

软件分析与测试

第三次课

严俊



中国科学院大学
University of Chinese Academy of Sciences

ISCAS

中国科学院软件研究所
Institute of Software, Chinese Academy of Sciences



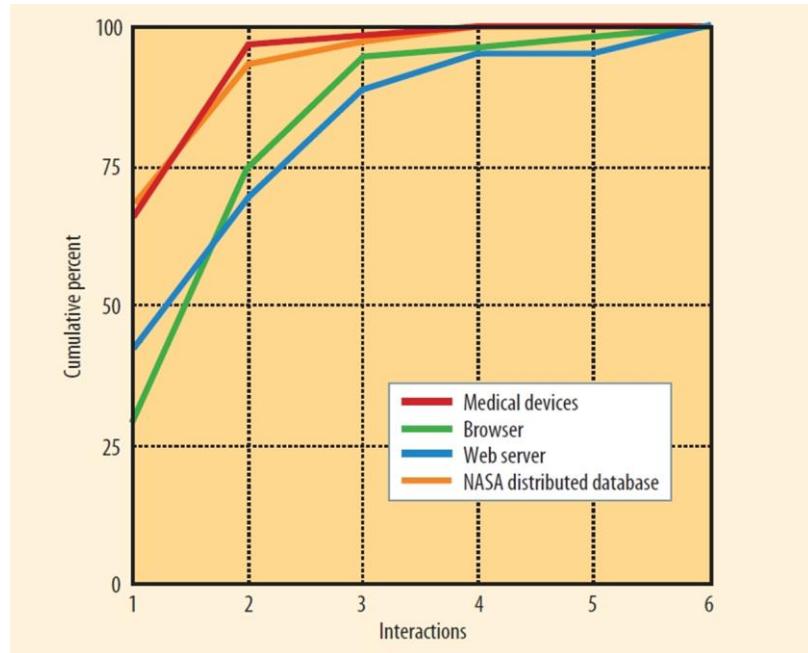


组合测试

参数的关联错误 (Interaction Fault)



- 大部分的错误出现于少数参数的取值组合
 - ☒ 70% 的软件故障是由两个参数的相互作用引发的
 - ☒ 90% 以上的软件故障是由三个参数的相互作用引发
 - ☒ ...
- 可以利用这些性质来减少测试用例的数目



IEEE Computer, Aug. 2009



- 按照如下目标设计测试用例
- 给定的一个小正整数 t (称为组合强度, 一般为2或者3)
- 测试用例应该**覆盖任意 t 个参数的所有取值组合**



- 用户可能会要求某些取值组合必须被测试，这种取值组合被称为种子(seed)。

例：AETG

这一段测试规范

```
seed {  
    a b c d  
    2 4 8 3  
}
```

表示(a, b, c, d) = (2, 4, 8, 3) 这种组合形式在测试用例集中至少出现一次。

- 可用于回归测试



- 某些参数组合事实上是不允许出现或者是没有意义的。

例：简单计算器

“十六进制” + “浮点数”的组合是无效组合

这种情况称为参数之间的冲突

- 可以用约束描述参数的限制

例： $(x > 5) \rightarrow (z < 3)$



- 允许不同参数之间的覆盖强度不同

例：M.B. Cohen 定义的变强度覆盖数组

VCA(27;2,3⁹2²,{CA(27;3,3³)³})

表示一个强度为2 的混合覆盖数组，其中包含三个子数组，并且在每个子数组内部需要所有的参数组合满足3 覆盖

- 适合于重点测试系统中的某些强关联的或者关键参数



- * **PICT (pairwise testing)**

<http://download.microsoft.com/download/f/5/5/f55484df-8494-48fa-8dbd-8c6f76cc014b/pict33.msi>

- * **ACTS**

<https://csrc.nist.gov/projects/automated-combinatorial-testing-for-software/downloadable-tools#acts>

