
بسم الله الرحمن الرحيم

دانشگاه تهران

پردیس دانشکده‌های فنی

دانشکده مهندسی برق و کامپیوتر

استفاده از افراز فضای ورودی در تعریف تابع سازواری آزمون جستجو محور

نگارش

غزل نیسی مینایی

استاد راهنما

دکتر رامتین خسروی

پروژه نهایی مقطع کارشناسی رشته مهندسی کامپیوتر

تابستان ۱۴۰۱

تعهدنامه اصالت اثر

بسمه تعالی

اینجانب غزل نیسی مینایی تایید می‌کنم که مطالب مندرج در این پایان‌نامه حاصل تلاش اینجانب است و به دستاوردهای پژوهشی دیگران که در این نوشته از آن‌ها استفاده شده است مطابق مقررات ارجاع گردیده است. این پایان‌نامه قبلا برای احراز هیچ مدرک هم سطح یا بالاتر ارائه نشده است.

نام و نام خانوادگی دانشجو:

غزل نیسی مینایی

امضای دانشجو:

چکیده

آزمون نرم‌افزار امروزه بخش مهمی از فرایند تولید نرم‌افزار را تشکیل می‌دهد. تاکنون روش‌های مختلفی برای تولید آزمون خودکار برای نرم‌افزار پیشنهاد شده است. یکی از حوزه‌هایی که به این مهم توجه دارد، حوزه آزمون بر مبنای توصیف است. در این روش‌ها توصیفی از سیستم به عنوان مدلی برای استخراج آزمون استفاده می‌شود. برای استخراج آزمون از توصیف تابعی نیز روش‌های مختلفی وجود دارد. یکی از این روش‌ها، رهیافت مکاشفه‌ای است که در آن تلاش می‌شود با بررسی فضای جست و جو و با توجه به محدودیت‌های تعریف شده برای تابع سازواری، پوشش توصیف بیشینه گردد.

گرچه رهیافت یادشده می‌تواند پوشش مناسبی برای توصیف تابعی ارائه دهد، نمیتوان از پوشش مناسب بر روی سیستم تحت آزمون مطمئن شد. چرا که توصیف مذکور در سطح انتزاع بالاتری نسبت به سیستم تحت آزمون قرار دارد. در این تحقیق، برای بهبود پوشش آزمون، از روش افراز فضای ورودی بهره می‌گیریم. در این روش، دامنه ورودی سامانه تحت بررسی، بر اساس نیاز و نوع سامانه به بخش‌هایی افراز می‌شود و مقادیر آزمون از این بخش ها انتخاب می‌گردد. این مقادیر می‌توانند با یکدیگر ترکیب شوند و بدین ترتیب آزمایه‌ها باید ترکیباتی از این بخش ها را شامل شوند.

افراز فضای ورودی با مباحثه با کارشناسان دامنه مبنا تهیه و سپس با بهره گیری از زبان پایتون پیاده سازی شده است. با توجه به ویژگی‌های مورد نیاز سامانه مبنا و همچنین تسلط گروه پژوهشی، هسته معاملات بورس بدین منظور انتخاب شده است. سامانه انتخاب شده علاوه بر پیچیدگی بالا، توصیف تابعی جامعی از بخش‌های کارکردی به زبان هسکل دارد که مقایسه پوشش آن با پوشش سامانه اصلی، از انگیزه‌های این تحقیق است.

بررسی نتایج حاصل از اجرای آزمون جستجو محور به روش‌های مختلف نشان میدهد که گرچه با پوشش فضای ورودی فعلی بهبود قابل توجهی در نتایج پوشش سامانه حاصل نشد، با توجه به بررسی‌های صورت گرفته امید است با تکمیل پوشش فضای ورودی همه بخش‌ها، نتایج آزمون بهبود پیدا کند.

کلمات کلیدی:

افراز فضای ورودی - آزمون جستجو محور - آزمون مبتنی بر توصیف - هسته معاملات بورس

فهرست مطالب

۱. مقدمه

۱،۱. اهمیت و انگیزه تحقیق

امروزه سرویس‌های نرم افزاری بسیاری وجود دارند که به خاطر نیازمندی‌های کارکردی و غیرکارکردی، پیچیدگی‌های ساختاری بسیاری دارند. برای نمونه می‌توان از سرویس‌های تهیه تاکسی‌های آنلاین، هسته معاملات بورس و فروشگاه‌های آنلاین نام برد. در این سیستم‌ها مجموعه بزرگی از نیازمندی‌های کارکردی برای سیستم تعریف شده است که بسیاری از آن‌ها منطق‌های پیچیده‌ای را پیاده‌سازی می‌کنند.

