

استفاده از امکان‌سنجی معیار منطق برای کاهش اندازه تست با تضمین تشخیص

نقص^۱

چکیده

برخی از معیارهای پوشش تست منطق نرم‌افزار نیازمند ورودی‌هایی^۲ هستند که تضمین‌کننده تشخیص مجموعه بزرگی از انواع نقص‌ها دارند. یکی از معیارهای قدرتمند در این زمینه، MUMCUT^۳ است که از سه معیار تشکیل شده است که در آن هر مولفه تشخیص نوع خاصی از نقایص را تضمین می‌کند. در عمل، معیارها ممکن است با توجه به نوع نقصی که تشخیص داده شده است با یکدیگر همپوشانی داشته باشند در نتیجه این موضوع منجر به آزمون‌های زائد متعددی خواهد شد، اما به دلیل این حقیقت ناخوشایند که نیازمندی‌های آزمون غیرعملی در آزمون‌ها مشخص نمی‌شود، تمامی معیارهای سازنده مورد نیاز است. بینش کلیدی این مقاله در تجزیه و تحلیل امکان‌سنجی معیار تشکیل‌دهنده‌ای است که می‌تواند برای کاهش اندازه مجموعه آزمون بدون فداکردن تشخیص نقص مورد استفاده قرار بگیرد. به عبارت دیگر، معیارهای گران را می‌توان برای مواقعی رزرو کرد که استفاده از آن‌ها واقعا ضروری باشد. در این مقاله یک معیار منطقی جدیدی با نام، کمینه^۳ MUMCUT، بر اساس بینش مقاله معرفی شده است. با توجه به گزاره داده شده در کمینه^۴ DNF، تعیین می‌کند که کدام یک از معیارهای سازنده در سطح اصطلاحات و لیترال‌ها^۴ منحصر به فرد امکان‌پذیر است. این موضوع به نوبه خود تعیین می‌کند که کدام معیارها مجددا در سطح اصطلاحات و لیترال‌ها منحصر به فرد ضروری هستند. در این مقاله یک مطالعه تجربی با استفاده از

¹ fault

² inputs

³ Minimal-MUMCUT

⁴ literals

گزاره‌ها در نرم‌افزار اویونیک^۵ ارائه شده است. این مطالعه نشان می‌دهد که کمینه MUMCUT سبب کاهش اندازه مجموعه آزمون تا حدی می‌شود که تنها چند درصد از اندازه مجموعه آزمون در صورتی که امکان‌پذیری در نظر گرفته نشده باشد، مورد نیاز است و تشخیص نقص را نیز فدای این موضوع نمی‌کند.

کلمات کلیدی: تست منطق نرم‌افزار، نقص نرم‌افزار، معیار، MUMCUT، فرم نرمال اشتراکی^۶.

1. مقدمه

نیازمندی‌های^۷ آزمون غیرممکن^۸ تقاضاهایی برای آزمون‌ها هستند که به سادگی وجود ندارند. آن‌ها حقایقی ناخوشایند از زندگی در آزمون نرم‌افزار هستند. آن‌ها مهندسان آزمون را گیج می‌کنند، مهندسان آزمون باید تصمیم بگیرند که آیا نیازمندی آزمونی معین واقعا غیرممکن است یا اگر یک جستجوی دقیق‌تر برای ورودی مناسب در سفارش است. آن‌ها همچنین تلاش‌های محققان برای ارتباط معیارهای پوشش را نیز به سردرگمی می‌کشاند. پس بر اساس تعریف، نیازمندی آزمون غیرممکن برای یک معیار معین، نتیجه آزمون را نمی‌دهد. اگر وقوع نیازمندی آزمون متناظر برای یک معیار «ضعیف‌تر» ممکن باشد، امکان‌ناپذیری می‌تواند معیار ظاهرا «قوی‌تری» برای شکست در گنجاندن «ضعیف‌تر» باشد. بسیاری از موارد معروف از این پدیده در پیشینه تحقیق انجام آزمون یافت می‌شود. در این مقاله، ما به امکان‌ناپذیری در زمینه معیارهای آزمون منطقی طراحی‌شده برای سلسله‌مراتب^۹ نقض لاو^{۱۰} و یو^{۱۱} خواهیم پرداخت [9].

⁵ avionics

⁶ Disjunctive Normal Form

⁷ requirements

⁸ Infeasible

⁹ hierarchy

¹⁰ Lau

¹¹ Yu

ما آزمون گزاره‌ها را بر روی متغیرهای بولین¹² در انزوا¹³ در نظر می‌گیریم. در این دامنه محدود، تعیین اینکه آیا نیازمندی آزمون امکان‌پذیر است امری ساده خواهد بود. البته، هنگامی که این گزاره‌ها در داخل برنامه‌های واقعی دفن می‌شوند، همچنان یک مسئله قابل کنترل دشوار در انتخاب ورودی‌ها برای هدایت متغیرها در گزاره‌ها به مقادیر مطلوب وجود دارد اما این موضوع تمرکز این تحقیق نیست.

یک گزاره در متغیرهای n حداکثر دارای 2^n آزمون است. برای برنامه‌های کاربردی که در آن‌ها n بزرگ است، مجموعه آزمون جامع اغلب بسیار گران‌قیمت است. از این رو، برخی از معیارهای منطقی قابلیت تشخیص نقص را برای کاهش اندازه مجموعه آزمون فدا می‌کنند. در این مقاله امکان‌سنجی برای بهبود راه‌حل‌های این تبادل مورد تجزیه و تحلیل قرار می‌گیرد. ما بر روی سه نقص تمرکز خواهیم کرد: نقص درج لیترال¹⁴ (LIF)، نقص مرجع لیترال (LRF) و نقص از قلم‌افتادگی لیترال [8] اسامی این سه مورد هستند. یک LIF شامل درج یک لیترال یا یک لیترال منفی در داخل یک اصطلاح است. یک LRF مستلزم جایگزینی یک لیترال یا یک لیترال منفی از یک اصطلاح دیگر است. یک LOF مستلزم حذف یک لیترال است. LIF، LRF و LOF به دو دلیل مهم هستند. نخست، آن‌ها اشتباهات برنامه‌نویس را تقلید می‌کنند. فرضیه برنامه‌نویس شایسته [1] بیان می‌کند که برنامه‌نویس شایسته برنامه‌هایی را می‌نویسد که از نسخه فعلی با تعداد نسبتاً کمی از نقص‌های نمونه تفاوت دارد و بنابراین نقص‌های سلسله‌مراتبی خطاهای¹⁵ احتمالی هستند که باید مورد آزمون قرار بگیرند. دوم سلسله‌مراتب نقص لاو و یو [6] اطمینان می‌دهد که تشخیص این نقص‌ها سبب شناسایی سایر نقص‌ها خواهد شد. یعنی این سه خطا در بالای یک سلسله‌مراتبی از نقص‌ها قرار دارند.

چن¹⁶، لاو و یو [4] معیار پوشش MUMCUT را توسعه دادند که به طور خاص برای تضمین شناسایی تمامی نقص‌ها در سلسله‌مراتب نقص‌ها کاربرد دارد. معیار MUMCUT مشتمل بر سه معیار مولفه اصلی است که عبارتند از: چند نقطه درست منحصر به فرد (MUTP)، چند نقطه غلط نزدیک (MNFP) و نقطه درست

¹² Boolean

¹³ isolation

¹⁴ the Literal Insertion Fault

¹⁵ errors

¹⁶ Chen

منحصربه‌فرد متناظر با نقطه غلط نزدیک (CUTPNFP). جزئیات این معیارهای سازنده در بخش 2 ارائه شده است. برای این مقاله، مسئله کلیدی نقش امکان‌سنجی در هر معیار مولفه ضروری است.

