

از بین هفت مسئله‌ی داده شده چهار مورد را به دلخواه خود حل کنید. بیش از چهار مورد اگر تحویل داده شود موارد اضافه به دلخواه کنار گذاشته خواهد شد و فقط چهار مورد آن مورد تصحیح قرار می‌گیرد. همچنین اگر دانشجویی اقدام به حل مسئله‌ای کند که دانشجویان دیگر کمتر آن را حل کرده باشند نمره‌ی تشویقی خواهد داشت (البته مشروط به اینکه آن مسئله صحیح حل شده باشد).

(۱) یک روش کشف اشکال که تعمیمی بر روش DWC است به این صورت عمل می‌کند که به جای اینکه از یک واحد F دو عدد داشته باشیم و ورودی یکسان x را به هر دو واحد داده و دو خروجی y (که به صورت $y=F(x)$ هستند) را مقایسه کنیم، از واحد F واحد F^{-1} که عمل معکوس آن را انجام می‌دهد داریم. سپس بعد از محاسبه‌ی $y=F(x)$ اقدام به محاسبه‌ی معکوس آن یعنی $x=F^{-1}(y)$ نموده و سپس دو مقدار x را مقایسه می‌کنیم. مزایای این روش این است که اولاً diversity دارد و ثانیاً برخی مواقع سربار پیاده‌سازی F^{-1} از F کمتر است که آن را گزینه‌ی مناسبی می‌کند. اما عیب آن این است که موجب افزونگی زمانی می‌گردد. بر اساس این روش فقط یکی از دو مورد زیر را انتخاب نموده و طراحی مربوطه را انجام دهید:

(الف) یک مدار دیجیتال قابل سنتز که با زبان Verilog یا VHDL توصیف شده است و عمل تقسیم ورودی‌های ۱۶ بیتی را انجام می‌دهد (اعداد ۱۶ بیتی Fixed Point هستند به نحوی که Point همیشه در وسط قرار داشته و ۸ بیت بخش صحیح و ۸ بیت بخش کسری است) توصیف کنید. با استفاده از روش مذکور سخت‌افزار کشف اشکال را به آن اضافه کنید. کل سیستم (تقسیم کننده به همراه مدار کشف اشکال) باید به شکل قابل سنتز توصیف گردد و برای یک تکنولوژی مقصد دلخواه گزارش کنید که سربار سخت‌افزاری این روش کشف خطا چقدر است. ابزار سنتز مورد استفاده و تکنولوژی مقصد را گزارش کنید.

(ب) قطعه کدی به زبان C که عمل تقسیم چندجمله‌ای‌ها را انجام می‌دهد توسعه دهید و مکانیسم کشف خطای فوق را برای آن بکار بگیرید (فقط زبان C قابل قبول است). این برنامه دو چندجمله‌ای با ضرایب ثابت را در ورودی دریافت نموده و سپس خارج قسمت و باقی‌مانده را در خروجی ارائه می‌دهد. سربار زمانی ناشی از اعمال این روش کشف خطا را اندازه‌گیری نموده و گزارش کنید. در گزارش خود ذکر کنید که برای اجرای قطعه کد مذکور از چه سیستمی استفاده کرده‌اید (با گزارش مشخصات سخت‌افزاری سیستم مورد استفاده و سیستم عامل آن).

(۲) یک روش کشف خطای CFE که می‌توان آن را در سطح کد منبع (source code) به جای سطح زبان ماشین (زبان اسمبلی) اعمال نمود به این شکل است که برای تمام توابع (روتین‌ها یا متوذهای) موجود در یک برنامه در ابتدا که وارد آن تابع می‌شویم id خاص آن تابع را در یک پشته push می‌کنیم و سپس هر زمان که هنگام بازگشت از آن تابع است

همان id را از پشته pop می‌کنیم و واریسی می‌کنیم که آیا مقدار id همخوانی دارد یا خیر. به این ترتیب اگر خروج نادرستی از یک تابع (روتین یا متود) وجود داشته باشد، یا ورود اشتباهی به میانه‌ی آن وجود داشته باشد و یا حتی اگر پشته دچار اشکال گردد توسط این روش کشف می‌شود. اعمال چنین روشی بصورت دستی (manual) توسط توسعه دهنده‌ی کد کار آسانی نبوده و خود دارای احتمال بالای خطای انسانی (خطای طراحی یا خطای پیاده‌سازی) است. روش صحیح اعمال چنین روش‌هایی این است که اعمال آن‌ها خودکار گردد. یک قطعه کد به زبان Python بنویسید که برای قطعه‌کدهای نوشته شده به زبان C++ این کار را انجام دهد. به این شکل که کد Python شما کد اولیه به زبان C++ را دریافت می‌کند و سپس یک کد C++ جدید تولید می‌کند که روش کشف خطای مذکور به آن اعمال شده است.

۳) یک قطعه کد Maple یا Mathematica توسعه دهید که توصیف یک نمودار RBD را در قالب یک فایل متنی در ورودی دریافت کند (برای توصیف RBD در یک فایل متنی خودتان یک قرارداد ارائه دهید) و سپس عبارت جبری Reliability کل سیستم را در خروجی ارائه دهد. دقت کنید که خروجی باید عبارت جبری باشد و نه یک عدد.