- * **Cascade**

- * **IPOG**

- * **CASA**



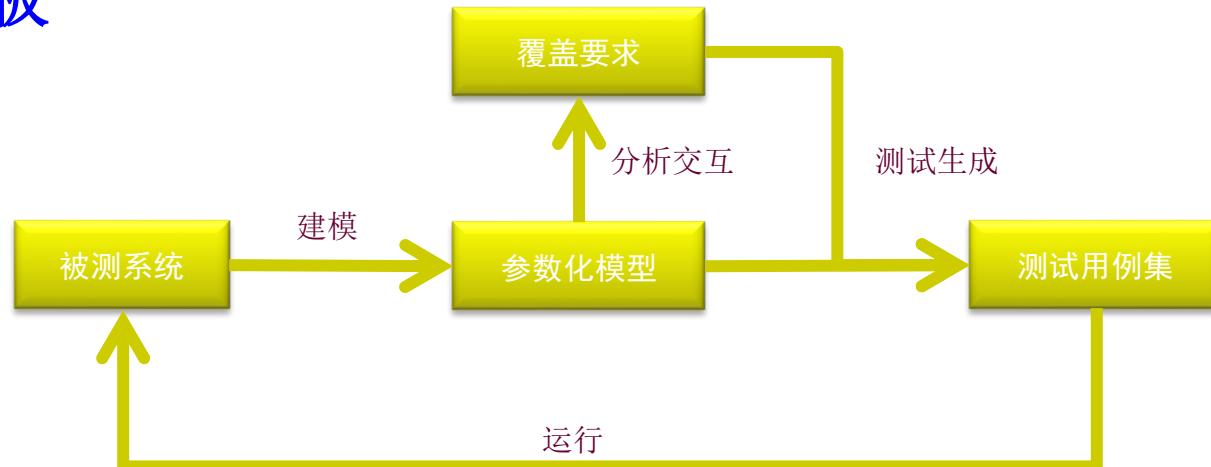
➤ 组合测试案例

组合测试的常用模式——具体建模



➤ 被测系统的接口即为参数化模型

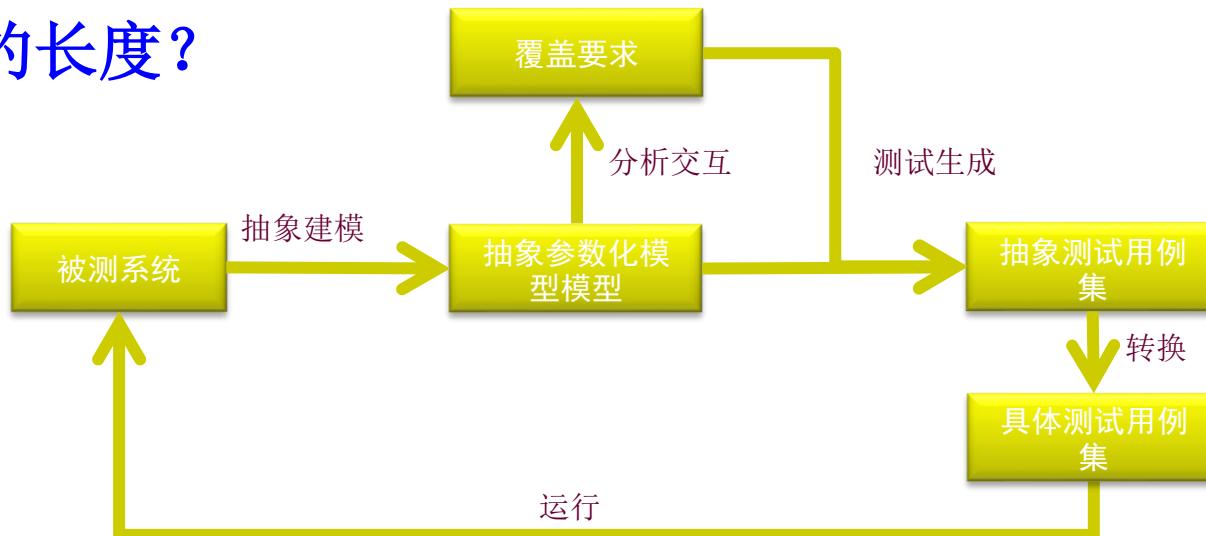
- ☒ 函数参数
- ☒ 系统配置
- ☒ 操作面板



组合测试的常用模式——抽象建模



- 无法具体建模，多以输入的某个属性作为参数
 - ☒ 三角形的类别？
 - ☒ 是否有某项属性？
 - ☒ 输入的长度？



组合测试建模的一些建议



➤ 建模过程主要分为如下阶段

- ☒ 给定测试的目标
- ☒ 参数识别——有哪些参数？有哪些组合？
- ☒ 约束识别
- ☒ 交互识别

➤ 一般情况下

- ☒ 参数不要过多（常见规模约为**4-30**）
- ☒ 参数取值不要过多（测试用例数量多项式增长）
- ☒ 约束不要太长（难以理解，不易计算）
- ☒ 覆盖强度不要太高（常用**2-3**）