به طور خاص اگر MUTP امکان‌پذیر باشد، می‌توان آن را با آزمون‌های کمتری نسبت به آنچه که CUTPNFP یا MNFP نیاز دارد، تقویت کرد و در عین حال کل سلسله مراتب نقص‌ها را شناسایی کرد. در صورتی که MUTP غیرممکن باشد، شرایط پیچیده خواهد شد. زمانی که MUTP غیرممکن است ولی CUTPNFP امکان‌پذیر است، اصلاً نیازی به MNFP نیست. اگر هم MUTP و هم CUTPNFP غیرممکن باشند، پس MNFP مورد نیاز است. یک جنبه کلیدی این مقاله این است که استدلال‌های غیرممکن در سطح ریزساختار اصطلاحات و لیترال‌ها اعمال می‌شود و بنابراین CUTPNFP و در صورت لزوم MNFP می‌تواند تنها در جایی که ضروری است مورد استفاده قرار بگیرد.

MUMCUT رویکرد ساده‌ای دارد که نیاز به تمام سه معیار مولفه اصلی دارد. این رویکرد قطعاً کارآیی لازم را دارد ولی گران است. CUTPNFP و به ویژه MNFP نیاز به تعداد زیادی از آزمون‌ها دارد اما همانطور که در بالا اشاره شد، تنها در موارد نسبتاً کمی ضروری است.

اعانه‌های این مقاله عبارتند از:

1) استفاده از تجزیه و تحلیل امکان‌سنجی معیار مولفه اصلی MUMCUT در سطح لیترال‌ها و اصطلاحات، این تجزیه و تحلیل سبب می‌شود که اندازه مجموعه آزمون بدون قربانی کردن تشخیص نقص کاهش پیدا می‌کند.

2) اصلاحی بر روابط تشخیص نقص سلسله‌مراتبی کار لاو و یو [9] بر اساس امکان‌سنجی معیار مولفه اصلی از معیار MUMCUT را انجام می‌دهد (شکل 1. 3).

3) یک معیار پوشش منطق جدیدی را ارائه می‌دهد که نام آن کمینه MUMCUT است و همچنین الگوریتمی برای تولید مجموعه آزمون‌های کمینه MUMCUT را ارائه می‌کند.

4) یک مطالعه موردی را نشان می‌دهد که در آن کاهش اندازه آزمون با کمینه MUMCUT امکان‌پذیر است.

این مقاله بدین شرح مرتب شده است که باقی بخش 1 به بررسی اصطلاحات منطق بولین و کارهای مرتبط در معیار منطق می‌پردازد. در بخش دوم، MUMCUT و سه معیار مولفه اصلی MNFP و CUTPNFP و MUTP مورد بررسی قرار خواهند گرفت. در بخش 3 سلسله‌مراتب نقص‌ها مورد بررسی خواهند گرفت. در بخش 4 نتایجی را توسعه خواهیم داد که تاثیر غیرممکن بودن را در تشخیص نقص ارائه می‌دهند و الگوریتمی را برای تعیین امکان‌سنجی هر کدام از معیارها در سطح اصطلاحات و لیترال‌ها نمایش خواهد داد و در نهایت این نتایج را به الگوریتم کمینه MUMCUT می‌افزاید. در بخش 5 یک مطالعه موردی به منظور ارزیابی کاهش اندازه مجموعه آزمون توسط کمینه MUMCUT ارائه شده است. در بخش 6 در مورد این موضوع بحث می‌کنیم که چگونه این کار مربوط به مسائل کلی‌تری از آزمون است و در بخش 7 به جمع‌بندی مقاله خواهیم پرداخت.

1.1 مجموعه اصطلاحات منطق بولین

جدول 1.1 تعاریف مربوط به اصطلاحات مورد استفاده در این مقاله را فهرست می‌کند.

جدول 1.1 تعاریف پایه

اصطلاح یا نماد	تعریف
1	مقدار بولین برای درست (TRUE)
0	مقدار بولین برای غلط (FALSE)
لیترال‌ها	متغیرهایی که نشان‌دهنده بندها در یک گزاره هستند
+	عملگر OR
مجاورت بین لیترال‌ها	عملگر AND
اصطلاح	مجموعه‌ای از لیترال‌ها که توسط AND متصل شده‌اند
~	نقیض
فرم نرمال اشتراکی (DNF)	نحوه ¹⁷ گزاره‌ای که در آن اصطلاحات توسط OR جدا شده‌اند و لیترال‌ها توسط AND جدا شده‌اند.
مفسر ¹⁸	یک اصطلاحی که زمانی که TRUE است بدین معنی است که گزاره درست است.
مفسران اولیه	مفسرانی که زمانی که یک لیترال را حذف می‌کنند احتمالاً قادر به تغییر ارزش گزاره هستند.
DNF غیرزائد ¹⁹	نحوه گزاره‌ای که در آن امکان این وجود دارد که هر اصطلاح را به ترتیب TRUE کنیم

¹⁷ syntax

¹⁸ Implicant

	درحالیکه سایر اطلاعات FALSE هستند.
کمینه DNF	نحو گزاره‌ای در DNF غیرزائد که در آن تمامی مفسران، مفسران اولیه هستند.
نقطه درست منحصر به فرد (UTP)	تخصیص مقادیری که در آن‌ها تنها یک اصطلاح TRUE است. در $ab + cd$ ، UTP‌ها برای ab برابر است با 1100 ، 1101 ، 1110 .
نقطه غلط نزدیک (NFP)	تخصیص مقادیری که در آن گزاره FALSE است اما نقیض یک لیترال تنها سبب می‌شود که گزاره TRUE شود [3]. در $ab + cd$ ، NFP‌ها برای a برابر هستند با 0100 ، 01010 ، 0110 .
NFP متناظر	یک NFP که از یک UTP برای لیترال اصطلاح تنها در مقدار لیترال تفاوت دارد. در $ab + cd$ ، NFP 0100 متناظر برای a است زیرا از 1100 UTP برای ab تنها در مقدار a تفاوت دارد.
امکان پذیر	یک معیار منطقی امکان‌پذیر است اگر و تنها قادر به ساخت تمامی آزمون‌های مورد نیاز باشد.

1.2 کار مرتبط

معیار منطقی از نظر نحوی (با فرض اینکه یک گزاره در یک قالب خاص است) و معنایی (بدون فرض هیچ‌گونه قالبی) مورد بررسی قرار گرفته است. چایلنسکی^{۲۰} و میلر^{۲۱} [6] در مورد معیار پوشش تصمیم/ شرایط اصلاح‌شده (MC/DC) بحث کرده‌اند که بهترین معیار منطقی معنایی شناخته شده است. با این حال آزمون‌های MC/DC تشخیص اکثر نقص‌ها در سلسله‌مراتب نقص‌های لاو و یو را تضمین نمی‌کند [7]. ویووکر^{۲۲}، گورادیا^{۲۳} و سینق^{۲۴} [13] معیارهای نحوی MAX-A و MAX-B را پیشنهاد دادند، که آزمون‌های آن‌ها سبب تضمین تشخیص تمامی نقص‌ها در سلسله‌مراتب شده است. چن، لاو و یو [4] یک معیار MUMCUT را توسعه دادند که در آن آزمون‌ها سبب تضمین تشخیص تمامی نقص‌ها در سلسله‌مراتبی با اندازه مجموعه آزمون کوچک‌تری شد. چن و لاو [2] الگوریتم حریمانه MUTP^{۲۵} را برای برآوردسازی معیار MUTP به عنوان مولفه اصلی معیار MUMCUT پیاده‌سازی کردند. کامینسکی^{۲۶}، ویلیامز^{۲۷} و اممان^{۲۸} [7] معیار MUTP/NFP را ارائه کردند که در آن آزمون‌ها

¹⁹ Irredundant

²⁰ Chilenski

²¹ Miller

²² Weyuker

²³ Goradia

²⁴ Singh

²⁵ Greedy

²⁶ Kaminski

²⁷ Williams

²⁸ Ammann

سبب تضمین تشخیص تمامی نقص‌ها در سلسله مراتب می‌شود و علاوه بر این سبب کاهش اندازه مجموعه آزمون نیز می‌شد ولی تنها در صورتی که معیار امکان‌پذیر بود. سان²⁹ و همکاران [12] به تجزیه و تحلیل این پرداختند که چگونه MUMCUT را می‌توان گسترش داد تا بر روی گزاره‌هایی در هر قالبی قابل اعمال باشد و اکون³⁰، بلک³¹ و یشا³² [11] نشان دادند که چگونه یک سلسله‌مراتب نقص منطق را می‌توان در هر گزاره‌ای در هر قالبی اعمال کرد. کار اصلی در تدوین یک سلسله مراتب نقص منطقی است که توسط کوهن³³ [8] انجام شده است. لائو و یو کار کوهن را با معرفی نقص‌های جدید و روابط تشخیصی جدید اصلاح کردند.

