Analyzing the Internal Consistency of the Linux KConfig Model

Stefan Hengelein stefan.hengelein@fau.de

Daniel Lohmann dl@cs.fau.de

System Software Group
Friedrich-Alexander University Erlangen-Nürnberg (FAU)

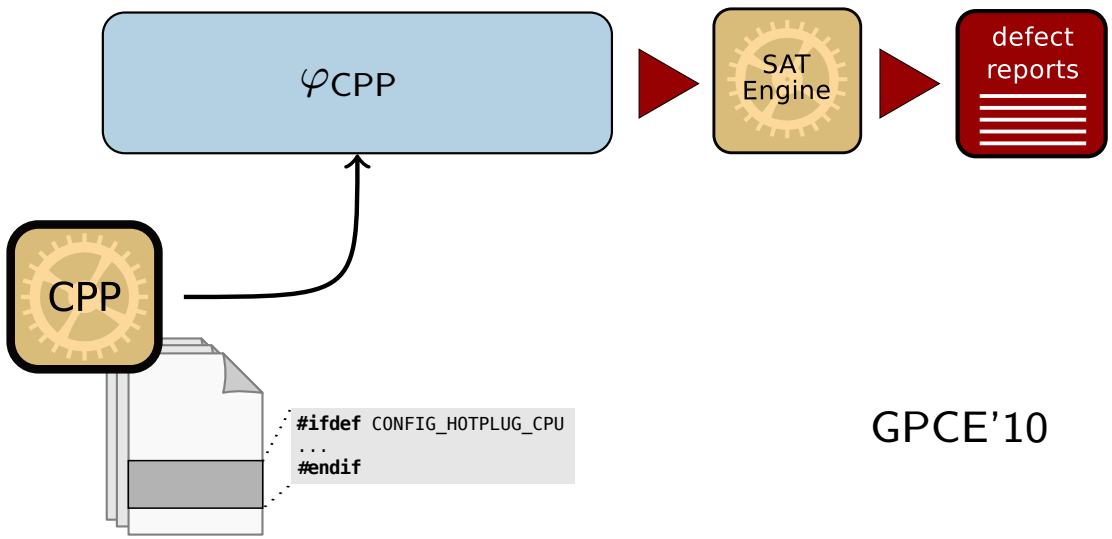