۴) برای یک سرور یک واحد یدک داغ (Hot Spare) با Coverage کشف خطایی برابر C داریم. با خراب شدن سرور اگر خرابی آن کشف گردد به سرعت یدک داغ جایگزین آن می‌گردد و تبدیل به واحد اصلی می‌گردد و همزمان عملیات تعمیر سرور خراب شده شروع می‌شود (از زمان جایگزینی واحد یدک به جای واحد اصلی صرف نظر کنید چراکه این فرض به خصوص با توجه به اینکه از یدک داغ استفاده می‌گردد فرض مناسبی است). پس از تعمیر سرور خراب شده از آن به عنوان یدک داغ جدید برای سروری که در حال کار است استفاده می‌شود. با فرض اینکه نرخ خرابی ۸ و نرخ تعمیر μ باشد مقدار Availability حالت پایدار این سیستم را با استفاده از مدل مارکوف به دست آورید. برای حل مسئله و از جمله حل دستگاه معادلات دیفرانسیل می‌توانید از بسته‌های نرم‌افزاری دلخواه خود استفاده کنید.

۵) برای یک حسگر آنالوگ (حسگر را به دلخواه خود انتخاب کنید) واحد TMR را به شکل عملی و با استفاده از یک برد نهفته (مانند برد آردوینو) پیاده‌سازی کنید. برد نهفته باید خود عمل رأی‌گیری را برای حسگرها انجام دهد. پیاده‌سازی باید عملی و کامل باشد و شبیه‌سازی مورد قبول نیست. همچنین باید در عمل نشان دهید که با قطع کردن ارتباط یکی از حسگرها با برد نهفته کماکان مقدار صحیح توسط برد نهفته به دست می‌آید. در این پیاده‌سازی از دو لایه TMR استفاده می‌شود. از یک سو سه عدد حسگر وجود دارد که بین مقادیری که آن‌ها ارائه می‌کنند رأی‌گیری می‌شود و از سوی دیگر مقدار هر حسگر نیز سه بار خوانده می‌شود و بین آن‌ها رأی‌گیری می‌شود.

۶) برای یک پردازنده‌ی چند هسته‌ای (که می‌تواند پردازنده‌ی کامپیوتر شخصی خودتان باشد) یک برنامه بنویسید که یکی از هسته‌ها نقش Watchdog Processor برای هسته‌ی دیگر داشته باشد. برنامه باید بر روی یک پردازنده‌ی واقعی اجرا

گردد و شبیه‌سازی مورد قبول نیست. روش کشف اشکال به این شکل است که برای برنامه‌ای که روی یک هسته اجرا می‌گردد باید گراف جریان کنترل (Control Flow Graph) به هسته‌ی دیگر داده شود. همچنین باید پیام‌های مناسب از هسته‌ی مجری به هسته‌ی Watchdog ارسال گردد که هسته‌ی Watchdog بتواند نظارت را انجام دهد. این سیستم را برای برنامه Selection Sort پیاده‌سازی کنید.

(۷) برخی از الگوریتم‌ها دارای درجاتی از تحمل‌پذیری ذاتی در مقابل انواعی از اشکال‌ها هستند. به عنوان نمونه بسیاری از روش‌های یادگیری ماشین قابلیت ذاتی بازیابی در مقابل انواعی از اشکال‌های گذرا و ماندگار را دارا هستند. برای حل مسئله‌ی حرکت یک ربات در داخل یک maze از الگوریتم یادگیری ماشین Q-Learning استفاده کنید. کد مربوطه را با زبان Python توسعه دهید (اگر از نمونه‌های آماده‌ی کد Python استفاده کنید نیز مورد قبول است). با استفاده از کد خود آزمایش کنید که این الگوریتم در مقابل اشکال‌های گذرا و ماندگار در جدول Q-Table چه رفتاری دارد (اشکال گذرا در جدول به این معنی است که داده‌ای در جدول بطور تصادفی آسیب می‌بینید ولی قابلیت بازنویسی روی آن وجود دارد. اشکال ماندگار به این معنی است که خانه‌ای از جدول بطور ثابت مقداری به خود می‌گیرد که قابل تغییر نیست). همچنین آزمایش کنید که اگر وضعیت (state) محیط که به agent منتقل می‌شود دچار اشکال گذرا گردد یا action که agent نسبت به محیط می‌خواهد نشان دهد دچار اشکال گذرا گردد چه تأثیری بر رفتار الگوریتم دارد. آزمایش‌ها را برای مقادیر و زمان‌های گوناگون تکرار کنید و نتایجی که مشاهده می‌کنید را ثبت نموده و گزارش کنید. اشکال‌ها را در دو نوع از شرایط ایجاد کنید. یکی وقتی که الگوریتم به وضعیت پایدار رسیده و راه‌حل را یافته است و بعد از یافتن راه‌حل شروع به ایجاد اشکال‌ها کنید. یکی دیگر وقتی که الگوریتم هنوز راه‌حلی پیدا نکرده و در حال یادگیری اولیه است در حین یادگیری اولیه اشکال‌ها را ایجاد کنید.

موفق باشید

اجلالی