با توجه به اهمیت این گونه از سیستم‌ها، صحت عملکرد سیستم اهمیت بسیار بالایی دارد. گرچه در این سیستم‌ها معمولا مجموعه قابل قبولی از آزمون‌های واحد[۱] برای هر قسمت از سیستم تولید می‌شود، نمی‌توان از درستی کارکرد سیستم به صورت یک مجموعه کامل اطمینان حاصل کرد. علی رغم اینکه در هر بخش به صورت مجزا اطمینان نسبی نسبت به صحت عملکرد وجود دارد ولی مشکلات بسیاری در تعامل این بخش‌ها با یکدیگر و همچنین مفروضات این بخش‌ها نسبت به یکدیگر می‌تواند وجود داشته باشد که مورد بررسی قرار نگرفته است. در نتیجه نیاز به مجموعه آزمون دیگری است که سیستم را به صورت یک مجموعه یک‌جا مورد بررسی قرار دهد. این مجموعه آزمون، آزمون‌های سرتاسری[۲] نام دارند.

از آنجا که در تولید آزمون‌های سرتاسری، هدف پوشش تمامی نیازمندی‌های کارکردی است، نیاز است روشی برای تولید و اطمینان از پوشش کامل مجموعه آزمون ایجاد شود. این نکته باعث ایجاد روش‌های آزمون مبتنی بر مدل شده است. در این روش‌ها، بر اساس نیازمندی‌های کارکردی سیستم، مدلی ساخته می‌شود که با پوشش کامل می‌توان اطمینان حاصل کرد که سیستم مورد آزمون از نظر تمامی نیازمندی‌های کارکردی مورد بررسی قرار گرفته و عملکرد مناسبی دارد. فایده دیگری که استفاده از روش‌های آزمون مبتنی بر مدل به همراه دارد، خودکارسازی فرآیند تولید آزمون است.

روش‌های مدل‌سازی بسیاری برای تولید مدلی که بتواند برای تولید خودکار آزمون مورد استفاده قرار گیرد، وجود دارد. در برخی از این روش‌ها سعی می‌شود تا در یک زبان تابعی[۳] (مانند هسکل[۴]) برای توصیف سیستم به روش توصیف تابعی استفاده شود. یکی از راهکارهای اعمال این روش به‌کارگیری روش‌های مکاشفه‌ای (مثل

---

[۱] Unit Tests

[۲] End to End Tests

[۳] Functional Programming Language

[۴] Haskell

الگوریتم ژنتیک[۵]) برای تولید آزمایه‌ها است. در این روش تابع سازواری[۶] طوری تعیین میشود، که پوشش توصیف را بیشینه کنند. [۱]

مشکل این رهیافت در تولید خودکار آزمایه به روش مکاشفه‌ای از روی توصیف تابعی، این است که توصیف مذکور در سطح انتزاع بالاتری از برنامه تحت آزمون قرار دارد. به همین دلیل صرف به دست آوردن پوشش کامل روی توصیف تابعی به معنای به دست آوردن پوشش مناسب بر روی برنامه تحت آزمون نیست. در این تحقیق از روش افراز فضای ورودی[۷] بهره میگریم تا علاوه بر ارزیابی مجموعه آزمون‌های پیشین از نظر پوشش فضای ورودی، تابع سازواری را طوری تغییر دهیم که علاوه بر پوشش توصیف تابعی، پوشش مناسبی از داده‌های فضای ورودی را نیز در نظر بگیرد.

۱،۲. تعریف مسئله

مهم ترین هدف انجام این تحقیق ارائه یک افراز فضای ورودی برای سیستم مبنا و تعریف این افراز به عنوان یکی از محدودیت‌های تابع سازواری در روش مکاشفه‌ای تولید خودکار آزمایه است، تا فضای ورودی سیستم نیز تحت پوشش آزمایه‌ها قرار گیرد و به این ترتیب بخش‌های پیمایش نشده سیستم نیز توسط این روش آزمون، آزموده شود.

