Analyzing all $2^{10000}$ configurations of the Linux kernel

10,000 features, 6 million lines of C code

```c
static void rt_mutex_init_task(struct task_struct *p) {
    raw_spin_lock_init(&p->pi_lock);
    #ifdef CONFIG_RT_MUTEXES
    plist_head_init_raw(&p->pi_waiters, &p->pi_lock);
    p->pi_blocked_on = NULL;
    #endif
}
```
Highly Configurable Software Systems
(e.g., Software Product Lines)
Processor type and features

Arrow keys navigate the menu.  <Enter> selects submenus --->.  Highlighted letters are hotkeys.  Pressing <Y> includes, <N> excludes, <M> modularizes features.  Press <Esc><Esc> to exit, <?> for Help, </> for Search.  Legend: [*] built-in [ ] excluded <M> module < > module capable

[ ] Tickless System (Dynamic Ticks)
[ ] High Resolution Timer Support
[ ] Symmetric multi-processing support
[ ] Support for extended (non-PC) x86 platforms
[ ] Single-depth WCHAN output
[ ] Paravirtualized guest support --->
[ ] Memtest
  Processor family (Generic-x86-64) --->
  Preemption Model (No Forced Preemption (Server)) --->
[ ] Reroute for broken boot IRQs (NEW)
[ ] Machine Check / overheating reporting
[ ] Dell laptop support
[ ] /dev/cpu/microcode - microcode support
[ ] /dev/cpu/*/msr - Model-specific register support
[ ] /dev/cpu/*/cpuid - CPU information support
  Memory model (Sparse Memory) --->
[*] Sparse Memory virtual memmap (NEW)
[ ] Allow for memory hot-add (NEW)
[ ] Enable KSM for page merging
(4096) Low address space to protect from user allocation
[ ] Check for low memory corruption
[ ] Reserve low 64K of RAM on AMI/Phoenix BIOSen
-[*] MTRR (Memory Type Range Register) support
  MTRR cleanup support
[ ] Enable seccomp to safely compute untrusted bytecode
[ ] Enable -fstack-protector buffer overflow detection (EXPERIMENTAL)
  Timer frequency (250 HZ) --->
[ ] kexec system call

<<Select>>  << Exit >>  << Help >>
Variability $\sim$ Complexity
33 optional, independent features

a unique configuration for every person on this planet
320 optional, independent features

more configurations than estimated atoms in the universe
2000 features

10000 features
Correctness?
Checking Products

2000 Features
100 Printers
30 New Printers per Year

Printer Firmware
Checking Product Line Implementation with 10000 Features + Generator

Linux Kernel
Variability-Aware Analysis

Parser
Type System
Static Analysis
Bug Finding
Testing
Model Checking
Theorem Proving
...

...
Variability-Aware Analysis

Product Generation

Conventional Analysis

Variability-Aware Analysis → Product Generation → Conventional Analysis
static int _rep_queue_filedone(
    DB_ENV *dbenv,
    REP *rep,
    __rep_fileinfo_args *rfp) {
#ifdef NO_QUEUE
    COMPQUIET(rep, NULL);
    COMPQUIET(rfp, NULL);
    return (__db_no_queue_am(dbenv));
#else
    db_pgno_t first, last;
    u_int32_t flags;
    int empty, ret, t_ret;
#ifdef DIAGNOSTIC
    DB_MSGBUF mb;
#endif
    // over 100 lines of add. code
#endif
}
#endif
40 Open-Source C Projects

Number of features vs Lines of code

apache, berkely db, cherokee, clamav, dia, emacs, freebsd, gcc, ghostscript, gimp, glibc, gnumeric, gnuplot, irssi, libxml, lighttpd, linux, lynx, minix, mplayer, mpsolve, openldap, opensolaris, openvpn, parrot, php, pidgin, postgresql, privoxy, python, sendmail, sqlite, subversion, sylpheed, tcl, vim, xfig, xine-lib, xorg-server, xterm

[ICSE’10, AOSD’11]
Correctness?
Variability-Aware Analysis

Parser
Type System
Static Analysis
Bug Finding
Testing
Model Checking
Theorem Proving
...