➤ 建议对参数按功能进行分割

- ☒ 尽量避免无关参数之间的组合



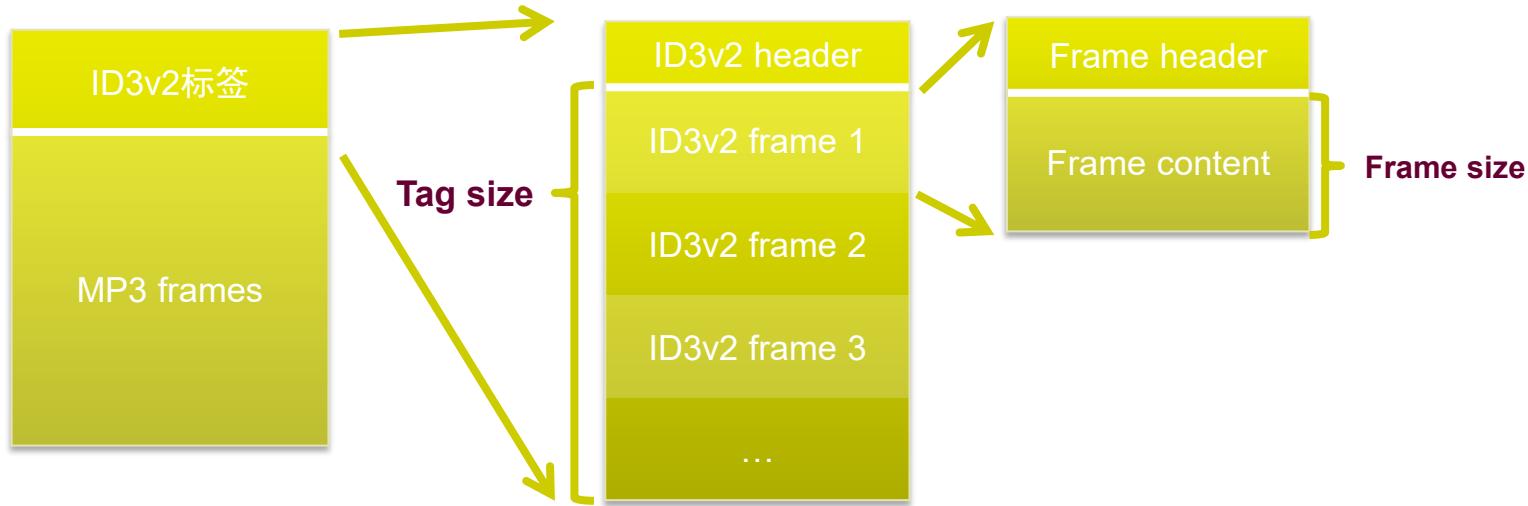
案例：MP3播放器测试

Zhiqiang Zhang, et al., Combinatorial Testing on ID3v2 Tags of MP3 Files. IWCT 2012

MP3文件的ID3标签



➤ MP3文件的ID3标签



ID3v2 Header:

ID3v2/file identifier	"ID3"
ID3v2 version	\$03 00
ID3v2 flags	%abc00000
ID3v2 size	4 * %0xxxxxxxx

ID3v2 Frame Header:

Frame ID	\$xx xx xx xx (four characters)
Size	\$xx xx xx xx
Flags	\$xx xx



➤ 文本类ID3v2帧的内容

Text encoding \$xx

Information <text string>

➤ 中国常用的编码类型：

☒ \$00: ASCII/GBK

☒ \$01: Unicode (UTF-16)

- 小尾端 (LE)

- 大尾端 (BE)

☒ \$02: Unicode大尾端

☒ \$03: UTF-8



➤ 测试目标

☒ 对于各种编码的**ID3v2**帧，播放器是否能够识别和显示？

➤ 常用的6种标签

☒ **TIT2** (标题)

☒ **TPE1** (艺术家)

☒ **TALB** (专辑)

☒ **TRCK** (音轨号)

☒ **TCON** (风格)

