

定向覆盖模糊测试工具的设计与实现

毕业设计中期检查

雷尚远

南京邮电大学计算机学院

2023 年 4 月 17 日



① Background

② 研究现状

③ 研究内容

④ 计划进度

⑤ 参考文献

1 Background

Pre-Knowledge
Motivation
Research Status

2 研究现状

3 研究内容

4 计划进度

5 参考文献

1 Background

Pre-Knowledge

Motivation

Research Status

2 研究现状

3 研究内容

4 计划进度

5 参考文献

What Fuzzing is?

Defination

- **Fuzzing** Fuzzing is the execution of the PUT using input(s) sampled from an input space (the “fuzz input space”) that protrudes the expected input space of the PUT[1].
 - PUT: Program Under Test
- **Fuzz testing** Fuzz testing is the use of fuzzing to test if a PUT violates a correctness policy[1].
- **Fuzzer** A fuzzer is a program that performs fuzz testing on a PUT[1].
- **Bug Oracle** A bug oracle is a program, perhaps as part of a fuzzer, that determines whether a given execution of the PUT violates a specific correctness policy[1].
- **Fuzz Configuration** A fuzz configuration of a fuzz algorithm comprises the parameter value(s) that control(s) the fuzz algorithm[1].

What Fuzzing is?

Fuzz Testing

Input: $\mathbb{C}, t_{\text{limit}}$

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4      $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5      $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
6     //  $O_{\text{bug}}$  is embedded in a fuzzer
7      $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
8      $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
9      $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

What Fuzzing is?

Fuzz Testing

```
1 Input:  $\mathbb{C}$ ,  $t_{\text{limit}}$ 
2 Output:  $\mathbb{B}$  // a finite set of bugs
3  $\mathbb{B} \leftarrow \emptyset$ 
4  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
5 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
6      $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
7      $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
8     //  $O_{\text{bug}}$  is embedded in a fuzzer
9      $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
10     $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
11     $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
12 return  $\mathbb{B}$ 
```

- \mathbb{C} : a set of fuzz configurations
- t_{limit} : timeout
- \mathbb{B} : a set of discovered bugs

What Fuzzing is?

Fuzz Testing

Input: $\mathbb{C}, t_{\text{limit}}$

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4      $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5      $tcs \leftarrow \text{InputGen}(\text{conf})$ 
6     //  $O_{\text{bug}}$  is embedded in a fuzzer
7      $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, tcs, O_{\text{bug}})$ 
8      $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

PREPROCESS (\mathbb{C}) $\rightarrow \mathbb{C}$

- Instrumentation
 - grey-box and white-box fuzzers
 - static/dynamic
- Seed Selection
 - weed out potentially redundant configurations
- Seed Trimming
 - reduce the size of seeds
- Preparing a Driver Application
 - library Fuzzing, kernel Fuzzing

What Fuzzing is?

Fuzz Testing

Input: C, t_{limit}

Output: B // a finite set of bugs

```
1  $B \leftarrow \emptyset$ 
2  $C \leftarrow \text{Preprocess}(C)$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(C)$  do
4      $\text{conf} \leftarrow \text{Schedule}(C, t_{\text{elapsed}}, t_{\text{limit}})$ 
5      $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
6     //  $O_{\text{bug}}$  is embedded in a fuzzer
7      $B', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
8      $C \leftarrow \text{ConfUpdate}(C, \text{conf}, \text{execinfos})$ 
9      $B \leftarrow B \cup B'$ 
9 return  $B$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- CONTINUE (C)
→ {True, False}
- Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4      $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5      $tcs \leftarrow \text{InputGen}(\text{conf})$ 
6     //  $O_{\text{bug}}$  is embedded in a fuzzer
7      $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, tcs, O_{\text{bug}})$ 
8      $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
9      $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{CONTINUE}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4    $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5    $tcs \leftarrow \text{InputGen}(\text{conf})$ 
6    $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, tcs, O_{\text{bug}})$ 
7    $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
8    $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{CONTINUE}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4    $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5    $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
   //  $O_{\text{bug}}$  is embedded in a fuzzer
6    $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
7    $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
8    $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{CONTINUE}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4    $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5    $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
   //  $O_{\text{bug}}$  is embedded in a fuzzer
6    $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
7    $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
8    $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{CONTINUE}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4    $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5    $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
6   //  $O_{\text{bug}}$  is embedded in a fuzzer
7    $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
8    $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
9    $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{Continue}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

What Fuzzing is?

Fuzz Testing

Input: \mathbb{C} , t_{limit}

Output: \mathbb{B} // a finite set of bugs