در این مقاله به بهبود تحقیقات پیشین با تمرکز بر امکان‌سنجی معیار برای اصطلاحات و لیترال‌های تکی پرداختیم. نتیجه معیار جدیدی است که سبب کاهش اندازه مجموعه آزمون بدون قربانی کردن تشخیص نقص شده است، حتی اگر غیرممکنی برای کل گزاره به وقوع بپیوندد.

2. معیار منطق

اندازه آزمون منطقی جامع، به شکل نمایی³⁴ رشد می‌کند، آزمون‌های مورید نیاز $O(2^n)$ است که در آن n تعداد لیترال‌های منحصر به فرد است. آزمایش‌کنندگان معیارهای ارزان‌تری را اختراع کردند. چهار مورد از چنین معیارهایی در مثالی به شکل $ab+cd$ توصیف شده است. خلاصه‌ای از هر کدام به همراه یک معیار منطق جدید در بخش 4 قسمت ضمیمه A ارائه شده است. توجه داشته باشید که برای تمامی این معیارها، اگر غیرممکن بودن به وقوع بپیوندد، آزمون‌های انتخاب شده باید نیازمندی‌ها را به طور کامل برآورده کنند.

²⁹ Sun

³⁰ Okun

³¹ Black

³² Yesha

³³ Kuhn

³⁴ exponentially

MUTP 2.1

چند نقطه درست منحصر به فرد (MUTP): یک گزاره DNF کمینه از آزمون‌هایی برای یک UTP برای هر اصطلاح داده شده است که در آن تمامی لیترال‌ها در اصطلاح مقادیر 1 و 0 را بدست نیاوردند. یک UTP برای اصطلاح اول باید دارای مقادیر $a=1$ و $b=1$ باشد. آزمون‌های لازم برای c و d برای هر کدام باید به شکل $c=0$ و $d=1$ باشد که به شکل 1101 و 1110 هستند. یک UTP برای اصطلاح دوم باید به شکل $c=1$ و $d=1$ باشد. آزمون‌های لازم برای a و b هر کدام به شکل $a=0$ و $b=1$ است و به شکل 0111 و 1011 هستند. یک مجموعه آزمون به شکل $\{1101, 1110, 0111, 1011\}$ است.

MNFP 2.2

چند نقطه غلط نزدیک (MNFP): یک گزاره DNF کمینه از آزمون‌هایی برای یک NPF از هر لیترال داده شده است که در آن تمامی لیترال‌ها از نظر اصطلاح لیترال مقادیر 0 و 1 را کسب نکرده است. یک UTP برای اصطلاح اول باید به شکل $a=1$ و $b=1$ باشد. NFPها برای a و b و بنابراین برای c و d هر کدام $=0$ و 1 است که به شکل 0110، 0101، 1001 و 1010 است. یک UTP برای اصطلاح دوم باید دارای $c=1$ و $d=1$ باشد. NFPهای لازم برای c و d و بنابراین برای a و b هر کدام $=0$ و 1 است که به شکل 010، 1001، 0110، 1010 باشد. یک مجموعه آزمون به شکل

$\{0101, 0110, 1001, 1010\}$ خواهد بود.

CUTPNFP 2.3

نقطه درست منحصر به فرد متناظر با نقطه غلط نزدیک (CUTPNFP): یک گزاره DNF کمینه از آزمون‌هایی برای هر لیترال یافت شده در یک UTP و NFP داده شده است که در آن تنها لیترال تغییر مقدار می‌دهد. یک UTP برای اصطلاح اول باید به شکل $a=1$ و $b=1$ باشد. اگر $c=0$ و $d=1$ باشد، آزمون‌ها برای ab به

شکل 1101، 0101 و 1001 است. یک UTP برای اصطلاح دوم باید دارای مقادیر $c=1$ و $d=1$ باشد. اگر $a=1$ و $b=0$ باشد، آزمون‌ها برای cd به شکل 1011، 1001 و 1010 خواهد بود. یک مجموعه آزمون به شکل $\{1101, 0101, 1001, 1011, 1010\}$ خواهد بود.

MUMCUT 2.4

این مورد معیارهای $MUTP$ ، $MNFP$ ، $CUTPNFP$ را $(MUMCUT)$ $MUTP$ / $MNFP$ / $CUTPNFP$ برآورده می‌کند. 1101 و 1110 UTP‌هایی برای ab هستند. 0101 و 0110 NFP‌هایی برای a هستند که متفاوت از یک UTP برای ab تنها در مقدار a هستند. 1001 و 1010 NFP‌هایی برای b هستند که در یک UTP برای فقط ab در مقدار b متفاوت هستند. 0111 و 1011 UTP‌هایی برای cd هستند. 0101 و 1001 NFP‌هایی برای c هستند که از یک UTP از cd تنها در مقدار c تفاوت دارند. 0110 و 1010 NFP‌هایی برای d هستند که از یک UTP برای cd تنها در مقدار d تفاوت دارند. در NFP‌های بالا تنها هر لیترال از نظر جذب مقدار بهره 1 و 0 مطرح نیستند. یک مجموعه آزمون به شکل زیر خواهد بود:

$\{1101, 1110, 0101, 0110, 1001, 1010, 0111, 1011\}$

3. آزمون‌های منطقی و سلسله مراتب نقص

یک روش برای ارزیابی آزمون‌ها تعیین تعداد نه نقص DNF کمینه در جدول 1. 3 است که یک مجموعه آزمون با تضمین تشخیص است [8,9].

جدول 3.1. 1. نقص‌های منطقی DNF کمینه رایج

نقص	توضیح
نقص نقیص عبارت (ENF)	نقیص گزاره $ab+c$ به شکل $\sim(ab+c)$ پیاده‌سازی شده است.
نقص نقیص اصطلاح (TNF)	یک اصطلاح نقیص شده: $ab+c$ به شکل $\sim(ab)+c$ پیاده‌سازی شده است.
نقص مرجع عملگر $+$ ^{۳۵} (ORF+)	جایگزینی OR با AND: $a+b$ به شکل ab پیاده‌سازی شده است.
نقص مرجع عملگر $.$ (ORF.)	جایگزینی AND با OR: ab به شکل $a + b$ پیاده‌سازی شده است.
نقص مرجع لیترال (LRF)	یک لیترال که توسط یک لیترال یا نقیص یک لیترال که در اصطلاح نیست جایگزین شده است: $ab+cd$ به شکل $cb+cd$ یا به شکل $\sim cb+cd$ پیاده‌سازی شده است.
نقص ازقلم‌افتادگی اصطلاح (TOF)	یک اصطلاحی که به شکل $ab+cd$ است به شکل ab پیاده‌سازی شده است.
نقص ازقلم‌افتادگی لیترال (LOF)	یک لیترال از قلم افتاده است یعنی ab به شکل a پیاده‌سازی شده است.
نقص درج لیترال (LIF)	یک لیترال به حالت خود یا به حالت نقیص خود درج نشده است: $ab+cd$ به شکل $abc+cd$ یا به شکل $ab\sim c + cd$ پیاده‌سازی شده است.