https://cados.cs.fau.de

FOSD Meeting '15

supported by DFG

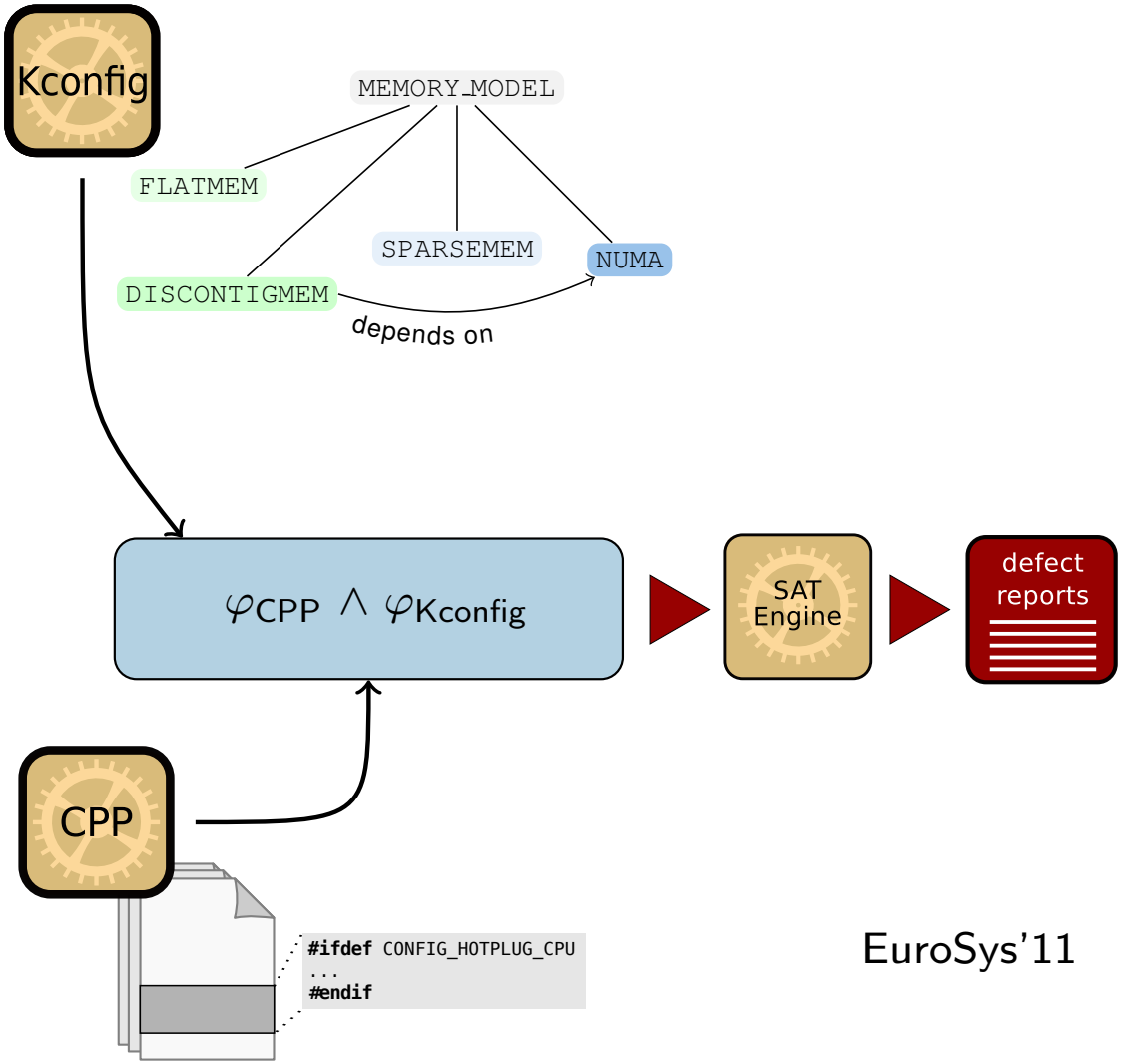
The VAMOS/CADOS approach so far

Check for Satisfiability



The VAMOS/CADOS approach so far

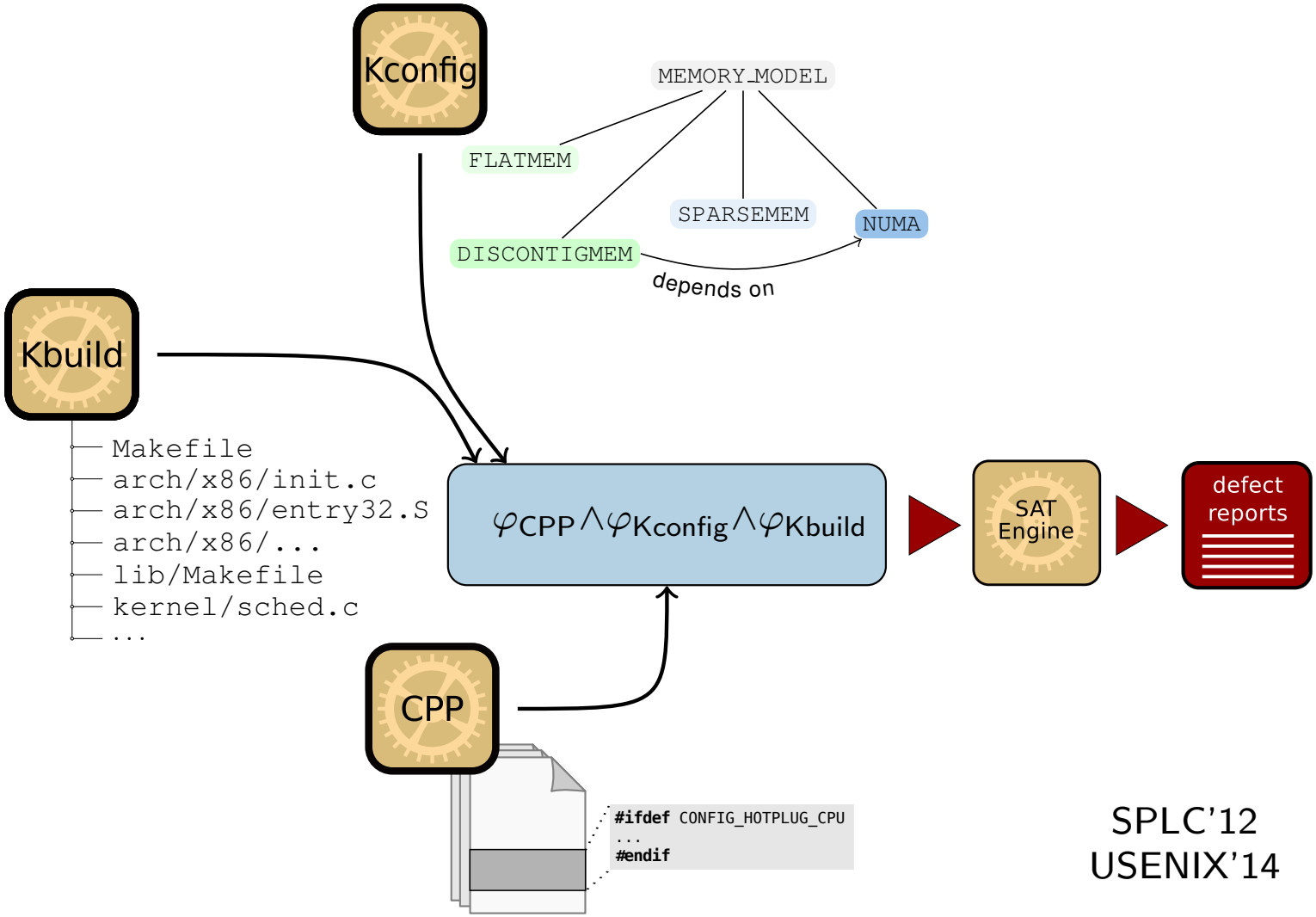