...
#include <stdio.h>

#ifndef WORLD
char * msg = "Hello_World\n";
#endif
#ifndef BYE
char * msg = "Bye_bye!\n";
#endif

main() {
    printf(msg);
}
#include <stdio.h>

#ifdef WORLD
char * msg = "Hello_World\n";
#endif
#ifdef BYE
char * msg = "Bye_bye!\n";
#endif

main() {
    printf(msg);
}
```c
#include <stdio.h>

#ifdef WORLD
char * msg = "Hello_World\n";
#endif

#ifdef BYE
char * msg = "Bye_bye!\n";
#endif

main() {
    printf(msg);
}
```
Reachability: \( pc(\text{caller}) \rightarrow pc(\text{target}) \)
Conflicts: \( \neg (pc(\text{def1}) \land pc(\text{def2})) \)

Found 2 type errors:
- [WORLD & BYE] file hello.c:8:8
  redefinition of msg
- [!]WORLD & !BYE] file hello.c:11:8
  msg undeclared
Variability Model:

```
#include <stdio.h>

#ifndef WORLD
char * msg = "Hello_World\n";
#endif

#ifndef BYE
char * msg = "Bye_bye!\n";
#endif

main() {
    printf(msg);
}
```

- `VM -> (true -> true)`
- `VM -> (true -> (WORLD v BYE))`
- `VM -> \neg (WORLD \land BYE)`
Variability-Aware Analysis

Product Generation

Conventional Analysis
```c
#include <stdio.h>

#ifdef WORLD
  char * msg = "Hello, World\n";
#else
  char * msg = "Hello, World\n";
#endif

#ifdef BYE
  char * msg = "Bye, bye!\n";
#else
  char * msg = "Hello, World\n";
#endif

main() {
  printf(msg);
}
```

Extended Lookup Mechanism
int put_eol(fd)
    FILE *fd;
{
    if (  
        #ifdef USE_CRNL          
            !(putc(\r, fc) < 0)) || 
        #ifdef MKSESSION_NL      
            !mksession_nl && 
        #endif
    if (n =  
        #endif
    return
        (putc(\n, fd) < 0))    
    return FAIL;
    return OK;
```c
#include <stdio.h>

#ifdef WORLD
char * msg = "Hello_World\n";  
#endif

#ifdef BYE
char * msg = "Bye_bye!\n"; 
#endif

main() {
    printf(msg);
}
```
Library of Variability-Aware Parser Combinators in Scala

([OOPSLA'11])
TypeChef

Variability-Aware Lexer

Variability-Aware Parser

Build System Analysis

Feature Model Analysis

[OOPSLA'11, OOPSLA'12]

https://github.com/ckaestne/TypeChef
TypeChef

Variability-Aware Lexer

Variability-Aware Type Syst.

Variability-Aware Control/ Data-Flow Simulator Library

Variability-Aware Bug Finding

Variability-Aware Linker

Variability-Aware Interpreter

Variability-Aware Refactoring

Variability-Aware Metrics

Feature Model Analysis

#ifdef A
#define X 4
#else
#define X 5
#endif

(2*3)+X

https://github.com/ckaestne/TypeChef
0 C files (x86)
0 included header files per C file
0 distinct macros per C file
lines of processed C code
0 % conditional
0 seconds per file (median)
7665
353
8590
899 M
2.6.33.3 X86
struct globals {
    double cur_time;
    //... skipped 11 lines
#if ENABLE_FEATURE_NTPD_SERVER
    int listen_fd;
#endif
    unsigned verbose;
    //... skipped 73 lines
};

int ntpd_main(int argc UNUSED_PARAM, char **argv) {
    #undef G
    struct globals G;
    //... skipped 81 lines
    if (i > (ENABLE_FEATURE_NTPD_SERVER && G.listen_fd != -1)) {

    }

ntpd.c: 2128
[CONFIG_NTPD && !CONFIG_FEATURE_NTPD_SERVER]
field listen_fd unknown in struct globals
Analyzing all $2^{10000}$ Configurations of the Linux Kernel

https://github.com/ckaestne/TypeChef