شرایط برای شناسایی LIF در ادامه شرح داده شده است [4]. اگر لیترالی که قرار نبوده در اصطلاح X درج شده باشد، در اصطلاح X خودش یا نقیضش درج شده باشد، پس یک مجموعه از UTP‌ها برای X که در آن تمامی لیترال‌هایی که در X نیستند مقادیر 0 و 1 را کسب می‌کنند خطا را تشخیص می‌دهند. آزمون‌های MUTP سبب تضمین تشخیص در یک LIF خواهند شد [4]. با این حال، زمانی که معیار MUTP غیرممکن باشدف یک LRF وجود دارد که آزمون‌های MUTP ممکن است آن را شناسایی نکنند (به قضیه 4.1 نگاه کنید). $ab + ac + bc$ را به عنوان یک LIF در نظر بگیرید که $ab\sim c + ac + bc$ را تولید می‌کند. معیار MUTP برای ab غیرممکن است زیرا تنها UTP برای ab برابر با 110 است. بنابراین، آزمون‌های MUTP مقادیر متناظر LRF‌ها را تشخیص نمی‌دهد که عبارتند از: $\sim cb + ac + bc$ و $a\sim c + ac + bc$.

آزمون‌های MUTP برای شناسایی یک LRF برای یک لیترال تضمین شده است در صورتی که معیار MUTP برای لیترال اصطلاح امکان‌پذیر باشد [7]. در این مورد، لازم است که معیار MUTP و معیار NFP برآورده شود (یک NFP برای هر لیترال در اصطلاح) تا به تضمین تشخیص یک LIF، LRF و LOF در آن اصطلاح است [7]. NFP برای یک لیترال در یک اصطلاح قابل‌اجرای MUTP با NFP‌هایی برای سایر لیترال‌ها در اصطلاحات

³⁵ Operator

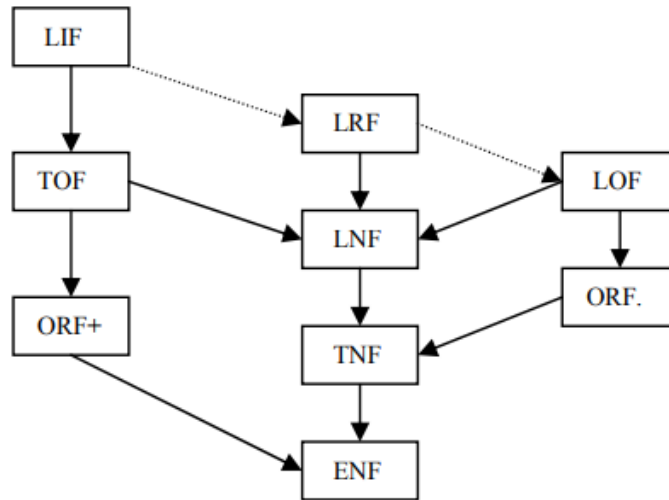
قابل اجرای MUTP و با NFP‌هایی در آزمون‌هایی در CUTPNFP یا MNFP است زیرا هر NFP برای تشخیص لیترال یک LOF برای لیترال است [4]. اگر یک اصطلاح MUTP غیرممکن باشد ولی تمامی لیتراهای خارجی که نمی‌توانند در یک UTP برای اصطلاح موجود در اصطلاحات تک لیترالی 0 یا 1 باشند، تشخیص LRF همچنان توسط آزمون‌های MUTP تضمین شده است. یک LRF شامل جایگزینی یک لیترال با یک لیترال (یا نقیض آن) است که در یک اصطلاح تک لیترالی وجود دارد و نتیجه این امر یک گزاره LOF، TOF یا یک گزاره TRUE خواهد بود. از آنجایی که UTP تضمین‌کننده شناسایی یک TOF [4] است و یک NFP تضمین‌کننده تشخیص یک LOF یا یک نقص را می‌کند که در آن گزاره‌ها در 1 گیر کرده‌اند، مجموعه آزمون MUTP با همپوشانی NFP‌ها تکمیل شده است که تضمین تشخیص LRF را در پی دارد. برای مثال، در $a+b$ ، با جایگزینی a با b به یک TOF برای a می‌رسیم و با جایگزینی a با $\sim b$ این گزاره برابر با 1 خواهد شد. در $ab + c$ با جایگزینی a با c به یک TOF برای ab می‌رسیم و با جایگزینی a با $\sim c$ به یک LOF برای a خواهیم رسید.

شرایط برای شناسایی LRF مطابق با [4] است که در ادامه توصیف شده است. اگر لیترال x در X به طور غیرمستقیم به عنوان برخی از سایر لیترال‌ها یا به عنوان نقیض برخی از سایر لیترال‌هایی که در X نیستند پیاده‌سازی شده باشد، سپس هر یک از مواردی که در ادامه ذکر شده است یک خطا را تشخیص می‌دهد: مجموعه‌ای از UTP‌ها برای X که در آن لیترال‌هایی که در X نیستند مقادیر 0 و 1 را کسب می‌کنند. یک مجموعه از NFP‌ها برای x که در آن تمامی لیترال‌هایی که در X نیستند مقادیر 0 و 1 را کسب می‌کنند. یک جفت UTP-NFP که در آن نقاط فقط در مقدار x متفاوت هستند. معیار CUTPNFP به منظور تولید آزمون‌هایی برای شناسایی یک LRF طراحی شده سات ولی زمانی که این امر غیرممکن است موفق به تولید آزمون‌ها نخواهد شد (قضیه 2.4 را ببینید). با این حال، هنگامی که معیار CUTPNFP غیرممکن باشد، آزمون‌های MNFP را می‌توان برای تضمین تشخیص LRF اضافه کرد [4]. $abc + abd + \sim b \sim d + \sim de$ را در نظر بگیرید. معیار CUTPNFP برای b در abc غیرممکن است. تنها UTP برای abc ، 11100 است. NFP متناظر 10100 برای b در abc امکان‌پذیر نیست زیرا یک نقطه TRUE است. حال $a \sim ec + abd + \sim b \sim d + \sim de$ LRF را در نظر بگیرید. از آنجایی که معیار

CUTPNFP برای b در abc غیرممکن است، LRF با آزمون‌های CUTPNFP غیرقابل شناسایی است. یک NFP تکی برای b در abc برای شناسایی یک LRF نیز تضمین نشده است. معیار MNFP نیازمند آن است که NFP 10110 برای b در abc مورد استفاده قرار بگیرد و LRF را شناسایی کند.

شکل 3.1 سلسله مراتب نقص لاو و یو [9] که بر اساس چگونگی تاثیر امکان‌سنجی معیار بر روی تشخیص نقص اصلاح شده است. یک فلش توپر از یک نقص منبع به یک نقص مقصد نشان می‌دهد که اگر یک آزمون یک منبع نقص را شناسایی کند، همچنین نقص مقصد مربوطه را نیز تشخیص می‌دهد. هنگامی که معیار MUTP غیرممکن است، یک مجموعه آزمونی که تمامی LIFها را شناسایی کرده است، تضمین نمی‌کند که تمامی LRFها را نیز شناسایی کرده باشد. بنابراین فلش توپر بین LIF و LRF در سلسله مراتب لاو و یو به فلش‌هایی نقطه‌چین تغییر یافته است. در سلسله مراتب لاو و یو هیچ فلشی بین LRF و LOF وجود ندارد. یک فلش نقطه‌چین به منظور نمایش زمانی که تشخیص تمامی LIFها تضمین شده است ولی تشخیص تمامی LRFها تضمین نشده است، اضافه شده است، افزایش آزمون‌هایی برای تشخیص LRFهای شناسایی نشده سبب تشخیص تمامی LOFهای متناظر خواهد شد (مگر اینکه معیار CUTPNFP غیرممکن باشد). دلیل آن این است که وقتی معیار MUTP غیرممکن باشد اما معیار CUTPNFP امکان‌پذیر باشد، UTP یک LRF را شناسایی نخواهد کرد اما NFP مربوطه آن را شناسایی خواهد کرد [4].

