به نام خدا

روش‌های رسمی در مهندسی نرم‌افزار

مرجان عظیمی
Azimi.marjan@gmail.com

مرجع

- Baier, Christel, Joost-Pieter Katoen, and Kim Guldstrand Larsen. *Principles of model checking*. MIT press, 2008.

ارزشیابی

۳۰٪	امتحان میان‌ترم
۳۵٪	امتحان پایان‌ترم
۲۰٪	ارایه
۵٪	فعالیت کلاسی
۱۰٪	تمرین

سیستم‌های ICT(Information and Communication Technology)

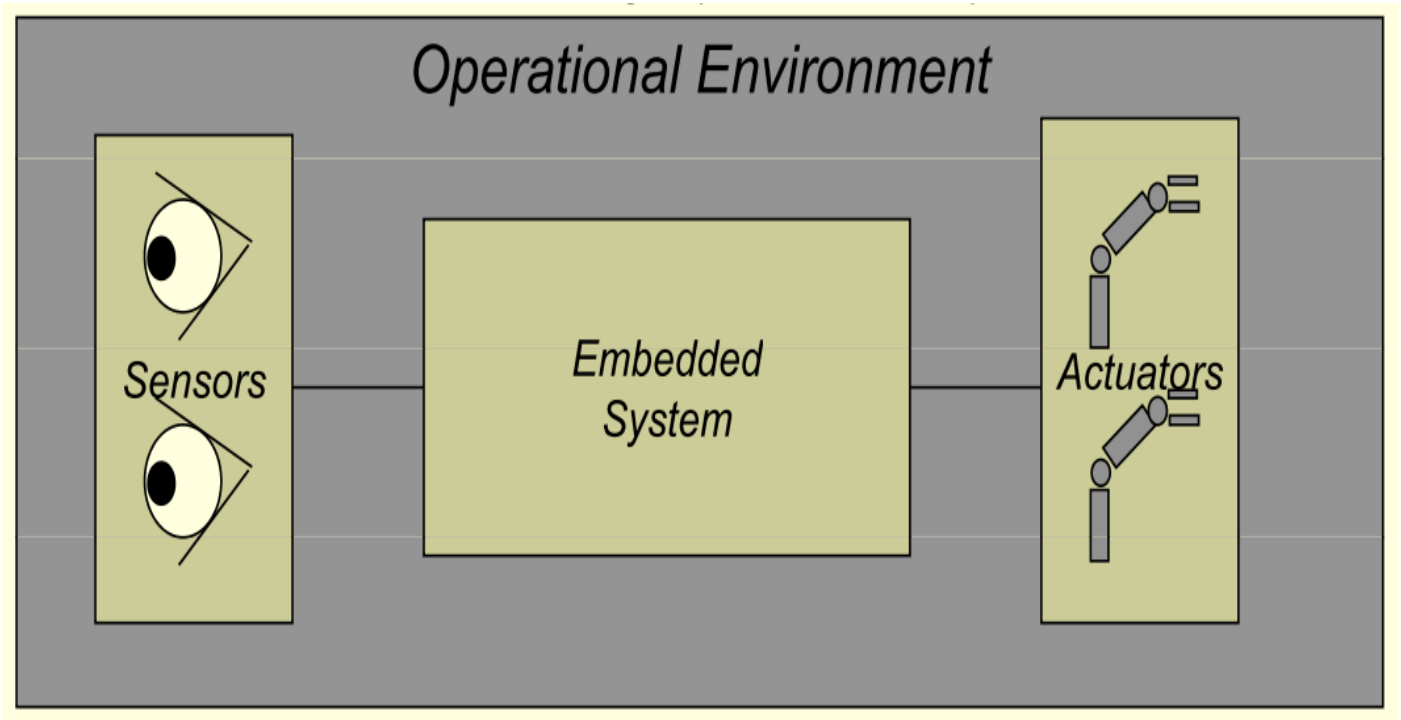
- خطا در دستگاه ویدئو
- خطا در چک چمدان مسافران فرودگاه

قابل اعتماد بودن سیستم‌های ICT یک نکته کلیدی در طراحی آن‌ها می‌باشد.