☒ **TYER** (年份)

TIT2	TPE1	TALB	TRCK	TCON	TYER
NONE	NONE	NONE	NONE	NONE	NONE
ASCII	ASCII	ASCII	ASCII	ASCII	ASCII
GBK	GBK	GBK			
ULE	ULE	ULE			
UBE	UBE	UBE			
UBE2	UBE2	UBE2			
UTF-8	UTF-8	UTF-8			



测试结果（某车载播放器）

条件	结果
所有情况	ASCII标签能够正常显示
所有情况	UBE2标签不能显示
TIT2（标题）为空	标题显示为文件名
TPE1（艺术家）为空	艺术家显示为目录名
$TPE1 \in \{\text{ASCII}, \text{UBE2}\}$ 且 $TIT2 \in \{\text{GBK}, \text{ULE}, \text{UBE}, \text{UTF-8}\}$	标题不能显示
$TPE1 \notin \{\text{ASCII}, \text{UBE2}\}$ 且 $TIT2 \notin \{\text{GBK}, \text{ULE}, \text{UBE}, \text{UTF-8}\}$	标题显示为文件名 艺术家显示为目录名

测试结果（Sony播放器）



条件	结果
<code>TIT2 ∈ {UBE2, UTF-8}</code>	标题显示为文件名
<code>TIT2 !=NONE &&</code> <code>TPE1 ∈ {UBE2, UTF-8}</code>	标题显示为文件名
其他情况	所有信息可以正常显示

测试结果 (Windows Media Player 12)



条件	结果
所有情况	UBE2标签不能正常显示
所有情况	UTF-8标签不能正常显示
TIT2（标题）为空	标题显示为文件名
$TPE1 \in \{\text{UBE2}, \text{UTF-8}\}$	所有非空信息显示不正常



案例：支持HTML5的浏览器测试

Xi Deng, et al., Combinatorial Testing of Browsers' Support for Multimedia. IEEE Trans on Reliability, 2020.

HTML5



**HTML5: standard for web page
programming language.**

**New requirement for browsers:
support the multimedia natively
(instead of by other third-party
plugins formerly).**



Model of CT



Resource loading:

Element: {audio, video}
ElementSrc: {http, https, ftp, file, relative, not_set}
SourceSrc: {http, https, ftp, file, relative}
CROS: {anonymous, use-credentials, not_set}
Type:{audio/mp3, ..., video/mp4,...}
ContainerFormat:{MP3, AAC, Oga, MP4, WebM, Ogv}
Constraints:
 $(ES \neq \text{not_set}) \rightarrow ((SS = \#) \wedge (TY = \#))$
 $((ES=\text{relative}) \vee (SS=\text{relative})) \rightarrow CO = \text{not_set}$

Can load?
Can decode?

Visual contents' layout:

VideoCF: {MP4, WebM, Ogv}
PosterFormat : {PNG, JPEG, GIF, BMP, None}
WR, HR, PWR, PHR: {0.1, 1, 10, None}
Constraints:
for poster image

Properly displayed?

Playback of audio:

AudioCF: {MP3, AAC, Oga}
SampleRate : {8,24,48,96}
ChannelNumber: {mono, stereo, 5.1}
FrameSR, FCN: {<,=, >}
BitrateMode: {VBR, CBR}
Constraints:
according to container formats

Normally played?

Playback of video:

VCF: {MP4, WebM, Ogv}
FrameRate : {1, 25, 50}
ChromaSubSampling: {4:4: 4, 4:2:2, 4:2:0}
SSR, SCN, SFR, SCSS: {<,=, >}
Constraints:
according to container formats

Normally played?

部分测试结果

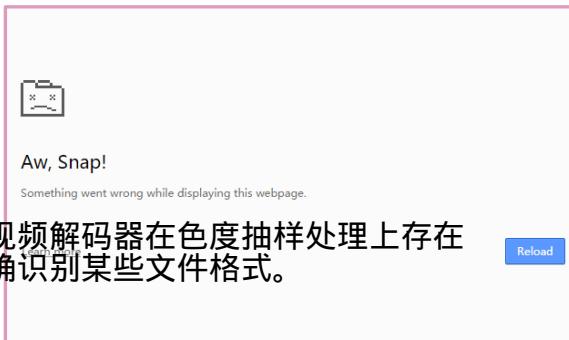


Playback of video objects:

sub-sampling rate fi高色度采样率（如 4:4:4）或低采样率（如 4:2:0）支持不足

Inadequate support for high chroma

- eg. VCF = MP4, CSS ≠ 4:2:0
- Result: video is played with no video image in Firefox, Opera, IE and Edge.
- eg. VCF = WebM, CSS = 4:4:4, SCSS ≠ eq
- Result: web page crashes in Opera and Chrome