سیستم مبنا در این پژوهش، هسته معاملات بورس است که در این پژوهش برای چندین بخش اصلی این سیستم روش افراز فضای ورودی را اعمال می‌کنیم. این بخش‌ها سفارش جدید، سفارش دارای حداقل کمیت و سفارش دوبخشی[۸] هستند که در بخش پیش زمینه منطق حاکم بر هر یک را بررسی می‌کنیم. بررسی افراز فضای ورودی سایر بخش های سیستم به تحقیقات آتی موکول گردیده است.

همچنین در این پژوهش آزمایه‌های تولید شده قبلی با روش مکاشفه‌ای مبتنی بر توصیف تابعی را از نظر پوشش مناسبه بر روی فضای ورودی مورد بررسی قرار می‌دهیم.

---

[۵] Genetic Algorithm

[۶] Fitness Function

[۷] Input Space Partitioning

[۸] Iceberg Order

۱,۳. روش انجام پژوهش

روش انجام این پژوهش، مباحثه با کارشناسان دامنه مبنا یعنی هسته معاملات بورس برای درک درست این سامانه است. از آنجا که سامانه انتخابی، بسیار پیچیده و دارای بخش‌های کارکردی فراوان است این مباحثه گامی حیاتی برای فهم درست سیستم مذکور است.

سپس به ارائه افراز فضای ورودی مناسب برای کارکردهای انتخابی این سیستم می‌پردازیم. انجام صحیح این بخش در گروی همان درک صحیحی است که در گام قبل به دست آمده است.

در آخر نیز به پیاده‌سازی این افرازها می‌پردازیم تا بتوانیم هم آزمایه‌های پیشین را بررسی کنیم و هم پوشش این افرازها را به عنوان محدودیتی در تابع سازواری آزمون مکاشفه‌ای تعریف کنیم.

۱,٤. ارزیابی و دستاوردهای تحقیق

برای درک درست هسته معاملات بورس، مباحثاتی با کارشناسان دامنه مذکور صورت گرفت تا بخش های اصلی سیستم مشخص و درک شود. بعد از این گام فضای ورودی بخش‌های انتخابی افراز گردید و پیاده سازی شد. سپس، تابع سازواری به گونه ای تغییر یافت که بیشینه کردن پوشش افرازهای فضای ورودی در کنار پوشش کد از اهداف آن باشد. این آزمایش بار دیگر با شرط محدودیت بودن پوشش فضای ورودی و هدف بودن پوشش کد برای تابع سازواری تکرار گردید. منظور از محدودیت در تابع سازواری این است که این تابع باید آزمایه‌هایی ایجاد می‌کرد که حتما همه بخش‌های افراز را در خود داشته باشد.

۱,٥. خلاصه و ساختار فصل‌ها

در فصل آتی، به شرح سه مفهوم کلیدی و پیش‌زمینه این پژوهش خواهیم پرداخت و اصول و مبنای کلی آن‌ها را توضیح خواهیم داد. سپس در فصل سوم به توضیح روش انجام پژوهش، افرازهای صورت گرفته، تصمیم‌ها و انتخاب‌های طول مسیر، اصول و پیاده سازی نهایی می‌پردازیم. فصل چهارم به بررسی روش ارزیابی توصیف و نتایج این ارزیابی اختصاص دارد. در فصل آخر نیز یک خلاصه کلی از هدف پروژه و تعریف مسئله ارائه خواهد شد و در مورد اصلی‌ترین نتایج به دست آمده و همچنین موضوعات پیشنهادی برای تحقیقات آتی سخن خواهیم گفت.

۲. پیش زمینه

۲/۱. آزمون خودکار

آزمون خودکار[۹] به معنای استفاده از نرم‌افزار در اجرای آزمون‌ها، مقایسه نتیجه آزمون با نتیجه مورد انتظار، آماده کردن پیش نیازهای آزمون و سایر فرایندهای آزمودن نرم‌افزار و گزارش گیری است. آزمون خودکار هزینه و زحمت آزمودن نرم افزار را کاهش می‌دهد و این امکان را به ما می‌دهد که بارها و بارها یک آزمون را با مقادیر یکسان تکرار کنیم. همچنین خطای انسانی را کاهش می‌دهد [۲].