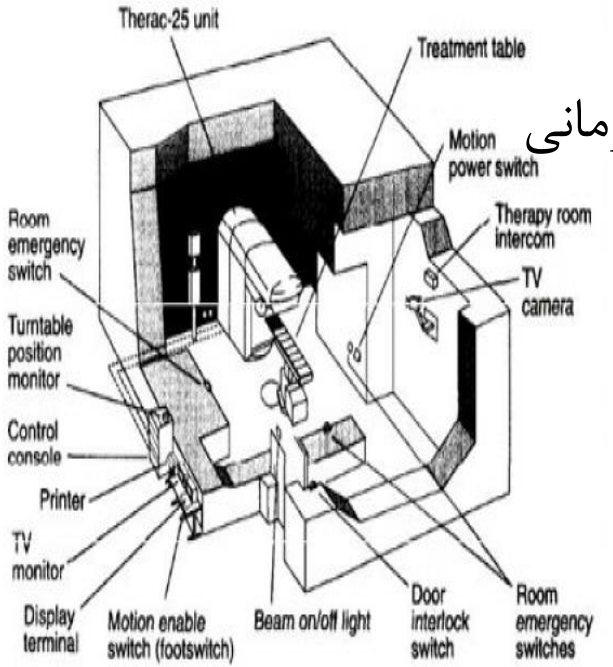
سیستم‌های نهفته

خاص منظوره هستند.

معمولا با کاربر تعامل ندارند یا تعامل محدودی دارند.

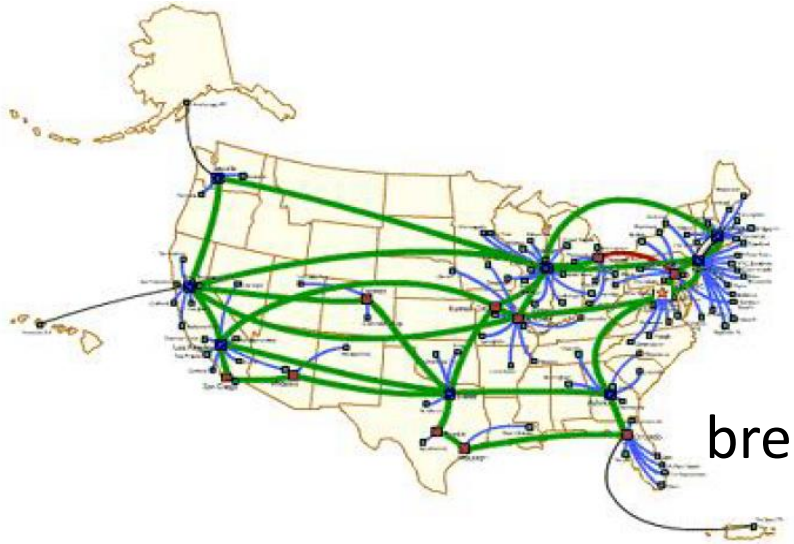


خطای نرم‌افزاری



- ماشین پرتو برای درمان بیماران سرطانی
- حداقل در شش نمونه، تابش بیش از حد در بازه زمانی ۱۹۸۵ تا ۱۹۸۷
- سه بیمار سرطانی فوت کردند.
- علت مشکل: وجود خطا در بخش کنترل نرم‌افزار

خطای نرم‌افزاری



- در سال ۱۹۹۰ مشکلی در شهر نیویورک به وجود آمد که منجر به قطع ۹ ساعتی شبکه تلفن گردید.
- منجر به خسارت ۱۰۰ میلیون دلاری گردید.
- منبع خطا: تفسیر نادرست از جمله break در زبان c

خطای نرم‌افزاری

- سقوط موشک در سال ۱۹۹۶
- خسارت بیش از ۵۰۰ میلیون دلار
- منبع خطا: غیر فعال شدن کنترل نرم‌افزار

سیستم‌های نهفته

- ماشین‌های پیشرفته
- دستگاه‌های پزشکی
- تلفن همراه
- حمل و نقل
- سیستم‌های ترافیک کنترل