```
1  $\mathbb{B} \leftarrow \emptyset$ 
2  $\mathbb{C} \leftarrow \text{Preprocess}(\mathbb{C})$ 
3 while  $t_{\text{elapsed}} < t_{\text{limit}} \wedge \text{Continue}(\mathbb{C})$  do
4      $\text{conf} \leftarrow \text{Schedule}(\mathbb{C}, t_{\text{elapsed}}, t_{\text{limit}})$ 
5      $\text{tcs} \leftarrow \text{InputGen}(\text{conf})$ 
6     //  $O_{\text{bug}}$  is embedded in a fuzzer
7      $\mathbb{B}', \text{execinfos} \leftarrow \text{InputEval}(\text{conf}, \text{tcs}, O_{\text{bug}})$ 
8      $\mathbb{C} \leftarrow \text{ConfUpdate}(\mathbb{C}, \text{conf}, \text{execinfos})$ 
9      $\mathbb{B} \leftarrow \mathbb{B} \cup \mathbb{B}'$ 
9 return  $\mathbb{B}$ 
```

stop condition

- $t_{\text{elapsed}} < t_{\text{limit}}$
- $\text{CONTINUE}(\mathbb{C}) \rightarrow \{\text{True}, \text{False}\}$
 - Determine whether a new fuzz iteration should occur

1 Background

Pre-Knowledge
Motivation
Research Status

2 研究现状

3 研究内容

4 计划进度

5 参考文献

Why Grey-box Fuzzing ?

classification of fuzzing

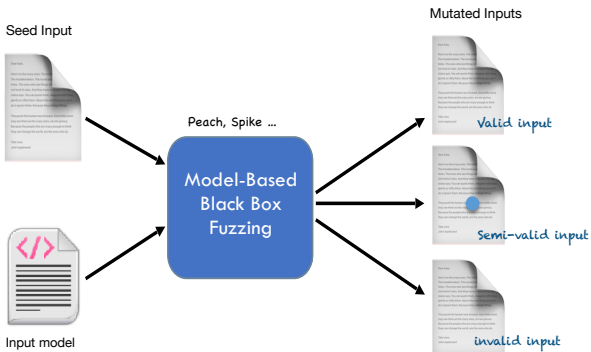
- **Black-box Fuzzing**
 - no program analysis, no feedback
- **White-box Fuzzing**
 - mostly program analysis
- **Grey-box Fuzzing**
 - no program analysis, but feedback

Why Grey-box Fuzzing ?

- Black-box Fuzzing

Definition: techniques that do not see the internals of the PUT, and can observe only the input/output behavior of the PUT, treating it as a black-box[1].

-No **program analysis**, no **feedback**

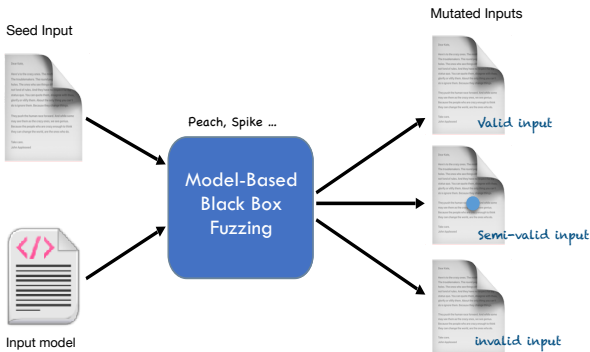


Why Grey-box Fuzzing ?

- **Black-box Fuzzing**

Definition: techniques that do not see the internals of the PUT, and can observe only the input/output behavior of the PUT, treating it as a black-box[1].

- No **program analysis**, no **feedback**



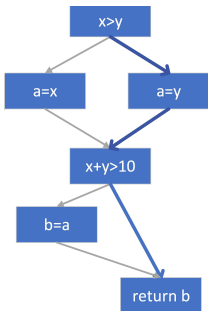
- You have no view of the PUT, but have some view of the input/output domain
- Fuzzing process is not changed according to some feedback
- Random mutated (not **effective**)

Why Grey-box Fuzzing ?

- White-box Fuzzing

Defination: techniques that generates test cases by analyzing the internals of the PUT and the information gathered when executing the PUT[1].

- Requires heavy-weight **program analysis** and constraint solving.



Cover more paths

$x \leq y \wedge x + y \leq 10$
 $x \leq y \wedge \neg x + y \leq 10$
 $\neg x \leq y$

Static Analysis

- Symbolic Execution
- Constrains Satisfaction iterative algorithm

Seed Input



(optional) crash locations

PUT States
Dynamic Symbolic Execution
(optional) taint analysis

Test cases



Why Grey-box Fuzzing ?

- White-box Fuzzing

Defination: techniques that generates test cases by analyzing the internals of the PUT and the information gathered when executing the PUT[1].

- Requires heavy-weight **program analysis** and constraint solving.

Seed Input



(optional) crash locations

PUT States
Dynamic
Symbolic
Execution
(optional) taint analysis

Test cases



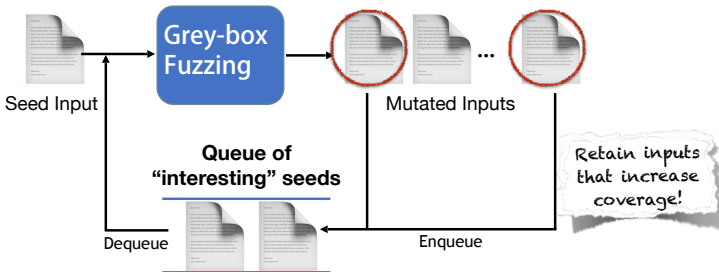