不同浏览器的视频解码器在色度抽样处理上存在差异，无法正确识别某些文件格式。

视频格式参数之间的兼容性（VCF、FR、SCSS）不一致会直接影响解码质量。

Abnormal playback for certain frame rates and mismatches of video features

- freezing
- macro-blocking
 - eg. VCF = Ogv, FR = 25, SCSS ≠ eq
 - Result: video image is messed up in Opera, Chrome and UC
- accelerating or decelerating



部分测试结果



旧Bug跨版本延续

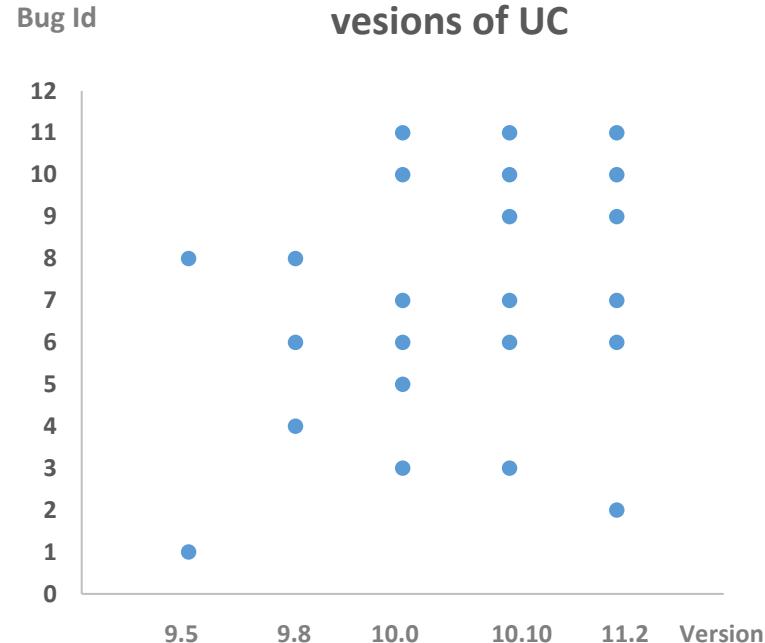
→ Bugs may last in multiple versions.

- eg. Bug with id 6
- ACF = Oga, FSR ≠ eq
- Result: The playback of sound is accelerated or decelerated.

↗ New bugs get introduced in newer releases.

- eg. Bug with id 3
- EL = audio, URL = https, CF in {Ogv, AAC}
- Being brought in with the extending of supported formats

Occurrences of bugs in historical versions of UC



新Bug在新版本中不断引入



案例：编译时可配置文件的静态分析

Dong Yan, et al., Comprehensive Static Analysis for Configurable Software via Combinatorial Instantiation. COMPSAC 2017