تلاش برای صحت نرم‌افزار

Henk Barendregt

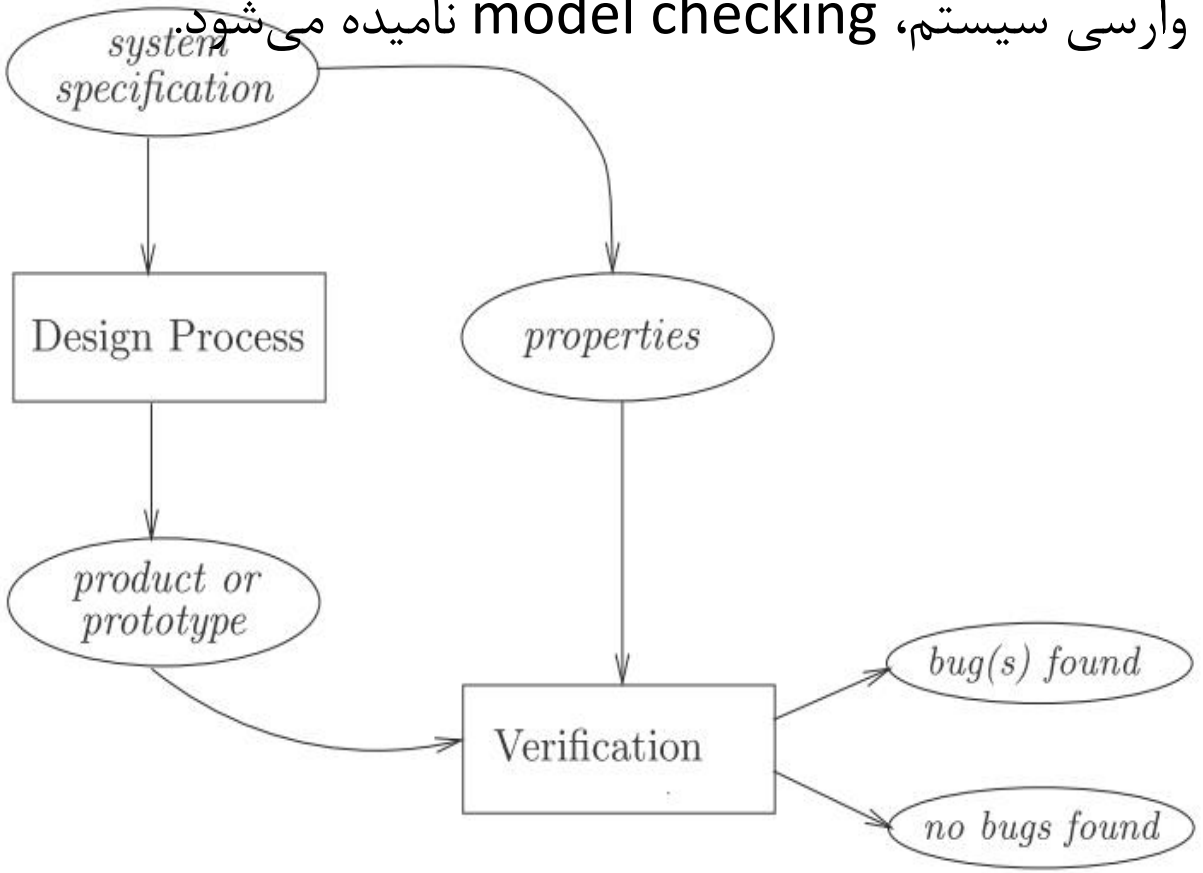
,, این منصفانه است که بگوییم در دنیای دیجیتال برای پردازش اطلاعات، داشتن سیستم صحیح (correct systems) با ارزش‌تر از طلا است.،،

وارسی سیستم(System Verification)

- وارسی سیستم برای تصدیق این که طراحی یا تولید ویژگی‌های خاصی که مد نظر ما هست را برآورده می‌کند یا خیر.
- Specification: تعیین می‌کند که سیستم چه کارهایی را باید انجام دهد و چه کارهایی را نباید.
- زمانی که عیبی در سیستم یافت می‌شود، یعنی یکی از ویژگی‌های توصیف برآورده نشده است.

وارسی سیستم

- از این پس وارسی سیستم، model checking نامیده می‌شود.



وارسی نرم‌افزار(Software Verification)

- Peer reveiwing
  - وارسی دقیق توسط تیمی از مهندسین
  - کد اجرا نمی‌شود بلکه به صورت کاملا ایستا تحلیل می‌گردد.
  - تشخیص خطاهای الگوریتمی و همروندی توسط این روش دشوار است.