Combine and Check for Satisfiability



EuroSys'11

The VAMOS/CADOS approach so far

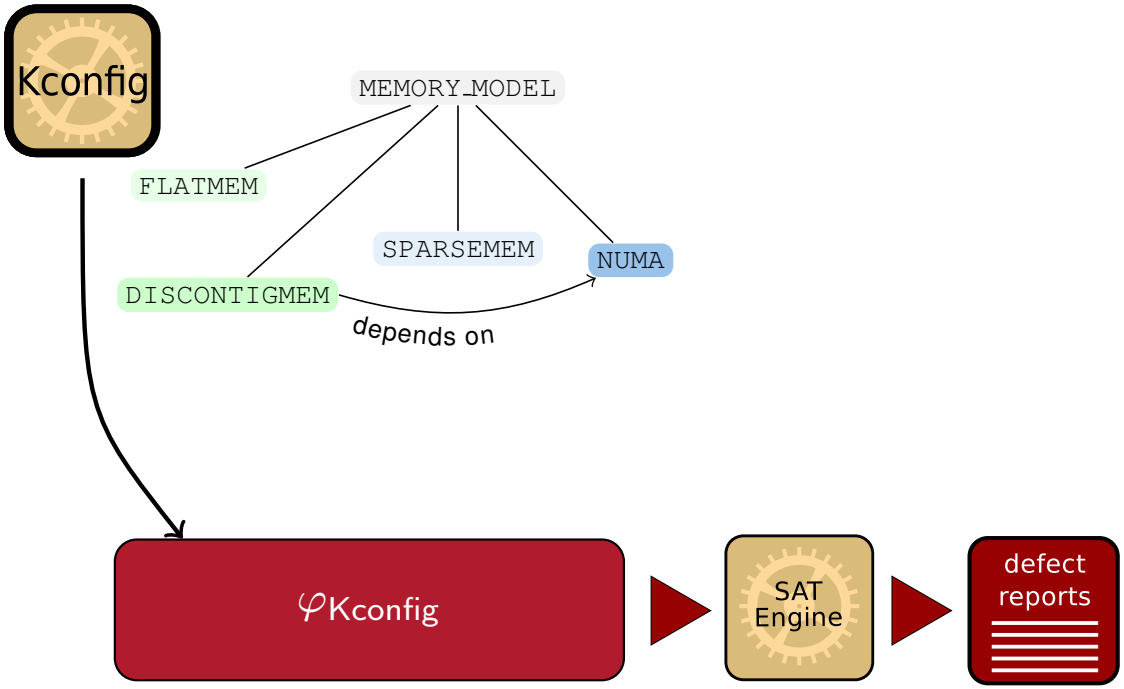
Combine and Check for Satisfiability



SPLC'12
USENIX'14

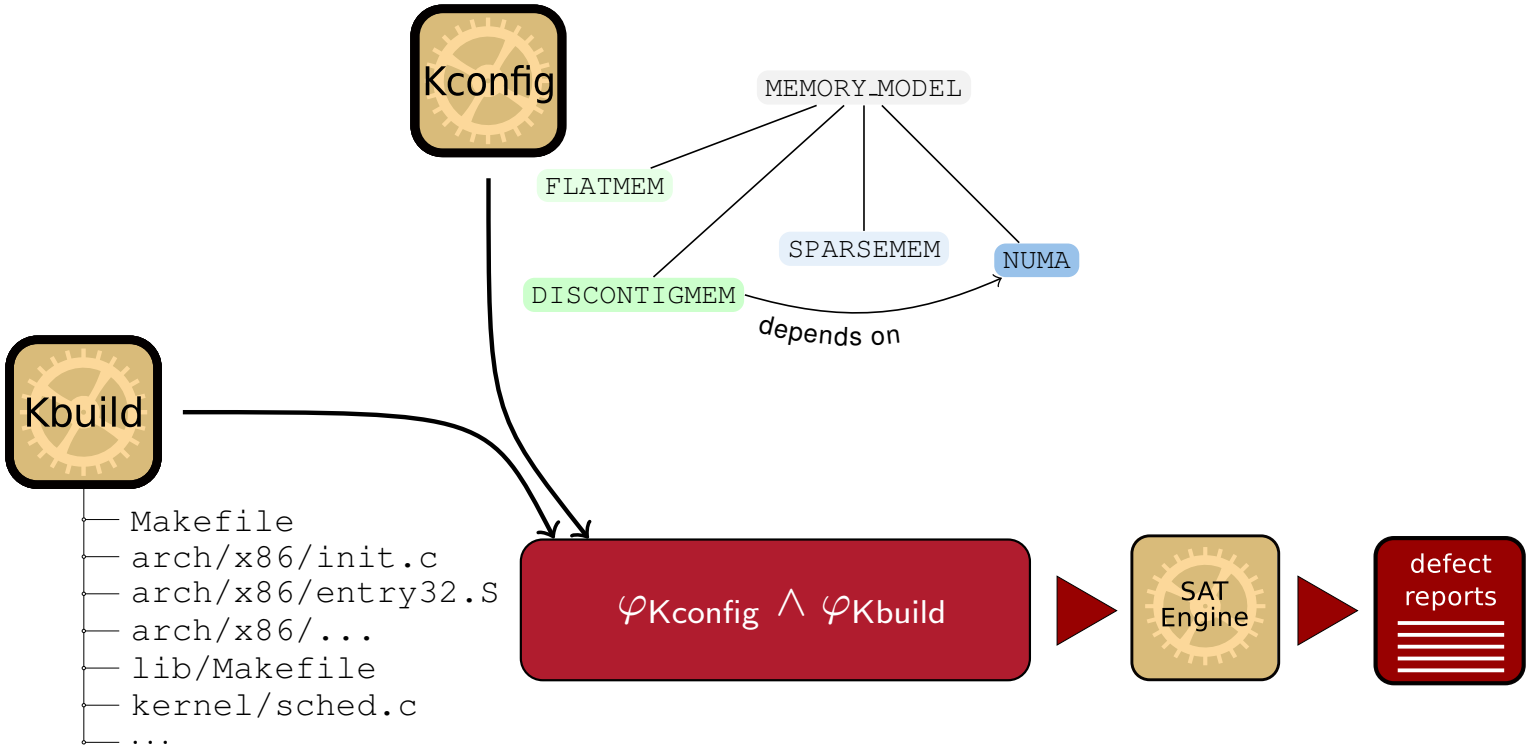
But, what about...