编译时可配置软件



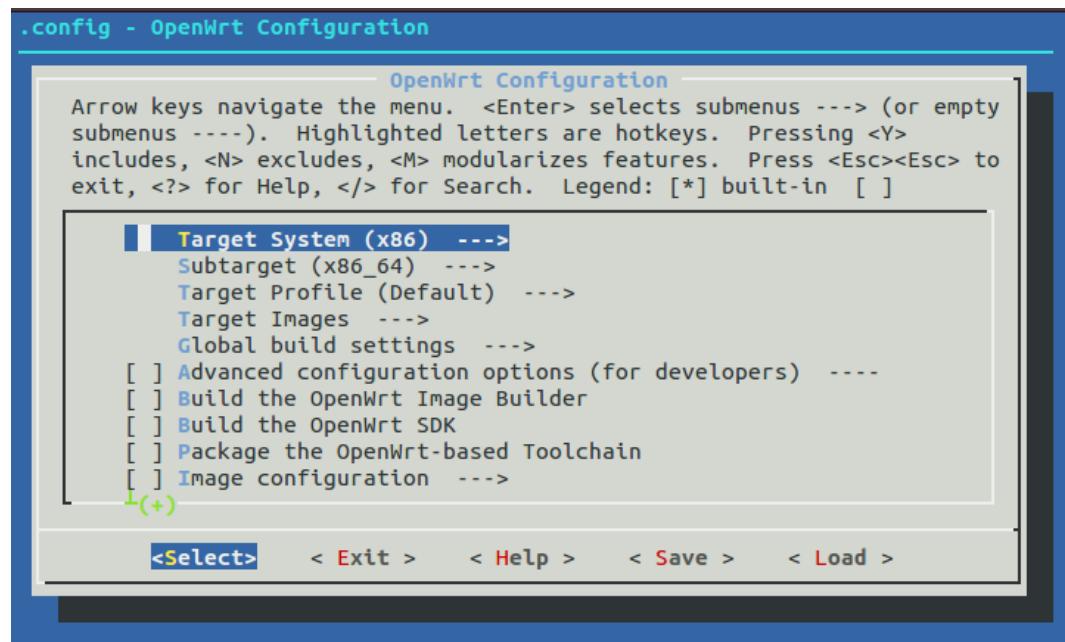
➤ 在编译前可以通过配置定制的软件，比如Linux内核，openWRT，coreboot等。

➤ 通过一些描述语言如KConfig来管理配置项

- ☒ 易于用户使用
- ☒ 方便维护
- ☒ 有配置界面

➤ KConfig功能

- ☒ 描述变量
- ☒ 组织变量
- ☒ 描述变量间约束



KConfig-描述变量



```
menu "MENU"
    config CONFIG_1
        bool 'CONFIG_1'
        default y
    choice
        prompt "CHOICE"
        depends on CONFIG_1
        default CHOICE_1
        config CHOICE_1
            bool "CHOICE_1"
        config CHOICE_2
            bool "CHOICE_2"
        config CHOICE_3
            bool "CHOICE_3"
    endchoice
    config ANON_CONFIG
        def_bool n
config BOOL_CONFIG
    bool "BOOL_CONFIG"
    default y
    select ANON_CONFIG if CHOICE_1
config INT_CONFIG
    int "INT_CONFIG"
    default 6 if !CONFIG_1
    default 10 if CONFIG_1
    range 5 32
    depends on BOOL_CONFIG
config STR_CONFIG
    string
    prompt "STR_CONFIG"
    default "def_str"
    depends on BOOL_CONFIG
if BOOL_CONFIG
    config CONFIG_2
        bool "CONFIG_2"
endif
endmenu
```

- ① 声明变量： config关键字
- ② 变量类型：
bool,tristate(y,n,m),hex,int,string
- ③ 文字描述： 可见变量和不可见变量
- ④ 默认值： 多个默认值互斥

```
[*] CONFIG_1
    CHOICE (CHOICE_1) --->
    [*] BOOL_CONFIG
    (6) INT_CONFIG
    (def_str) STR_CONFIG
    [ ] CONFIG_2
```