- You have the view of the PUT state(CFG,CG)
- Static analysis (effective but not **efficient**!)

Why Grey-box Fuzzing ?

- Grey-box Fuzzing

Definition: techniques that can obtain some information internal to the PUT and/or its executions[1] to generates test cases.

- Uses only lightweight instrumentation to glean some program structure
- And coverage **feedback**



1 Background

Pre-Knowledge
Motivation
Research Status

2 研究现状

3 研究内容

4 计划进度

5 参考文献

Why Directed Grey-Box Fuzz?

- 大家都会 \LaTeX ，好多学校都有自己的 Beamer 主题

Why Directed Grey-Box Fuzz?

- 大家都会 \LaTeX ，好多学校都有自己的 Beamer 主题
- 中文支持请选择 Xe \LaTeX 编译选项

1 Background

2 研究现状

Beamer 主题分类

3 研究内容

4 计划进度

5 参考文献

1 Background

2 研究现状

Beamer 主题分类

3 研究内容

4 计划进度

5 参考文献

- 有一些 \LaTeX 自带的
- 有一些 Tsinghua 的
- 本模板来源自 THU Beamer Theme
- 但是最初的 [link](#) [2] 已经失效了
- 这是原作者在 16-17 年做的一些 ppt: [戳我](#)

1 Background

2 研究现状

3 研究内容

美化主题

如何更好地做 Beamer

4 计划进度

5 参考文献

1 Background

2 研究现状

3 研究内容

- 美化主题
- 如何更好地做 Beamer

4 计划进度

5 参考文献

这一份主题与原始的 THU Beamer Theme 区别在于

- 顶栏的小点变成一行而不是多行
- 中文采用楷书
- 修改了主题色为南邮校徽颜色
- 参考文献格式按照毕设标准进行了修改
- 更多该模板的功能可以参考
<https://www.latexstudio.net/archives/4051.html>
- 下面列举出了一些 Beamer 的用法，部分节选自
<https://tuna.moe/event/2018/latex/>

1 Background

2 研究现状

3 研究内容

美化主题

如何更好地做 Beamer

4 计划进度

5 参考文献

Why Beamer

- \LaTeX 广泛用于学术界，期刊会议论文模板

Microsoft® Word	\LaTeX
文字处理工具	专业排版软件
容易上手，简单直观	容易上手
所见即所得	所见即所想，所想即所得
高级功能不易掌握	进阶难，但一般用不到
处理长文档需要丰富经验	和短文档处理基本无异
花费大量时间调格式	无需担心格式，专心作者内容
公式排版差强人意	尤其擅长公式排版
二进制格式，兼容性差	文本文件，易读、稳定
付费商业许可	自由免费使用

排版举例

无编号公式

$$J(\theta) = \mathbb{E}_{\pi_{\theta}}[G_t] = \sum_{s \in \mathcal{S}} d^{\pi}(s) V^{\pi}(s) = \sum_{s \in \mathcal{S}} d^{\pi}(s) \sum_{a \in \mathcal{A}} \pi_{\theta}(a|s) Q^{\pi}(s, a)$$

多行多列公式¹

$$\begin{aligned} Q_{\text{target}} &= \mathbf{r} + \gamma Q^{\pi}(s', \pi_{\theta}(s')) + \epsilon \\ \epsilon &\sim \text{clip}(\mathcal{N}(0, \sigma), -c, c) \end{aligned} \tag{1}$$

¹如果公式中有文字出现，请用 `\mathrm{}` 或者 `\text{}` 包含，不然就会变成 clip，在公式里看起来比 clip 丑非常多。

编号多行公式

$$\begin{aligned} A &= \lim_{n \rightarrow \infty} \Delta x \left(a^2 + \left(a^2 + 2a\Delta x + (\Delta x)^2 \right) \right. \\ &\quad + \left(a^2 + 2 \cdot 2a\Delta x + 2^2 (\Delta x)^2 \right) \\ &\quad + \left(a^2 + 2 \cdot 3a\Delta x + 3^2 (\Delta x)^2 \right) \\ &\quad + \dots \\ &\quad \left. + \left(a^2 + 2 \cdot (n-1)a\Delta x + (n-1)^2 (\Delta x)^2 \right) \right) \\ &= \frac{1}{3} (b^3 - a^3) \quad (2) \end{aligned}$$

LaTeX 常用命令

命令

<code>\chapter</code> 章	<code>\section</code> 节	<code>\subsection</code> 小节	<code>\paragraph</code> 带题头段落
<code>\centering</code> 居中对齐	<code>\emph</code> 强调	<code>\verb</code> 原样输出	<code>\url</code> 超链接
<code>\footnote</code> 脚注	<code>\item</code> 列表条目	<code>\caption</code> 标题	<code>\includegraphics</code> 插入图片
<code>\label</code> 标号	<code>\cite</code> 引用参考文献	<code>\ref</code> 引用图表公式等	

环境

<code>table</code> 表格	<code>figure</code> 图片	<code>equation</code> 公式
<code>itemize</code> 无编号列表	<code>enumerate</code> 编号列表	<code>description</code> 描述

LaTeX 环境命令举例