شکل 3.1 سلسله مراتب کلاس نقص ها



4. استفاده از امکان‌سنجی معیار برای ارزیابی تشخیص نقص

یک نمونه از روند شناسایی هر LRF از هر لیترال در یک اصطلاح امکان‌پذیر MUTP توسط مجموعه آزمون MUTP در جدول 4.1 و 4.2 ارائه شده است. جدول 2.4 نشان می‌دهد که چگونه آزمون‌های MUTP هر LRF را با استفاده از آزمون‌ها در جدول 4.1 شناسایی می‌کند.

جدول 4.1 آزمون‌های MUTP برای $ab+ \sim ac$

نام آزمون	مقادیر	مورد آزمون
ab برای MUTP	110	1
ab برای MUTP	111	2
$\sim ac$ برای MUTP	001	3
$\sim ac$ برای MUTP	011	4

جدول 4.2 تشخیص نقص برای $ab + \sim ac$

کلاس نقص	گزاره خطا	مورد آزمون تشخیص خطا	ارزش گزاره اصلی	ارزش گزاره غلط
LRF	$cb + \sim ac$	1	True	False
LRF	$\sim cb + \sim ac$	2	True	False
LRF	$ac + \sim ac$	1	True	False
LRF	$a\sim c + \sim ac$	2	True	False
LRF	$ab + bc$	3	True	False
LRF	$ab + \sim bc$	4	True	False
LRF	$ab + \sim ab$	3	True	False
LRF	$ab + \sim a\sim b$	4	True	False

این بخش، با مدارک مرتبط با امکان‌پذیری معیار برای قابلیت تشخیص نقص ادامه پیدا می‌کند.

قضیه 4.1:

اگر معیار MUTP برای یک اصطلاح چند لیترالی X غیر ممکن باشد، یک LRF که در آن نقیض تعدادی لیترال Y در تعدادی از اصطلاح چند لیترالی X به جایگزین لیترال x در X می‌پردازد که توسط آزمون‌های MUTP شناسایی نمی‌شود.

اثبات:

بدون از دست دادن همبستگی بحث فرض کنید که Y باید برای یک UTP برای X برابر با 0 باشد. نقصی را در نظر بگیرید که در آن $\sim Y$ در داخل X درج شده باشد. LRF متناظر X را با $\sim Y$ جایگزین می‌کند. در یک UTP برای X تمامی لیترال‌ها به گونه‌ای هستند که فقط $X=1$ است. جایگزینی $\sim Y$ برای X همچنین سبب می‌شود که $X=1$ باشد زیرا Y باید در UTP از X برابر با 1 باشد. از آنجایی که X و Y اصطلاحات چندلیترالی هستند، LRF به TOF یا LOF منجر نخواهد شد. معیار MUTP فقط نیاز به نقاط TRUE برای X دارد تا آزمون‌های MUTP موفق به شناسایی یک LRF نشوند.

پایان اثبات

یک مثال در این مورد تغییر $ab + bc$ به $a\sim c + bc$ است. معیار MUTP برای ab غیرممکن است زیرا 110 تنها UTP برای ab است. 110 سبب تمایز بین ab و $a\sim c$ نمی‌شود زیرا آزمون‌های MUTP موفق به شناسایی LRF نخواهند شد.

الگوریتمی برای تعیین امکان‌پذیری MUTP برای هر اصطلاح در یک گزاره در ادامه ارائه شده است.

الگوریتم امکان‌سنجی MUTP

ورودی: تمامی UTPها برای هر اصطلاح

خروجی: فهرستی از اصطلاحات امکان‌پذیر MUTP

Declare list MUTPFeasibleTerms = All terms

For each term i

 For each literal x not in term i

 Boolean doesLiteralEqual_0 = false

 Boolean doesLiteralEqual_1 = false

 For each UTP u in the set of UTPs for term i

 If (literal x = 0 in UTP u) doesLiteralEqual_0 = true

 Else doesLiteralEqual_1 = true

 If (doesLiteralEqual_0 && doesLiteralEqual_1)

 Continue for each literal loop

 End For

 Remove term i from MUTPFeasibleTerms

 Continue for each term loop

 End For

End For

Return MUTPFeasibleTerms

قضیه 2.4:

اگر معیار CUTPNFP برای لیترال x در یک اصطلاح چندلیترالی X امکان پذیر باشد، یک LRF که در آن نقیض تعدادی لیترال y در تعدادی اصطلاح چندلیترالی Y جایگزین x شده است را نمی توان توسط آزمون های CUTPNFP مورد شناسایی قرار داد.

اثبات

چن، لاو و یو [4] نشان دادند که به طور کلی هنگامی که معیار MUTP غیر ممکن است، NFP برای شناسایی LRF مورد نیاز است. (آن ها تفاوتی بین LRF هایی که شامل اصطلاحات تک لیترالی و چند لیترالی بودند را مانند قضیه 1.4 قائل نبودند). هنگامی که معیار CUTPNFP برای x غیر ممکن است، یک LRF را نمی توان با یک NFP متفاوت شناسایی کرد زیرا هیچ کدام وجود ندارند. بنابراین اگر توسط یک UTP برای X قابل تشخیص نباشد، آزمون های CUTPNFP آن را شناسایی نمی کنند. آزمون های MUTP تضمین تشخیص LRF را در زمانی که معیار MUTP غیر ممکن است را نمی دهند و بنابراین چیزی که برای اثبات باقی می ماند اثبات این است که زمانی که معیار CUTPNFP برای x غیر ممکن است، معیار MUTP برای x غیر ممکن است. از آنجایی که هیچ NFP متناظری برای x وجود ندارد، تغییر مقدار x در یک UTP برای x سبب می شود که سایر اصطلاحات Y تبدیل به 1 شوند. در این مورد، معیار MUTP برای X غیر ممکن است زیرا با اثبات ارائه شده در تضاد است. فرض کنید که معیار MUTP برای X امکان پذیر باشد و معیار CUTPNFP برای x غیر ممکن باشد. هر لیترالی که در X است می تواند در یک UTP برای X برابر با 0 یا 1 باشد. بنابراین هر اصطلاحی به جز X می تواند برای یک UTP از X برابر با صفر باشد و مهم نیست که y برابر با 0 یا 1 باشد. بنابراین، تغییر مقدار x در یک UTP برای x می تواند یک NFP متناظر را ایجاد کند که متناقض با فرض اصلی است. از آنجایی که X و Y اصطلاحاتی چندلیترالی هستند، یک LRF نمی تواند یک TOF یا LOF را تولید کند.

پایان اثبات

به عنوان یک مثال، $abc + abd + \sim b \sim d + \sim de$ را در نظر بگیرید. معیار CUTPNFP برای b در abc غیرممکن است. تنها UTP برای abc برابر با 11100 است. NFP متناظر 10100 برای b در abc امکان پذیر نیست و این یک نقطه TRUE است. حال LRF $a \sim ec + abd + \sim b \sim d + \sim de$ را در نظر بگیرید. از آنجایی که معیار CUTPNFP برای b در abc غیرممکن است، آزمون‌های CUTPNFP، قادر به شناسایی LRF نیست. معیار MNFP در صورتی که NFP 10110 برای b در abc مورد استفاده قرار بگیرد، الزامی خواهد بود ز را LRF را شناسایی خواهد کرد.

الگوریتمی برای تعیین امکان‌سنجی معیار CUTPNFP برای هر لیترال در یک گزاره در بخش بعدی ارائه شده است.

الگوریتم امکان‌پذیری CUTPNFP

ورودی: تمامی UTP ها برای هر اصطلاح و تمامی NFP ها برای هر لیترال.

خروجی: فهرستی از تمامی لیترال‌های امکان‌پذیر CUTPNFP.