KConfig-组织变量

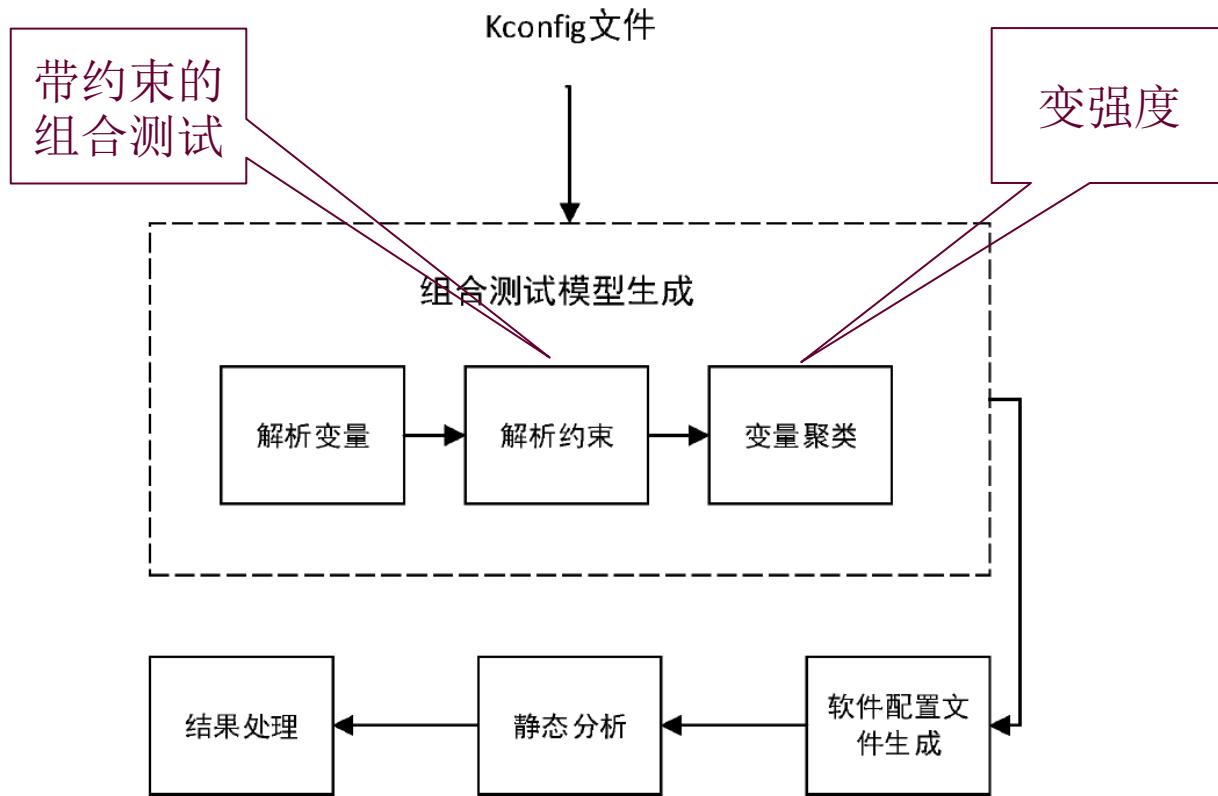


```
        config BOOL_CONFIG
            bool "BOOL_CONFIG"
            default y
            select ANON_CONFIG if CHOICE_1
config INT_CONFIG
    int "INT_CONFIG"
    default 6 if !CONFIG_1
    default 10 if CONFIG_1
    range 5 32
    depends on BOOL_CONFIG
config STR_CONFIG
    string
    prompt "STR_CONFIG"
    default "def_str"
    depends on BOOL_CONFIG
if BOOL_CONFIG
    config CONFIG_2
        bool "CONFIG_2"
endif
endmenu
menu "MENU"
config CONFIG_1
    bool "CONFIG_1"
    default y
choice
    prompt "CHOICE"
    depends on CONFIG_1
    default CHOICE_1
config CHOICE_1
    bool "CHOICE_1"
config CHOICE_2
    bool "CHOICE_2"
config CHOICE_3
    bool "CHOICE_3"
endchoice
config ANON_CONFIG
    def_bool n
```

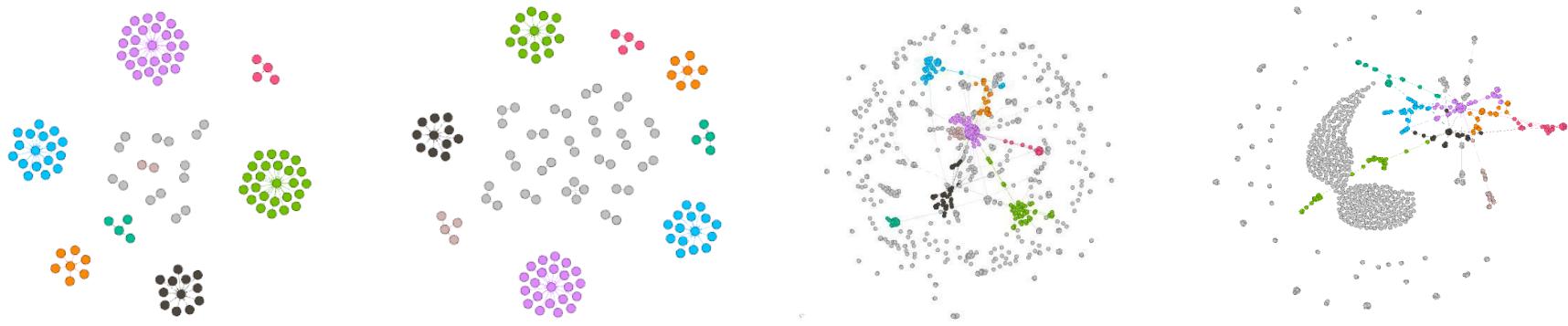
- ① menu结构
- ② choice结构
- ③ if结构
- ④ config结构

```
[*] CONFIG_1
    CHOICE (CHOICE_1) --->
    [*] BOOL_CONFIG
    (6)   INT_CONFIG
    (def_str) STR_CONFIG
    [ ]   CONFIG_2
```