Check for Satisfiability



But, what about...

Combine and Check for Satisfiability



Defect Types

- Logical:
  - Contradiction in direct dependencies
  - Contradiction in direct dependencies + transitive dependencies
- Missing:
  - Force disable undefined options on the current architecture
- Undead:
  - ~~always on symbols~~ intended undead
  - Transitive always on, selected by option that is always on itself
- Unreachable Symbols:
  - No prompt, no default, no selects

Defect Types

- Logical:
  - Contradiction in direct dependencies 0
  - Contradiction in direct dependencies + transitive dependencies 1
- Missing:
  - Force disable undefined options on the current architecture 39
- Undead:
  - ~~always on symbols~~ intended undead
  - Transitive always on, selected by option that is always on itself 28
- Unreachable Symbols: unique: 430
  - No prompt, no default, no selects accumulated: 1434

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
    depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
    bool "OMAP4 errata: Async Bridge Corruption"
    depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
   depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
    bool "ARMv6 based platforms (ARM11)"

endmenu
```

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
    depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
    bool "OMAP4 errata: Async Bridge Corruption"
    depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
   depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
    bool "ARMv6 based platforms (ARM11)"

endmenu
```

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
    depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
    bool "OMAP4 errata: Async Bridge Corruption"
    depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
   depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
    bool "ARMv6 based platforms (ARM11)"

endmenu
```

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
    depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
    bool "OMAP4 errata: Async Bridge Corruption"
    depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
   depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
    bool "ARMv6 based platforms (ARM11)"

endmenu
```

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
    depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
    bool "OMAP4 errata: Async Bridge Corruption"
    depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
   depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
    bool "ARMv6 based platforms (ARM11)"

endmenu
```

Example - Logical Defect in KConfig

```
menu "TI OMAP/AM/DM/DRA Family"
     depends on ARCH_MULTI_V6

config OMAP4_ERRATA_I688
     bool "OMAP4 errata: Async Bridge Corruption"
     depends on !ARCH_MULTIPLATFORM

endmenu


menu "Multiple platform selection"
    depends on ARCH_MULTIPLATFORM

config ARCH_MULTI_V6
     bool "ARMv6 based platforms (ARM11)"

endmenu
```

Conclusion

- Patch submitted (`https://lkml.org/lkml/2015/2/25/503`)
- Applied to "omap-for-v4.1/fixes-not-urgent", not yet merged
- Deleted 105 lines of code

Example - Missing Defect in KConfig

```
config GPIO_MB86S7X
    bool "GPIO support for Fujitsu MB86S7x Platforms"
    depends on ARCH_MB86S7X
    help
        Say yes here to support the GPIO controller in
            Fujitsu MB86S70 SoCs.
```

Example - Missing Defect in KConfig

```
config GPIO_MB86S7X
     bool "GPIO support for Fujitsu MB86S7x Platforms"
     depends on ARCH_MB86S7X
     help
          Say yes here to support the GPIO controller in
              Fujitsu MB86S70 SoCs.
```

never defined

Example - Missing Defect in KConfig

```
config GPIO_MB86S7X
     bool "GPIO support for Fujitsu MB86S7x Platforms"
     depends on ARCH_MB86S7X
     help
          Say yes here to support the GPIO controller in
              Fujitsu MB86S70 SoCs.
```

never defined

Conclusion

- GPIO_MB86S7X is only used to compile **one** file
- Currently 232 lines of dead code
- Introduced with commit 0da094d8 (2015-01-19)

Example - Undead in KConfig

- Undead:

```
config X86
    def_bool y
    ...
    select ANON_INODES
    ...

config ANON_INODES
    bool
```

Example - Undead in KConfig

- Undead:

```
config X86
    def_bool y
    ...
    select ANON_INODES
    ...

config ANON_INODES
    bool
```

remove

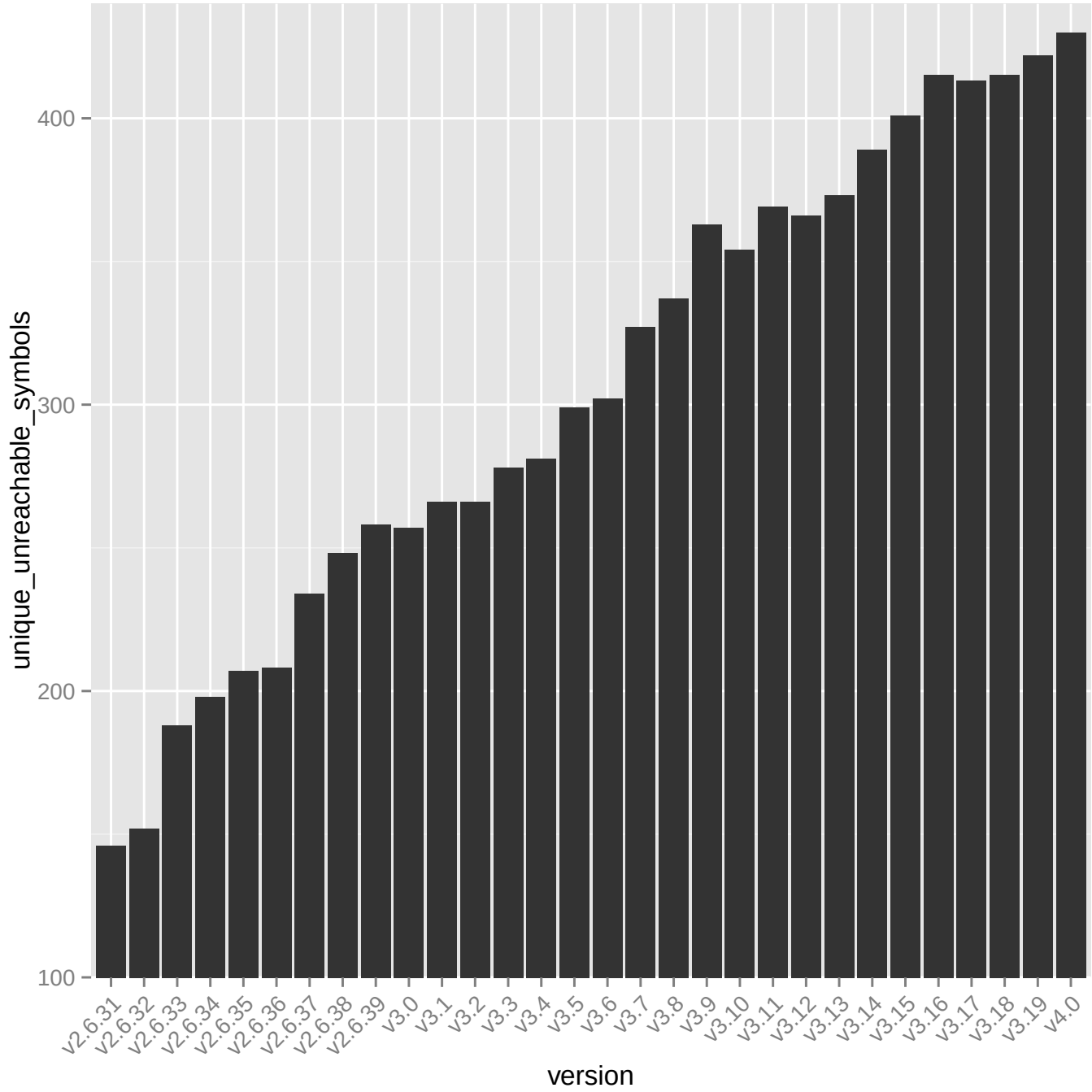
replace with “def_bool y”

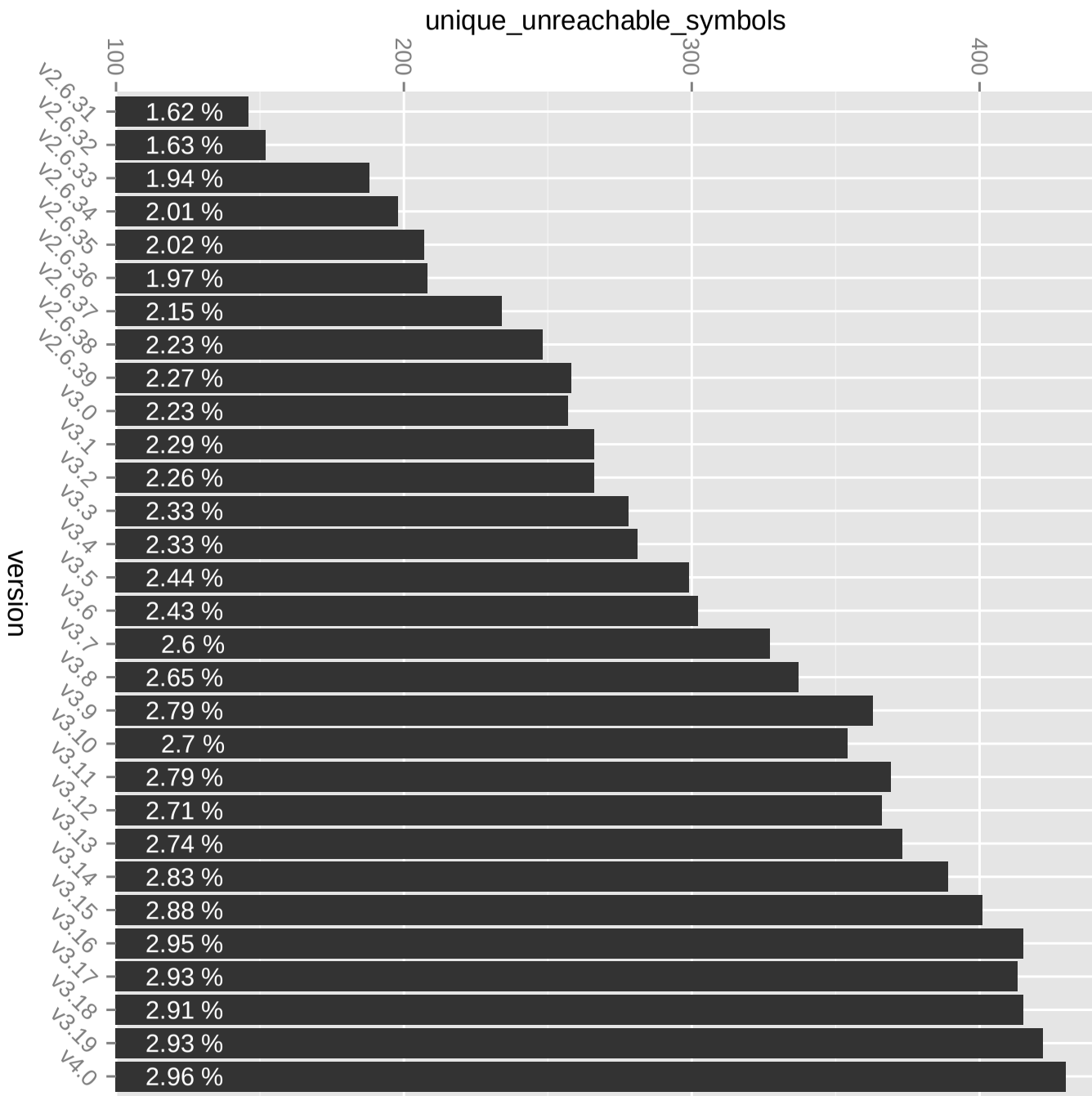
Unreachable Symbols

- Definition:
  - No prompt
  - No default or default n
  - No selects on symbol