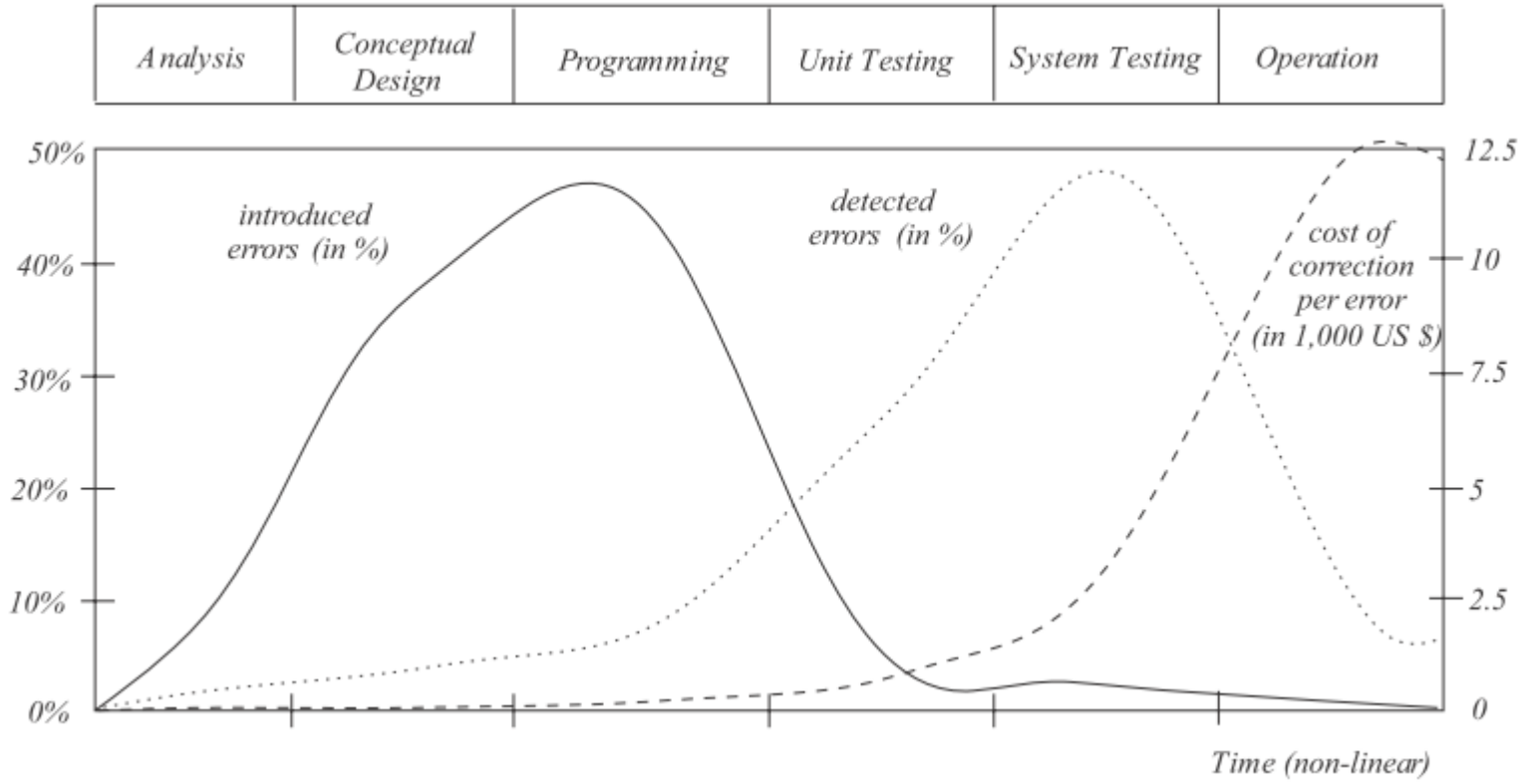
- Testing
  - برنامه به صورت پویا اجرا می‌شود.
  - بین ۳۰ تا ۵۰ درصد هزینه‌های هر پروژه به تست تخصیص می‌یابد.
  - می‌تواند روی انواع نرم‌افزارها اعمال شود.
  - تست تمامی مسیرهای اجرا غیر ممکن است.
  - تست فقط می‌تواند وجود خطا را تشخیص دهد و نه نبود آن را.
  - مشکل تست تشخیص زمان توقف آن است.

وارسی نرم‌افزار(Software Verification)

هزینه تعمیر نرم‌افزار در دوره نگهداری ۵۰۰ بار بیشتر از هزینه تعمیر در دوره طراحی است.

وارسی سیستم باید هرچه زودتر در مراحل طراحی انجام شود.

چرخه نرم‌افزار و پیدایش و شناسایی خطا



وارسی سخت‌افزار(Hardware Verification)

- Emulation : مانند تست است. یک پیکربندی مجدد از سیستم‌های سخت افزاری است. طوری پیکربندی می‌شوند که تحت شرایط خاص رفتار مشابهی داشته باشد. مشکلات روش تست را دارد.

- Simulation: یک مدل از مدار شبیه‌سازی می‌شود که برای شبیه سازی معمولا از زبان‌های توصیف سخت افزار استفاده می‌شود. ورودی‌ها یا توسط کاربر تعریف می‌شوند یا به صورت تصادفی تولید می‌شوند.

- Structural analysis

روش‌های رسمی(Formal Methods)

- روش رسمی: برای مدل‌سازی و تحلیل سیستم از ریاضیات استفاده می‌کند.
- سعی می‌کند وارسی را هر چه زودتر و در مراحل طراحی انجام دهد.
- تکنیک‌های وارسی کارایی را معرفی می‌نماید.
- زمان وارسی را کاهش می‌دهد.

