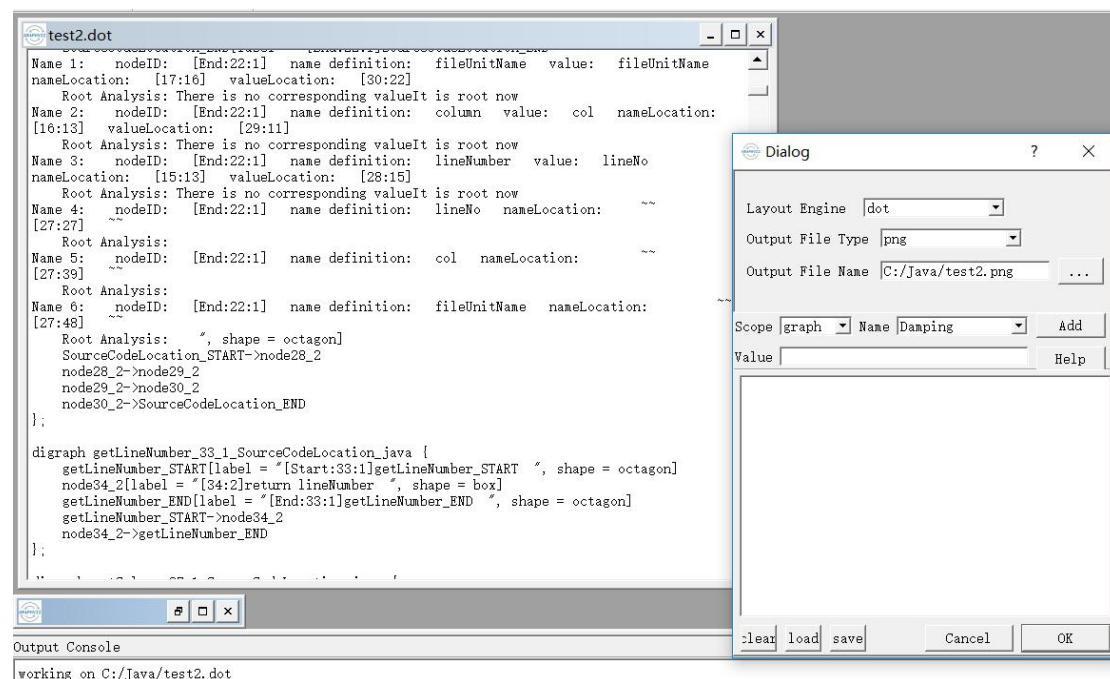
```

3.对应的图片的生成

此外，GVEdit 还有一个强大的功能就是根据.dot 导出图片。即 **view** 作为图片被保存到电脑



中。点击这个即可。点击以后会弹出 **Dialog**，导出什么格式的图片以及导出的图片存放在哪里由自己设定。设定好了点击 **OK** 即可。



最大定值到达分析的图会比较丰富，相比定值到达分析的话。（因为最大定值到达分析的可执行点规模很大，比普通的定值到达分析的第一个可执行点往往要大）

原理是一样的，运行完最大定值到达分析以后，关闭 GVEdit 再重新打开，选择新的.dot。然后导出图片以及在 GVEdit 软件中看 view 的操作都是和之前一样的。最大定值到达分析的图片显然更加复杂。