```
config SND_SOC_AU1XI2SC
    tristate
```

- However:
  - There is almost always code behind these symbols
- Often:
  - The same unreachable symbol is defined on multiple architectures


unique_unreachable_symbols
100
200
300
400
version
v2.6.31
v2.6.32
v2.6.33
v2.6.34
v2.6.35
v2.6.36
v2.6.37
v2.6.38
v2.6.39
v3.0
v3.1
v3.2
v3.3
v3.4
v3.5
v3.6
v3.7
v3.8
v3.9
v3.10
v3.11
v3.12
v3.13
v3.14
v3.15
v3.16
v3.17
v3.18
v3.19
v4.0


unique_unreachable_symbols
100
200
300
400
version
v2.6.31
v2.6.32
v2.6.33
v2.6.34
v2.6.35
v2.6.36
v2.6.37
v2.6.38
v2.6.39
v3.0
v3.1
v3.2
v3.3
v3.4
v3.5
v3.6
v3.7
v3.8
v3.9
v3.10
v3.11
v3.12
v3.13
v3.14
v3.15
v3.16
v3.17
v3.18
v3.19
v4.0
1.62 %
1.63 %
1.94 %
2.01 %
2.02 %
1.97 %
2.15 %
2.23 %
2.27 %
2.23 %
2.29 %
2.26 %
2.33 %
2.33 %
2.44 %
2.43 %
2.6 %
2.65 %
2.79 %
2.7 %
2.79 %
2.71 %
2.74 %
2.83 %
2.88 %
2.95 %
2.93 %
2.91 %
2.93 %
2.96 %

Unreachable Symbols

- Do `unreachable symbols` cause problems?
  - Functional Bugs? Didn't find one, yet
  - Dead code / dead files
  - At least one is intentional: `BROKEN`
- Why are they in the kernel?
  - Removed selects and forgot to remove symbol/code?
  - Not yet fully implemented features?
  - Used downstream kernels?

Studying the Impact of Unreachable Symbols

- Symbols can never be enabled

  $\Rightarrow$ Force disable them for dead analysis

- Dead analysis on Linux/`v4.0` with file preconditions:
	current	+ unreachable information	
defects	1461	1808	+347 (+23%)
  - 49 files are dead