```
1 \begin{itemize}
2   \item A \item B
3   \item C
4   \begin{itemize}
5     \item C-1
6   \end{itemize}
7 \end{itemize}
```

- A
- B
- C
 - C-1

LaTeX 环境命令举例

```
1 \begin{itemize}
2   \item A \item B
3   \item C
4   \begin{itemize}
5     \item C-1
6   \end{itemize}
7 \end{itemize}
```

- A
- B
- C
 - C-1

```
1 \begin{enumerate}
2   \item 巨佬 \item 大佬
3   \item 萌新
4   \begin{itemize}
5     \item[n+e] 瑟瑟发抖
6   \end{itemize}
7 \end{enumerate}
```

- ① 巨佬
- ② 大佬
- ③ 萌新
 - n+e 瑟瑟发抖

L^AT_EX 数学公式

```
1 $V = \frac{4}{3}\pi r^3$  
2  
3 \[  
4   V = \frac{4}{3}\pi r^3  
5 \]  
6  
7 \begin{equation}  
8   \label{eq:vsphere}  
9   V = \frac{4}{3}\pi r^3  
10 \end{equation}
```

$$V = \frac{4}{3}\pi r^3$$

$$V = \frac{4}{3}\pi r^3$$

$$V = \frac{4}{3}\pi r^3 \quad (3)$$

- 更多内容请看 [这里](#)

```
1 \begin{table}[htbp]
2   \caption{编号与含义}
3   \label{tab:number}
4   \centering
5   \begin{tabular}{cl}
6     \toprule
7     编号 & 含义 \\
8     \midrule
9     1 & 4.0 \\
10    2 & 3.7 \\
11    \bottomrule
12   \end{tabular}
13 \end{table}
14 公式~(\ref{eq:vsphere}) 的
15 编号与含义请参见
16 表~\ref{tab:number}。
```

Table 1: 编号与含义

编号	含义
1	4.0
2	3.7

公式 (3) 的编号与含义请参见表 1。

作图

- 矢量图 eps, ps, pdf
 - METAPOST, pstricks, pgf ...
 - Xfig, Dia, Visio, Inkscape ...
 - Matlab / Excel 等保存为 pdf
- 标量图 png, jpg, tiff ...
 - 提高清晰度，避免发虚
 - 应尽量避免使用



Figure 1: 这个校徽就是矢量图，虽然看起来不像，但确实是矢量图格式

1 Background

2 研究现状

3 研究内容

4 计划进度

5 参考文献

- 一月：完成文献调研
- 二月：研究 THU Beamer Theme 的实现
- 三、四月：修改 NJUPT Beamer 主题
- 五月：论文撰写

1 Background

2 研究现状

3 研究内容

4 计划进度

5 参考文献

- [1] MANÈS V J, HAN H, HAN C, et al. The art, science, and engineering of fuzzing: A survey[J]. IEEE Transactions on Software Engineering, 2019, 47(11): 2312–2331.
- [2] UNKNOWN. THU Beamer Theme[C/OL] //None. 2015: 1–10.
<http://far.tooold.cn/post/latex/beamertsinghua>.

Thanks!