Declare List CUTPNFPFeasibleLiterals

Declare String nfpComplement

Declare Boolean isCUTPNFPFeasible

For each term i

 For each literal j in term i

 isCUTPNFPFeasible = false

 For each nfp k for literal j in term i

 nfpComplement = k where j th bit is complemented

 If the set of utps for term i contains nfpComplement

 isCUTPNFPFeasible = true

Break

End For

If (isCUTPNFPFeasible)

Add literal j to CUTPNFPFeasibleLiterals

End For

End For

Return CUTPNFPFeasibleLiterals

یک معیار منطق جدید، که ما آن را کمینه MUMCUT می‌نامیم، با مجموعه آزمونی که توسط الگوریتم زیر تولید می‌شود، می‌تواند سبب کاهش اندازه مجموعه آزمون MUMCUT بدون قربانی کردن تشخیص نقص شود. این الگوریتم بر اساس امکان‌پذیری معیار برای 1) همپوشانی‌های NFPها در صورت امکان و 2) تولید آزمون‌های MNFP و CUTPNFP تنها در صورت لزوم بر مبنای لیترال به لیترال خواهد شد.

الگوریتم کمینه MUMCUT

For each term X

Generate MUTP tests for X

If the MUTP criterion is infeasible* for X

For each literal x in X

If the CUTPNFP criterion is feasible for x

Generate CUTPNFP tests for x to overlap NFPs**

Else Generate MNFP tests for x to overlap NFPs**

End For

Else

Generate a NFP for x to overlap NFPs**

End For

* معیار MUTP در این الگوریتم در صورتی غیرممکن است که اگر و تنها اگر X یک اصطلاح چندلیترالی باشد و یک اصطلاح y در یک اصطلاح چندلیترالی Y وجود دارد که در آن y نمی‌تواند هر دو مقادیر حقیقی را برای یک UTP برای X را بدست آورد.

** همپوشانی‌های NFPها یک مجموعه پوشش‌دهنده از مسئله بهینه‌سازی ترکیبی است که معروف به NP کامل^{۳۶} است. یک سلسله مراتب در این الگوریتم برای تقریب کمینه تعداد NFPهای تولید شده مورد استفاده قرار گرفته است.

به عنوان مثال، $ab + cd$ را در نظر بگیرید. 1101 و 1110 UTPهایی برای ab هستند و معیار MUTP برای ab امکان‌پذیر است. 0101 و 1010 NFPهایی برای a و b هستند. 0111 و 1011 UTPهایی برای cd هستند و معیار MUTP برای CD امکان‌پذیر است. 0101 و 1010 NFPهایی برای c و d هستند. یک مجموعه آزمون به شکل $\{1101, 1110, 0101, 1010, 0111, 1011\}$ است که دارای دو آزمون کمتر نسبت به مجموعه آزمون MUMCUT در بخش 2.4 است.

5. ارزیابی تجربی

چن، لاو و یو [4] اندازه مجموعه آزمون MUMCUT را (با استفاده از الگوریتم MUTP حریصانه [2]) برای 19 گزاره DNF کمینه از یک سیستم جلوگیری از تصادم ترافیک هوایی (TCAS) مورد ارزیابی قرار دادند. در واقع 20 گزاره وجود دارد که 12 مورد از آنها به دلیل فقدان پرانتزهای راست حذف شده است [13]. گزاره‌ها دارای 5 تا 13 لیترال منحصر به فرد هستند (ضمیمه B را ببینید). در این بررسی، کمینه MUMCUT برای هر گزاره ایجاد شده است و امکان‌سنجی MUTP برای هر اصطلاح و امکان‌سنجی CUTPNFP برای هر لیترال مورد ارزیابی قرار گرفته است. الگوریتم کمینه MUMCUT در زبان جاوا^{۳۷} و برای کسب نتایج پیاده‌سازی شده است.

³⁶ NP-complete.

³⁷ Java

نتایج نشان می‌دهند که معیار CUTPNFP برای تمامی 853 لیترال در دسترس امکان‌پذیر بوده است، بنابراین آزمون‌های MNFP برای هر لیترالی ضروری نبوده است. برای 204 لیترال (23.92٪) معیار MUTP برای لیترال اصطلاح امکان‌پذیر بوده است و بنابراین آزمون‌های MUTP یک LRF را شناسایی خواهند کرد. برای 649 لیترال دیگر (80.76٪) آزمون‌های CUTPNFP یک LRF را شناسایی کرده‌اند. برای چهار گزاره، معیار MUTP برای هر اصطلاح امکان‌پذیر بوده است بنابراین آزمون‌های CUTPNFP مورد نیاز نبوده است. برای 16 گزاره، معیار MUTP برای حداقل یک اصطلاح امکان‌پذیر بوده است. بنابراین، آزمون‌های CUTPNFP برای لیترال‌ها در حداقل یک اصطلاح در اکثر گزاره‌ها ضروری نبوده است. جدول 5.1 این نتایج را نشان می‌دهد. اندازه مجموعه آزمون‌های کمینه MUMCUT و MUMCUT نیز مورد مقایسه قرار گرفته است. به طور میانگین اندازه مجموعه آزمون کمینه MUMCUT 66.12٪ از اندازه مجموعه آزمون MUMCUT بوده است و 50.2٪ از اندازه مجموعه آزمون کامل بوده است. بزرگترین پس‌انداز برای گزاره 13 بوده است که در آن اندازه مجموعه آزمون کمینه MUMCUT 30.1٪ از اندازه مجموعه آزمون MUMCUT و 54.0٪ از اندازه مجموعه آزمون جامع بوده است. جدول 5.2 این نتایج را نشان می‌دهد. اندازه مجموعه آزمون کمینه MUMCUT نیز همواره کمتر از اندازه مجموعه آزمون MUMCUT بوده است به جز زمانی که هر لیترال در هر اصطلاح باشد که در این حالت اندازه آزمون یکسان است (گزاره‌های 8 و 9 را ببینید). در این مورد، هر اصطلاح تنها دارای یک UTP است و هر لیترال دارای تنها یک NFP است (که متناظر NFP است) و هیچ LIF یا LRF ای وجود ندارد.

جدول 5.1 امکان‌سنجی معیار و تشخیص LRF

تعداد لیرهایی که برای شناسایی LRF نیاز به CUTPNFR دارند	تعداد لیترال‌هایی که MUTP LRF را شناسایی کرده است	تعداد اصطلاحاتی که MUTP غیرممکن هستند	تعداد اصطلاحاتی که MUTP امکان‌پذیر هستند	
24	5	4	1	1
72	33	9	4	2
136	10	23	2	3
6	1	2	1	4
27	1	8	1	5
36	22	4	2	6

7	4	4	28	32
8	4	0	32	0
9	2	0	14	0
10	0	6	0	60
11	1	8	6	57
12*	N/A	N/A	N/A	N/A
13	0	6	0	14
14	0	6	0	16
15	1	10	2	30
16	1	22	2	85
17	2	4	8	24
18	2	6	8	30
19	4	0	20	0
20	2	0	12	0
مجموع	34	122	204	649

* شماره 12 به دلیل فقدان پرانته‌های راست حذف شده است.

جدول 2. 5 اندازه مجموعه آزمون کمینه MUMCUT و MUMCUT

	کمینه MUMCUT	MUMCUT [4]	درصد	2"
1	27	40.50	66.67%	128
2	81	116.00	69.83%	512
3	157	1026.57	15.29%	4096
4	9	14.40	62.50%	32
5	36	232.48	15.49%	512
6	66	89.96	73.37%	2048
7	66	119.80	55.09%	1024
8	36	36.00	100.00%	256
9	16	16.00	100.00%	128
10	62	142.83	43.41%	8192
11	72	888.26	8.11%	8192
12	N/A	N/A	N/A	N/A
13	22	1687.00	1.30%	4096
14	22	73.75	29.83%	128
15	39	187.39	20.81%	512
16	107	1595.32	6.71%	4096
17	40	852.78	4.69%	2048
18	48	182.54	26.30%	1024
19	16	65.64	24.38%	256
20	14	24.00	58.33%	128
مجموع	936	7391.22		37,408
میانگین	49.26	389.01	12.66%	1968.84

مجموعه‌های آزمون کمینه MUMCUT برای گزاره‌های 4 و 13 و 19 در ادامه ارائه شده است.