```
#B00:sound/soc/au1x/i2sc.c:0:0:sound/soc/au1x/i2sc.c:0:0:
B00
&&
( B0 <-> CONFIG_PM )
&& ( B1 <-> ( ! (B0) )
...
&& (CONFIG_FUTEX -> (CONFIG_RT_MUTEXES))
&& (CONFIG_GENERIC_IO -> ((CONFIG_HAS_IOMEM)))
&& (CONFIG_HAS_IOMEM -> (!CONFIG_NO_IOMEM && ((!CONFIG_NO_IOMEM) ->
     CONFIG_GENERIC_IO) && (!CONFIG_NO_IOMEM)))
&& (CONFIG_I2C -> (!CONFIG_I2C_MODULE && CONFIG_RT_MUTEXES))
&& (CONFIG_I2C_MODULE -> (!CONFIG_I2C && CONFIG_MODULES && CONFIG_RT_MUTEXES))
&& (CONFIG_INPUT -> (!CONFIG_UML && !CONFIG_INPUT_MODULE))
&& (CONFIG_INPUT_MODULE -> (!CONFIG_UML && !CONFIG_INPUT && CONFIG_MODULES))
&& (CONFIG_RT_MUTEXES -> ((CONFIG_FUTEX || CONFIG_I2C || CONFIG_I2C_MODULE)))
&& (CONFIG_SND -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML) && !
     CONFIG_SND_MODULE))
&& (FILE_sound_soc_au1x_i2sc.c -> ((CONFIG_SND || CONFIG_SND_MODULE)
    && (CONFIG_SND_SOC || CONFIG_SND_SOC_MODULE)
    && (CONFIG_SND_SOC_AU1XI2SC || CONFIG_SND_SOC_AU1XI2SC_MODULE)))
&& (CONFIG_SND_AC97_CODEC -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML &&
     CONFIG_SND) && !CONFIG_SND_AC97_CODEC_MODULE && ((((CONFIG_SOUND_MODULE ||
     CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE ||
     CONFIG_SND))) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_AC97_CODEC_MODULE -> (((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND)) && !
     CONFIG_SND_AC97_CODEC && CONFIG_MODULES && ((((CONFIG_SOUND_MODULE ||
     CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE ||
     CONFIG_SND))) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_CA0106 -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML &&
     CONFIG_SND && CONFIG_SND_PCI) && !CONFIG_SND_CA0106_MODULE && ((((
     CONFIG_SOUND_MODULE || CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (
     CONFIG_SND_MODULE || CONFIG_SND) && CONFIG_SND_PCI)) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_CA0106_MODULE -> (((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND) &&
     CONFIG_SND_PCI) && !CONFIG_SND_CA0106 && CONFIG_MODULES && ((((
     CONFIG_SOUND_MODULE || CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (
     CONFIG_SND_MODULE || CONFIG_SND) && CONFIG_SND_PCI)) -> CONFIG_SND_VMASTER)))
&& !CONFIG_SND_SOC_AU1XI2SC && !CONFIG_SND_SOC_AU1XI2SC_MODULE
&& (CONFIG_SND_HDA -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML && CONFIG_SND)
     && !CONFIG_SND_HDA_MODULE && ((((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND))) ->
```