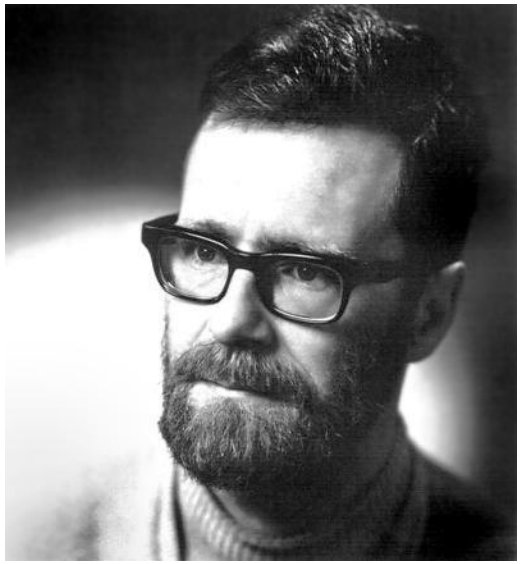
روش‌های رسمی

- هدف اصلی مهندسی نرم‌افزار گسترش سیستم‌های قابل اعتماد است.

- روش‌های رسمی
  - ✓ از زبان‌های ریاضی، ابزارها و تکنیک‌ها برای ایجاد و تایید سیستم استفاده می‌شود.
  - ✓ هدف: کمک به مهندسین نرم‌افزار برای ایجاد سیستم‌های قابل اعتماد

نقل قول دانشمندان

آموزش استفاده موثر از روش‌های رسمی به جوانان یکی از شادی‌های زندگی است به این دلیل که بسیار با ارزش است.



نقل قول دانشمندان

مهندسان نرم‌افزار می‌خواهند که مهندسین واقعی باشند.

مهندسین واقعی از ریاضیات استفاده می‌کنند.

روش‌های رسمی ریاضیات مهندسی نرم‌افزار می‌باشند.

بنابراین مهندسین نرم‌افزار باید از روش‌های رسمی استفاده نمایند.

روش‌های رسمی در سال‌های گذشته

- نمادهای مبهم
- تکنیک‌های غیر مقیاس پذیر
- ابزارهای پشتیبانی ناکافی
- کار کردن سخت با ابزارها
- مطالعات انجام شده بسیار کم

روش‌های رسمی در حال حاضر

- تلاش برای پیدا کردن نمادهای دقیقتر
- چک کردن مدل(model cheking)
- مطالعه موردی نمونه‌های صنعتی بیشتر
- تلاش محققان برای دست‌یابی به مزایای استفاده از روش‌های رسمی