گزاره 4: $a \sim bd + a \sim cd + e$

مجموعه آزمون MUTP برابر است با:

اصطلاح $a \sim bd$ 10110

اصطلاح $a \sim cd$ 11010

اصطلاح e 00001

اصطلاح e 11111

معیار MUTP برای $a \sim bd$ غیرممکن است زیرا در یک UTP برای $a \sim bd$ مقدار C باید برابر با 1 باشد و e باید برابر با 0 باشد. با این حال، e یک اصطلاح تک لیترالی است بنابراین آزمون‌های CUTPNFP تنها برای تشخیص یک LRF در جایی که C جایگزین یک لیترال در $a \sim bd$ شده است مورد نیاز است. معیار MUTP برای $a \sim cd$ غیرممکن است زیرا در یک UTP برای $a \sim cd$ ، b باید برابر با 1 و e باید برابر با 0 باشد. با این حال، e یک اصطلاح تک لیترالی است بنابراین آزمون‌های CUTPNFP تنها برای شناسایی LRF در جایی که v توسط یک لیترال در $a \sim cd$ جایگزین شده است مورد نیاز خواهد بود. از آنجایی که معیار MUTP برای e امکان‌پذیر است و e در یک اصطلاح تک لیترالی قرار دارد، نه آزمون‌های CUTPNFP و نه آزمون‌های NFP برای e ضروری نیستند.

آزمون‌های اضافی که برای یک مجموعه آزمون CUTPNFP ضروری هستند عبارتند از:

اصطلاح $a \sim bd$ ، لیترال a 00110

اصطلاح $a \sim bd$ ، لیترال a و اصطلاح $a \sim cd$ لیترال c 11110

اصطلاح $a \sim bd$ ، لیترال d 10100

اصطلاح $a \sim cd$ ، لیترال a 01010

اصطلاح $a \sim cd$ ، لیترال d 11000

گزاره 13: $a + b + c + \sim\text{def}\sim\text{g}\sim\text{h} + \text{ij}\sim\text{l} + \text{ik}\sim\text{l}$

مجموعه آزمون MUTP برابر است با:

a اصطلاح 100000000000

a اصطلاح 100111111111

b اصطلاح 010000000000

b اصطلاح 010111111111

c اصطلاح 001000000000

c اصطلاح 001111111111

$\sim\text{def}\sim\text{g}\sim\text{h}$ اصطلاح 000011000000

$\sim\text{def}\sim\text{g}\sim\text{h}$ اصطلاح 000011001111

$\text{ij}\sim\text{l}$ اصطلاح 000000001100

$\text{ij}\sim\text{l}$ اصطلاح 000111111100

$\text{ik}\sim\text{l}$ اصطلاح 000000001010

$\text{ik}\sim\text{l}$ اصطلاح 000111111010

معیار MUTP برای تمامی شرایط غیر ممکن است. با این حال، آزمون CUTPNFP برای a، b و c مورد نیاز نیست، زیرا این اصطلاحات به صورت تک لیترالی هستند. به همین ترتیب، آزمون CUTPNFP برای تشخیص یک LRF که در آن A، B یا C جایگزین یک لیترال در اصطلاح دیگری است نیز مورد نیاز نیست. هیچ آزمون CUTPNFP برای $\sim\text{def}\sim\text{g}\sim\text{h}$ مورد نیاز نیست زیرا تمامی لیتراهای خارجی در اصطلاحات چند لیترالی می توانند هم مقادی 0 و 1 را در یک UTP برای $\sim\text{def}\sim\text{g}\sim\text{h}$ بدست آورند. بنابراین هر NFP برای هر لیترالی در $\sim\text{def}\sim\text{g}\sim\text{h}$ می تواند برای تشخیص یک LOF تولید شود. تنها لیترال خارجی که در یک اصطلاح چند لیترالی نمی تواند مقادیر 0 و 1 را در یک UTP برای $\text{ij}\sim\text{l}$ کسب کند، k است. بنابراین، آزمونهای CUTPNFP برای

تشخیص یک LRF ای مورد نیاز است که در آن $k \sim$ جایگزین یک لیترال در $1 \sim ij$ می‌شود. تنها لیترال خارجی که در یک اصطلاح چند لیترالی نمی‌تواند مقادیر 0 و 1 را در یک UTP برای $1 \sim ig$ بدست آورد، بنابراین، آزمون CUTPNFP برای شناسایی یک LRF مورد نیاز است که در آن $z \sim$ جایگزین یک لیترال در $1 \sim ik$ شده است.

آزمون‌های NFP:

d لیترال اصطلاح $def \sim g \sim h$ ، 000111000000

e لیترال اصطلاح $def \sim g \sim h$ ، 000001000000

f لیترال اصطلاح $def \sim g \sim h$ ، 000010000000

g لیترال اصطلاح $def \sim g \sim h$ ، 000011100000

h لیترال اصطلاح $def \sim g \sim h$ ، 000011010000

آزمون‌های اضافه‌ای که برای یک مجموعه داده CUTPNFP مورد نیاز است:

i لیترال $1 \sim ij$ اصطلاح 000000000100

k لیترال $1 \sim ij$ و اصطلاح $1 \sim ik$ لیترال z اصطلاح 000000001000

l لیترال $1 \sim ij$ اصطلاح 000000001101

i لیترال $1 \sim ik$ اصطلاح 000000000010

l لیترال $1 \sim ik$ اصطلاح 000000001011

گزاره 19: $acefg + ace \sim fh + bdefg + bde \sim fh$

مجموعه آزمون MUTP برابر است با:

اصطلاح acefg 10111110

اصطلاح acefg 11101111

10111001 اصطلاح ace~fh

11101011 اصطلاح ace~fh

01111110 اصطلاح bdefg

11011111 اصطلاح bdefg

01111001 اصطلاح bde~fh

11011011 اصطلاح bde~fh

معیار MUTP در همه شرایط امکان پذیر است بنابراین یک مجموعه آزمون از NFPهای همپوشانی تولید می شود:

مجموعه آزمون از NFPهای همپوشانی برابر است با :

00111111 اصطلاح acefg ، لیترال a و اصطلاح bdefg ، لیترال b

11001111 اصطلاح acefg. لیترال c و اصطلاح bdefg ، لیترال d

11110110 اصطلاح acefg. لیترال e و اصطلاح bdefg ، لیترال e

11111010 اصطلاح acefg ، لیترال f و اصطلاح ace~fh ، لیترال h اصطلاح bdefg. لیترال f و اصطلاح

bde~fh. لیترال h

11111101 اصطلاح acefg ، لیترال g و اصطلاح ace~fh ، لیترال f و اصطلاح bdefg. لیترال bdefg. لیترال

g و اصطلاح bde~fh. اصطلاح f

01101001 اصطلاح ace~fh ، لیترال a و اصطلاح bde~fh ، لیترال d

10011011 اصطلاح ace~fh ، لیترال c و اصطلاح bde~fh ، لیترال b

11110011 اصطلاح ace~fh ، لیترال e و اصطلاح bde~fh لیترال e

این همپوشانی NFPها در صورتی که هر دو آزمون CUTPNFP و MUTP مورد نیاز باشد امکان پذیر نیست. به

عنوان مثال، 00111111 به عنوان یک NFP برای a در acefg و برای b در bdefg استفاده می شود، اما معیار

CUTPNFP برای b در bdefg را برآورده نمی کند، زیرا از UTP bdefg فقط در مقدار b متفاوت نیست.

بنابراین اگرچه NFP 00111111 برای b در bdefg است، اما NFP متناظر آن نیست. کوچکترین اندازه مجموعه آزمون که معیارهای CUTPNFP و MUTP را برآورده می‌کند 26 است، که 10 برابر بیشتر از اندازه تست کمینه MUMCUT است.