در میان فعالیت‌هایی که برای آزمودن نرم‌افزار انجام می‌شود، تولید آزمایه‌ها از مهم‌ترین و چالش برانگیزترین بخش‌ها می‌باشد چون تاثیر بسیار زیادی بر کارآمدی آزمون دارد. به همین دلیل، در سال‌های اخیر پژوهش‌های بسیاری برای تولید خودکار آزمایه‌ها[۱۰] صورت گرفته است. در نتیجه این تلاش‌ها، روش‌های زیادی برای تولید خودکار آزمایه‌ها ابداع و بررسی شده‌اند [۳].

یکی از این روش های تولید خودکار آزمایه‌ها، روش مکاشفه‌ای[۱۱] است. در این روش، با استفاده از الگوریتم‌های جست و جو محور سعی داریم تا آزمایه‌ها یا داده‌های ورودی آزمایه‌ها را با هدایت یک تابع سازواری تولید کنیم. تابع سازواری، مجموعه اهداف مورد نظر در آزمودن را مانند بیشینه کردن پوشش کد، در خود دارد. طراحی این تابع سازواری مهم‌ترین مسئله در تولید خودکار آزمایه‌ها در روش مکاشفه‌ای است. این تابع برای هدایت یک الگوریتم (مانند ژنتیک) برای جست و جوی ورودی‌های آزمایه‌ها طراحی می‌شود تا بتواند آن‌هایی که اهداف آزمون را برآورده می‌کنند پیدا کند [۳].

از آنجا که فضای آزمایه‌ها و ورودی‌های برنامه حتی برای یک برنامه کوچک می‌تواند بسیار بزرگ باشد، برای اندازه گیری این موضوع که آیا یک هدف آزمودن در آزمایه‌ها تامین شده است یا نه از معیارهای پوشش[۱۲] استفاده می‌کنیم. این معیارها یک روش مناسب برای تعیین این مسئله است که چه زمانی آزمایه‌ها کافی هستند و نیازی به تولید بیشتر آن ها نیست [۲].

---

[۹] Test Automation

[۱۰] Automated Test Case Generation

[۱۱] Search-based

[۱۲] Coverage Criteria

۲,۲. افراز فضای ورودی

افراز فضای ورودی یک روش در آزمون نرم‌افزار است که در آن دامنه ورودی مانند آرگومان‌های ورودی توابع، متغیرهای سراسری، اشیایی که حالت کنونی برنامه را نشان می‌دهند و سایر ورودی‌ها، به بخش‌هایی افراز می‌گردد. در این افراز فرض می‌شود که همه بخش‌ها دارای مقادیر با ارزشی از نظر آزمایش برنامه است. سپس مقادیر آزمون از هر بخش انتخاب می‌شوند. در این روش هر افراز بر اساس یکی از خصایص برنامه، ورودی‌های برنامه یا محیط آن می‌باشد. [۲]

برای مدل سازی فضای ورودی دو روش مبتنی بر تعامل[۱۳] و مبتنی بر کارکرد[۱۴] را در ادامه بررسی می‌کنیم. در روش مبتنی بر تعامل، هر پارامتر را جدا از سایرین در نظر می‌گیریم. در این روش، تشخیص خصایص ساده است اما از همه اطلاعاتی که درباره برنامه داریم استفاده نمیشود و ممکن است تقسیم بندی ناقص باشد. همچنین بعضی از کارکردهای برنامه ممکن است وابسته به ترکیب بعضی مقادیر باشد که در این روش به دلیل جدا بررسی شدن هر پارامتر این ترکیبات نادیده گرفته می‌شود. [۲]

در روش مبتنی بر کارکرد، خصیصه‌ها براساس کارکردهای سیستم تحت آزمون انتخاب می‌گردند، به همین دلیل دانشی که از دامنه داریم در مدل سازی فضای ورودی نمود پیدا می‌کند. در این روش، تشخیص خصایص بسته به پیچیدگی سیستم می‌تواند سخت باشد. اما از آنجا که اطلاعات بیشتری را در نظر می‌گیرد، می‌تواند به نتایج بهتری منجر شود. دو روش مفید در تشخیص این خصایص، پیش شرط‌ها و پس شرط‌ها هستند. [۲]

برای اندازه گیری پوشش، می‌توان مقادیر مختلف از افرازها را با یکدیگر ترکیب نمود. برای انتخاب این ترکیب، روش‌های مختلفی وجود دارد، مثلا می‌توان همه بخش‌ها را با یکدیگر ترکیب نمود، یا از ترکیب‌های دوتایی[۱۵] استفاده نمود. از آنجا که بیشتر مشکلات نرم افزاری از ترکیب بین یک یا دو پارامتر ایجاد می‌گردد [۴] ترکیب های دوتایی برای اندازه گیری پوشش، قابل قبول هستند.