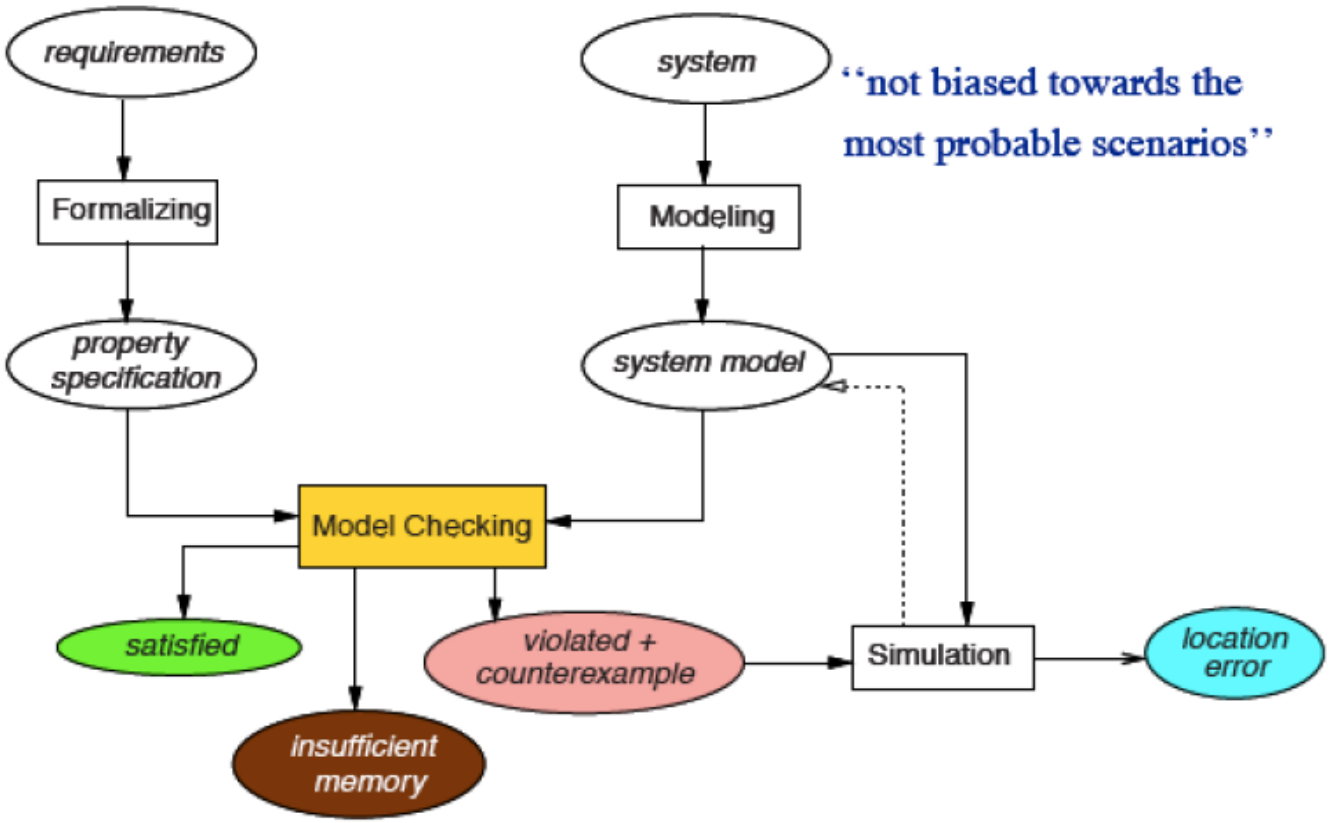
Model Checking

- چک کردن مدل، یک تکنیک خودکار است که یک مدل finite-state از سیستم و ویژگی‌هایش می‌دهد و سپس به طور سیستمی چک می‌کند که آیا این خصوصیات برای این مدل حفظ می‌شود یا خیر.

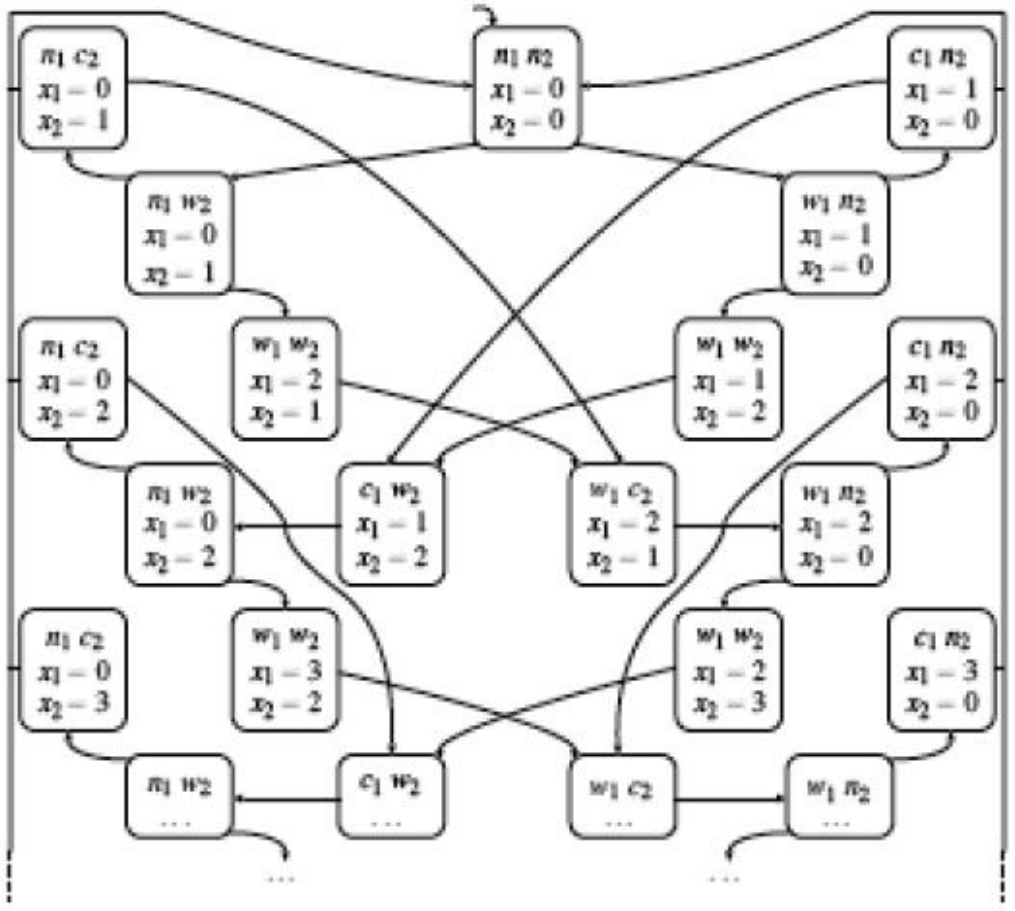
Model Checking

- بر مبنای توصیف رفتاری سیستم به صورت ریاضی و بدون ابهام است.
- مدل کردن دقیق سیستم منجر به شناسایی ناکاملی‌ها و ابهامات می‌شود.
- یک تکنیک وارسی است که تمام حالت‌های ممکن رفتار سیستم را بررسی می‌کند. مثل شطرنج
- چالش اساسی بررسی مدل‌هایی با فضای حالت زیاد است.
- این مدل می‌تواند فضای حالت 10^8 تا 10^9 نمونه را پشتیبانی کند.
- خطاهای نامحسوسی که با استفاده از تست و یا شبیه‌سازی پنهان می‌مانند با استفاده از این مدل پوشش داده می‌شوند.