6. زمینه

برای اینکه معیار منطقی نحوی در عمل مفید باشد، باید سه مسئله جداگانه را حل کرد: مسئله متغیر داخلی، مسئله حداقل DNF، و مسئله اندازه پیش فرض نام این سه مسئله هستند. مسئله متغیر داخلی مربوط به آنچه است که ورودی‌ها هست که یک متغیر به یک مقدار خاص در برخی از بیانیه‌های خاص در یک برنامه می‌دهد. این مسئله به طور رسمی غیرقابل حل است، اما راه‌حل‌های جزئی با استفاده از محدودیت موجود برای این موضوع وجود دارد [10]. برای آزمون منطق، باید ورودی‌های برنامه به گونه‌ای باشند که به گزاره‌ها دست بیاییم و مقادیر لیترال، گزاره‌های اصلی و معیوب را بر اساس مقادیر حقیقی مختلفی مورد ارزیابی قرار می‌دهد. به عبارت دیگر، ورودی‌های برنامه‌ها باید به گونه‌ای باشند که یک معیار را برآورده کنند.

تشخیص نقص‌هایی که برای گزاره‌های DNF کمینه برقرار است برای گزاره‌های DNF غیرکمینه برقرار نیست. این دو مسئله را مطرح می‌کند. یکی اینکه آیا سلسله مراتب نقص‌ها برای گزاره‌های DNF غیر کمینه نیز برقرار است؟ و مسئله دیگر این است که چه نوع از نرم‌افزاری دارای گزاره‌های DNF کمینه است؟ برای مسئله اول، یو و لائو [14] یافته‌هایی از 20 مورد از گزاره‌های DNF غیر کمینه را پیدا کردند که بالای 99٪ از نقص‌ها در شکل 3.1 توسط آزمون‌هایی که به شناسایی نقص‌هایی مشابه با گزاره‌های DNF کمینه پرداخته بودند مورد شناسایی قرار گرفته‌اند. برای مسئله دوم، چاپلنسکی [5] متوجه شده است که 95٪ از 20 256 گزاره در نرم‌افزار اویونیک در DNF کمینه بودند. تنها 3٪ از این گزاره‌ها حاوی پنج یا تعداد بیشتری از لیترال‌های منحصر به فرد بودند اما 80٪ از این گزاره‌های در DNF کمینه بودند.

هنگامی که یک گزاره حاوی کمتر از پنج لیترال منحصر به فرد است، نویسندگان پیش‌بینی می‌کنند که آزمون جامع بهترین گزینه است. این موضوع این مسئله را مطرح می‌کند که کدام نوع نرم افزار به طور کلی دارای حداقل پنج لیترال منحصر به فرد است. چایلنسکی و میلر [6] گزارش دادند که نرم افزارهای آیونیک اغلب با بسیاری از لیترال‌ها تعریف می‌شود و چایلنسکی [5] یک گزاره را با 77 لیترال منحصر به فرد بدست آورده است. بنابراین، معیار کمینه MUMCUT باید برای آزمایش نرم افزار آیونیک مفید باشد.

7. نتیجه‌گیری

آزمون منطق نیاز به راه‌حلهایی کارآمد برای مسئله تبادل کاهش اندازه مجموعه آزمون بدون قربانی کردن تشخیص نقص دارد. چندین معیار منطقی برای حل این مسئله ارائه شده است برخی از آنها از معیارهای دیگری تشکیل شده‌اند. هنگامی که یک معیار مولفه اصلی امکان‌پذیر است، یک مجموعه آزمون کوچک‌تر می‌تواند اغلب خواسته‌های آن را برآورده کند و نیاز به استفاده از یک مجموعه آزمون بزرگ مطابق با معیار والد بدون قربانی کردن تشخیص نقص نیست. در این مقاله رویکردی ارائه شد که در آن یک گزاره DNF کمینه ارائه شده است که تعیین می‌کند که کدام یک از معیارها برای لیترال‌ها و اصطلاحات امکان‌پذیر هستند. این موضوع به نوبه خود می‌تواند معیارهای لازم برای شناسایی نقص‌ها را تعیین کند. این روش بر روی نمونه‌ای از گزاره‌ها مورد آزمون قرار گرفت (که دارای 5 الی 13 لیترال منحصر به فرد بودند) و سبب کاهش اندازه مجموعه آزمون شد (بدون قربانی کردن تشخیص نقص) تا حدی شد که در صورتی که امکان‌سنجی را در نظر نمی‌گرفتیم تنها 1.30٪ از اندازه اصلی مورد نیاز بود. تحقیقات آینده باید با تمرکز بر روی تعیین ورودی‌هایی که سبب می‌شوند لیترال‌ها مقادیر مورد نیاز برای رضایت معیار کمینه MUMCUT داشته باشند باید صورت بگیرد.

References

- [1] A.T. Acree, T.A. Budd, R.A. DeMillo, R.J. Lipton, and F.G. Saywood. "Mutation Analysis," Technical Report GIT-ICS-79/08, School of Information and Computer Science, Georgia Institute Of Technology, Atlanta GA, Sept 1979.
- [2] T.Y. Chen and M.F. Lau. An Empirical Study on the Effectiveness of the Greedy MUTP Criterion. *Software Engineering: Education and Practice*, 1998. Proceedings, 1998 International Conference. January, 1998. Pages 338 – 344.
- [3] T.Y. Chen and M.F. Lau. Test Case Selection Strategies Based on Predicates. *Software Testing, Verification, and Reliability*, 11(1):165-180, November 2001.
- [4] T.Y. Chen, M.F. Lau, and Y.T. Yu. MUMCUT: A Fault-Based Criterion for Testing Predicates. *Software Engineering Conference*, December, 1999. (APSEC '99) Proceedings. Sixth Asia Pacific. Takamatsu, Japan. Pages 606-613.
- [5] J.J. Chilenski. An Investigation of Three Forms of the Modified Condition Decision Coverage (MCDC) Criterion. Final Technical Report, DOT/FAA/AR01/18, April 2001.
- [6] J.J. Chilenski and S.P. Miller. Applicability of Modified condition/decision coverage to Software Testing. *IEE/BCS Software Engineering Journal*, 9(5): 193-200, September 1994.
- [7] G. Kaminski, G. Williams, and P. Ammann. Reconciling Perspectives of Logic Testing for Software. *Software Testing, Verification, and Reliability*, 18(3):149-188, September 2008.
- [8] D. Richard Kuhn. Fault Classes and Error Detection Capability of Predicate Based Testing. *ACM Transactions on Software Engineering and Methodology*, 8(4): 411-424, October 1999.
- [9] M.F. Lau and Y.T. Yu. An Extended Fault Class Hierarchy for Predicate-Based Testing. *ACM Transactions on Software Engineering and Methodology*, 14(3): 247-276, July 2005.
- [10] A.J. Offutt and J. Pan. Automatically Detecting Equivalent Mutants and Infeasible Paths. *Software Testing, Verification, and Reliability*, 7(3):165-192, September 1997.
- [11] V. Okun, P. Black, and Y. Yesha. Comparison of Fault Classes in Specification-Based Testing. *Information & Software Technology*, 46(8): 525-533, June 2004.
- [12] Chang-ai Sun, Yunwei Dong, R. Lai, K.Y. Sim, and T.Y. Chen. Analyzing and Extending MUMCUT for Fault-based Testing of General Boolean Expressions. *Proceedings of the Sixth IEEE International Conference on Computer Information Technology*, September, 2006. Pages: 184-189.
- [13] E. Weyuker, T. Goradia, and A. Singh. Automatically Generating Test Data from a Predicate. *IEEE Transactions on Software Engineering*, 20(5): 353-363, May 1994.
- [14] Y.T. Yu and M.F. Lau. Comparing Several Coverage Criteria for Detecting Faults in Predicates. In *Proceedings QSIC 2004: 4th International Conference on Quality Software*, Pages 14-21.