分析流程



编译配置项的聚类特性



axTLS

libpayload

busybox

coreboot

实验结果



软件	结果	组合测试策略				
		global	cluster	random	allyes	default
axTLS	配置文件	25	19	25	1	1
	静态分析时间	1033.7	924.1	684.1	213.0	46.9
	缺陷数	16	16	12	12	1
busybox	配置文件	44	20	44	1	1
	静态分析时间	7028.6	2373.6	1593.9	1403.2	1720.9
	缺陷数	49	45	5	24	22
coreboot	配置文件	95	10	95	1	1
	静态分析时间	3719.6	862.7	3171.8	224.7	227.9
	缺陷数	15	15	15	15	15
libpayload	配置文件	19	11	19	1	1
	静态分析时间	484.2	346.9	451.7	43.1	271.7
	缺陷数	16	16	16	2	7



缺陷的触发条件

软件	缺陷编号	描述	触发条件
axTLS	1	空指针解引用	SSL_PRIVATE_KEY_LOCATION=EMP SSL_GENERATE_X509_CERT=y
	2		SSL_SKELETON_MODE=y
	3	空指针传递给非空参数	HTTP_HAS_CGI=y
busybox	1	内存泄漏	ENABLE_FEATURE_DD_IBS_OBS=y ENABLE_FEATURE_DD_SIGNAL_HANDLING=y ENABLE_FEATURE_DD_THIRD_STATUS_LINE=y
	2		ENABLE_FEATURE_PREFER_APPLETS=n
	3		MODPROBE=y
	4, 5, 6	空指针解引用	FEATURE_2_4_MODULES=y

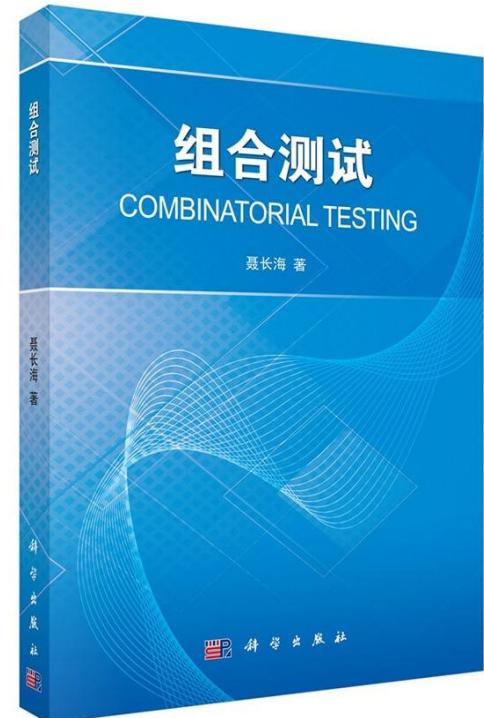


- 组合测试概念
- 经典组合测试与非经典组合测试
- 组合测试用例生成方法
- 组合测试的应用

参考资料



- 严俊,张健. 组合测试: 原理与方法. 软件学报. 2009;20(6).
- C. Nie, H. Leung: A survey of combinatorial testing. ACM Comput. Surv. 43(2): 11:1-11:29 (2011)
- J. Zhang, Z. Zhang, F. Ma (2014) Automatic Generation of Combinatorial Test Data. Springer Briefs in Computer Science. Springer, Berlin, Heidelberg. ISBN: 978-3-662-43428-4
- NIST, Combinatorial Testing Tutorial, <http://csrc.nist.gov/groups/SNS/acts/documents/kuhn-idga-mte.ppt>





- International Workshop on Combinatorial Testing
(IWCT @ ICST)

- International Workshop on Combinatorial Testing
and its Applications (CTA @ QRS)