Model Checking



مدل چیست؟


$n_1\ c_2$ $x_1 = 0$ $x_2 = 1$
$n_1\ n_2$ $x_1 = 0$ $x_2 = 0$
$c_1\ n_2$ $x_1 = 1$ $x_2 = 0$
$n_1\ w_2$ $x_1 = 0$ $x_2 = 1$
$w_1\ n_2$ $x_1 = 1$ $x_2 = 0$
$n_1\ c_2$ $x_1 = 0$ $x_2 = 2$
$w_1\ w_2$ $x_1 = 2$ $x_2 = 1$
$w_1\ w_2$ $x_1 = 1$ $x_2 = 2$
$c_1\ n_2$ $x_1 = 2$ $x_2 = 0$
$n_1\ w_2$ $x_1 = 0$ $x_2 = 2$
$c_1\ w_2$ $x_1 = 1$ $x_2 = 2$
$w_1\ c_2$ $x_1 = 2$ $x_2 = 1$
$w_1\ n_2$ $x_1 = 2$ $x_2 = 0$
$n_1\ c_2$ $x_1 = 0$ $x_2 = 3$
$w_1\ w_2$ $x_1 = 3$ $x_2 = 2$
$w_1\ w_2$ $x_1 = 2$ $x_2 = 3$
$c_1\ n_2$ $x_1 = 3$ $x_2 = 0$
$n_1\ w_2$ ...
$c_1\ w_2$ ...
$w_1\ c_2$ ...
$w_1\ n_2$ ...

مدل چیست؟

Transition System

حالت (state): توصیف کننده روابط، مجموعه‌ها و توابع هستند. اطلاعاتی است در مورد مقادیر فعلی یک متغیر، اجرای قبلی یک جمله و مانند این‌ها

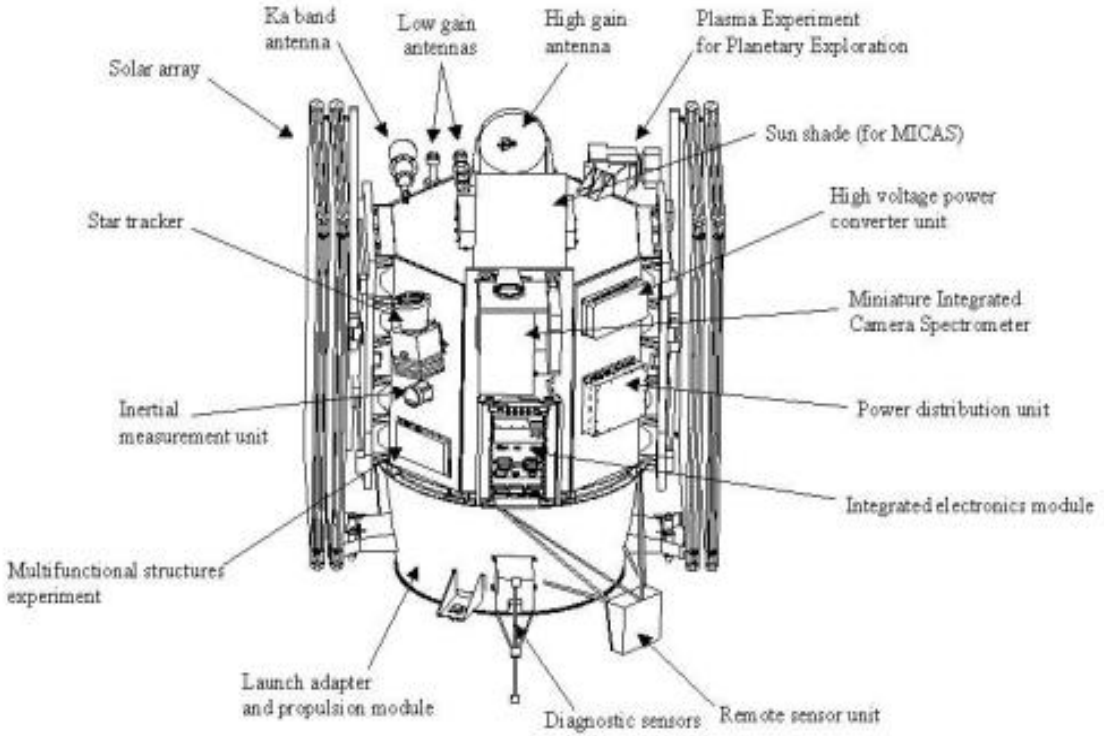
گذار(Transition) : می‌گوید چطور یک سیستم از یک حالت به حالت دیگر می‌رود.