```
#B00:sound/soc/au1x/i2sc.c:0:0:sound/soc/au1x/i2sc.c:0:0:
B00
&&
( B0 <-> CONFIG_PM )
&& ( B1 <-> ( ! (B0) )
...
&& (CONFIG_FUTEX -> (CONFIG_RT_MUTEXES))
&& (CONFIG_GENERIC_IO -> ((CONFIG_HAS_IOMEM)))
&& (CONFIG_HAS_IOMEM -> (!CONFIG_NO_IOMEM && ((!CONFIG_NO_IOMEM) ->
     CONFIG_GENERIC_IO) && (!CONFIG_NO_IOMEM)))
&& (CONFIG_I2C -> (!CONFIG_I2C_MODULE && CONFIG_RT_MUTEXES))
&& (CONFIG_I2C_MODULE -> (!CONFIG_I2C && CONFIG_MODULES && CONFIG_RT_MUTEXES))
&& (CONFIG_INPUT -> (!CONFIG_UML && !CONFIG_INPUT_MODULE))
&& (CONFIG_INPUT_MODULE -> (!CONFIG_UML && !CONFIG_INPUT && CONFIG_MODULES))
&& (CONFIG_RT_MUTEXES -> ((CONFIG_FUTEX || CONFIG_I2C || CONFIG_I2C_MODULE)))
&& (CONFIG_SND -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML) && !
     CONFIG_SND_MODULE))
&& (FILE_sound_soc_au1x_i2sc.c -> ((CONFIG_SND || CONFIG_SND_MODULE)
    && (CONFIG_SND_SOC || CONFIG_SND_SOC_MODULE)
    && (CONFIG_SND_SOC_AU1XI2SC || CONFIG_SND_SOC_AU1XI2SC_MODULE)))
&& (CONFIG_SND_AC97_CODEC -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML &&
     CONFIG_SND) && !CONFIG_SND_AC97_CODEC_MODULE && ((((CONFIG_SOUND_MODULE ||
     CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE ||
     CONFIG_SND))) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_AC97_CODEC_MODULE -> (((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND)) && !
     CONFIG_SND_AC97_CODEC && CONFIG_MODULES && ((((CONFIG_SOUND_MODULE ||
     CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE ||
     CONFIG_SND))) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_CA0106 -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML &&
     CONFIG_SND && CONFIG_SND_PCI) && !CONFIG_SND_CA0106_MODULE && ((((
     CONFIG_SOUND_MODULE || CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (
     CONFIG_SND_MODULE || CONFIG_SND) && CONFIG_SND_PCI)) -> CONFIG_SND_VMASTER)))
&& (CONFIG_SND_CA0106_MODULE -> (((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND) &&
     CONFIG_SND_PCI) && !CONFIG_SND_CA0106 && CONFIG_MODULES && ((((
     CONFIG_SOUND_MODULE || CONFIG_SOUND) && !CONFIG_M68K && !CONFIG_UML && (
     CONFIG_SND_MODULE || CONFIG_SND) && CONFIG_SND_PCI)) -> CONFIG_SND_VMASTER)))
&& !CONFIG_SND_SOC_AU1XI2SC && !CONFIG_SND_SOC_AU1XI2SC_MODULE
&& (CONFIG_SND_HDA -> ((CONFIG_SOUND && !CONFIG_M68K && !CONFIG_UML && CONFIG_SND)
     && !CONFIG_SND_HDA_MODULE && ((((CONFIG_SOUND_MODULE || CONFIG_SOUND) && !
     CONFIG_M68K && !CONFIG_UML && (CONFIG_SND_MODULE || CONFIG_SND))) ->
```

MUS-Analysis

- New "-u" flag for the undertaker:
  - Calculate the `minimal unsatisfiable subset` (MUS) after dead analysis
- Based on the PicoMUS Tool (part of the PicoSAT Toolchain[1])
- MUS-report:

ATTENTION: This formula _might_ be incomplete or even inconclusive!
Minimized Formula from:
p cnf 2794 6581
to
p cnf 2794 559

```
(B00)
^ (CONFIG_SND_SOC_AU1XI2SC v CONFIG_SND_SOC_AU1XI2SC_MODULE)
^ (!FILE_sound_soc_au1x_i2sc.c)
^ (!CONFIG_SND_SOC_AU1XI2SC)
^ (!CONFIG_SND_SOC_AU1XI2SC_MODULE)
^ (!B00 v FILE_sound_soc_au1x_i2sc.c)
```

[1] http://fmv.jku.at/picosat/

Example: Impact of an Unreachable Defect

- Force disable `unreachable symbols` in dead analysis
- Newly found defect:
  `sound/soc/au1x/i2sc.c.B00.kbuild.globally.dead`
  - MUS-Formula:

```
(B00) ^ (CONFIG_SND_SOC_AU1XI2SC) ^
(!FILE_sound_soc_au1x_i2sc.c) ^ (!CONFIG_SND_SOC_AU1XI2SC)
^ (!B00 v FILE_sound_soc_au1x_i2sc.c)
```

  - File preconditions:

```
FILE_sound_soc_au1x_i2sc.c "... && CONFIG_SND_SOC_AU1XI2SC"
```

  - Symbol unreachable: `SND_SOC_AU1XI2SC`

Example: Impact of an Unreachable Defect

- Force disable `unreachable symbols` in dead analysis
- Newly found defect:
  `sound/soc/au1x/i2sc.c.B00.kbuild.globally.dead`
  - MUS-Formula:

```
(B00) ^ (CONFIG_SND_SOC_AU1XI2SC) ^
(!FILE_sound_soc_au1x_i2sc.c) ^ (!CONFIG_SND_SOC_AU1XI2SC)
^ (!B00 v FILE_sound_soc_au1x_i2sc.c)
```

  - File preconditions:

```
FILE_sound_soc_au1x_i2sc.c "... && CONFIG_SND_SOC_AU1XI2SC"
```

  - Symbol unreachable: `SND_SOC_AU1XI2SC`

Conclusion

- The whole file is dead (323 lines of code)
- Introduced with commit b3c70c9e (2011-07-25)

Summary and Conclusions

- Found logical, missing and undead defects in KConfig
  - Identified more dead code
  - Identified more redundant code
  - Maybe even functional defects
- Logical defects are very rare in KConfig
  - Fixed the one logical defect in the current `linux-next`
- Undertaker–CheckPatch now warns if options become `unreachable`
- MUS-Analysis helps to identify defects

Questions?

stefan.hengelein@fau.de