Properties

- سیستم آن چه که باید انجام دهد را انجام می‌دهد؟(functional correctness)
- در بن‌بست می‌تواند پایان یابد؟(reachability)
- چیزهای بد هرگز اتفاق نیفتد؟(safety)
- چیزهای خوب همیشه رخ دهد؟(liveness)
- آیا برنامه خروجی درست را مهیا می‌کند؟
- آیا دو فرآیند می‌توانند همزمان در ناحیه بحرانی باشند؟

برای توصیف مشخصات از Temporal logic استفاده می‌کنیم.

NASA’s Deep Space-1 Spacecraft



یک قطعه کوچک از برنامه

process Inc = **while** *true* **do if** $x < 200$ **then** $x := x + 1$ **od**

process Dec = **while** *true* **do if** $x > 0$ **then** $x := x - 1$ **od**

process Reset = **while** *true* **do if** $x = 200$ **then** $x := 0$ **od**

is x always between (and including) 0 and 200?

شاخصه‌های Model Checking

- فاز مدلسازی(modeling phase)
  - مدل کردن سیستم مد نظر
  - انجام یکسری شبیه‌سازی
  - فرموله کردن ویژگی‌هایی که باید چک شوند.
- فاز اجرا(Running phase)
  - مدل چکر اجرا می‌شود تا اعتبار ویژگی‌ها سنجیده شود.
- فاز تحلیل(analysis phase)
  - اگر خطا نداریم: ویژگی بعدی را بررسی می‌کنیم.
  - اگر خطا داریم:
    - مثال نقض را با شبیه‌ساز بررسی می‌کنیم.
    - مدل، طراحی یا ویژگی‌ها را دوباره تعریف می‌کنیم. و دوباره کار را تکرار می‌کنیم.
  - اگر با کمبود حافظه مواجه شدیم: سعی کنیم مدل را کاهش دهیم و دوباره کار را تکرار کنیم.

وارسی سیستم(System Verification)

- وارسی سیستم میزانی است برای بررسی این‌که آیا سیستم نیازمندی‌های کیفی شناسایی شده را برآورده می‌کند یا خیر.

Verification ≠ Validation

- Verification = “check that we are building the thing right”
- Validation = “check that we are building the right thing”

نقاط قوت model checking

- روی رنج وسیعی از کاربردها قابل اعمال است.
- هر مشخصه می‌تواند به تنهایی بررسی شود. بنابراین می‌شود روی یک ویژگی خاص متمرکز شد.
- بر مبنای گراف، ساختمان داده و منطق است.
- باعث می‌شود زمان توسعه کوتاهتر شود.

نقاط ضعف model checking

- برای ابزارهای کنترلی است و نه داده‌ای زیرا دامنه بی‌نهایت می‌شود.
- برای سیستم‌های تصمیم پذیر کاربرد دارد.
- به تمرین زیاد برای به دست آوردن فضای حالت کم نیاز دارد.

محاسن و معایب روش‌های رسمی

- محاسن
  - ✓ کاهش نرخ خطا
  - ✓ افزایش اطمینان به سامانه
  - ✓ کاهش هزینه در نگاه کلی نگر
  - ✓ استفاده مجدد در حد توصیف

- معایب
  - ✓ افزایش هزینه اولیه
  - ✓ زمان بیشتر برای یادگیری
  - ✓ عدم درک توسط افراد غیر فنی

مطالعه موردی: CICS

- CICS(Customer Information Control System) یک سیستم پردازش تراکنش برخط است.
- در سال ۱۹۸۰ دانشگاه آکسفورد و IBM بخشی از CICS را با استفاده از زبان z رسمی کردند.
- باعث بهبودهای چشم‌گیری در کیفیت محصول گردید.
- تخمین زده شده است که با این روش ۹ درصد از هزینه‌های توسعه کاهش یافته است.

مطالعه موردی: NASA SATS

- هواپیماهای کوچک سیستم حمل و نقل
- استفاده از یک سیستم نرم افزاری است که توالی هواپیماها را به حریم هوایی SATS در غیاب کنترل فرودگاه انجام می‌دهد.
- مسایل ایمنی جدی در ارتباط با این سیستم‌های نرم‌افزاری و الگوریتم‌های کلیدی مربوطه وجود دارد.

مطالعه موردی: NASA SATS

- وضعیت بحرانی چنین سیستم‌هایی مستلزم این است که تضمینی قوی از ایمنی برای آن‌ها داده شود.

- محققان ناسا در پی تایید این نرم‌افزار با استفاده از روش‌های رسمی می‌باشند.
  - ✓ مدل کردن و تایید ترافیک هوایی
  - ✓ کشف تصادف و اخطار