۲,۳. مبانی هسته معاملات بورس

هسته معاملات، سفارش‌هایی از نوع خرید یا فروش را دریافت می‌کند و سفارش‌های دریافتی را با سفارش‌های موجود در صف سفارش‌ها تطبیق[۱۶] می‌دهد. این تطبیق به خصوصیت‌های مختلف سفارش‌ها مانند قیمت و حجم آن‌ها وابسته است. وقتی دو سفارش مطابق می‌شوند، یک معامله صورت می‌گیرد. این تطبیق پذیری براساس

---

[۱۳] Interface-Based Input Domain Modeling

[۱۴] Functionality-Based Input Domain Modeling

[۱۵] Pair-Wise

[۱۶] Matching

الگوریتم تطبیق و ویژگی‌های سفارش‌های مورد معامله انجام می‌شود. در ساده ترین حالت، تطبیق پذیری دو سفارش را بررسی می‌کنیم. [۵]

فرض می‌کنیم، در صف سفارش‌ها، تعدادی سفارش وجود دارد. این سفارش ها به ترتیب بر اساس قیمت و زمان ورود مرتب شده‌اند. یک سفارش خرید با حجم V_1 و قیمت C_1 وارد می‌شود. سفارش‌ها می‌توانند با سفارش مخالف خود (خرید با فروش و فروش با خرید) منطبق شوند. در این حالت سفارش جدید وارد شده می‌تواند با سفارش های فروش با قیمت C_1 و پایین تر منطبق شود. پس در صورت وجود چنین سفارش فروشی، یک معامله صورت می‌گیرد. حجم معامله شده برابر با مینیمم حجم سفارش‌های مورد معامله خواهد بود و اگر مقداری از یک سفارش باقی بماند، وارد صف خواهد شد.

آنچه بررسی کردیم، ساده ترین حالت تطبیق پذیری برای یک سفارش محدود[۱۷] بود. سفارش محدود، سفارش خرید یا فروشی است که در قیمت تعیین شده توسط مشتری یا بهتر انجام میشود. انواع دیگری از سفارش ها مانند سفارش بازار به محدود[۱۸] و سفارش بازار[۱۹] وجود دارد که در اینجا بررسی نمی‌کنیم. این سفارش‌ها می‌توانند به روش‌های مختلفی به هسته معاملات وارد شوند. یکی از این روش‌ها، سفارش دو بخشی[۲۰] است. سفارش دو بخشی، دارای دو پارامتر حجم نهان و حجم آشکار است. حجم آشکار برای همه معامله گران قابل مشاهده است و در هر مطابقت حداکثر به میزان حجم آشکار، معامله صورت خواهد گرفت. این سفارش، با از دست دادن اولویت زمانی دوباره به صف بازمی‌گردد تا بخش دیگری از آن معامله شود. در صورت اتمام حجم آشکار، از حجم پنهان آن کم شده و به حجم آشکار اضافه می‌شود تا زمانی که حجم پنهان نیز تمام شود. علاوه بر سفارش‌های دوبخشی، انواع دیگری از روش‌های وارد شدن سفارش‌ها به هسته معاملات وجود دارد که از جمله آن‌ها می‌توان انجام و ابطال[۲۱] و همه یا هیچ[۲۲] را نام برد. این دو روش آخر را در اینجا بررسی نمی‌کنیم.

سفارش‌های موجود در هسته معاملات، دارای مشخصه حجمی هستند. یکی از این مشخصه ها حداقل کمیت است. سفارش‌هایی که این مشخصه را دارند باید به اندازه حداقل کمیت ذکر شده در بدو ورود معامله شوند. اگر سفارشی در صف نباشد که با سفارش وارد شده به اندازه حداقل کمیت یا بیشتر معامله شود، سفارش ورودی لغو میگردد.[۵]

---

[۱۷] Limit Order
[۱۸] Market to Limit Order
[۱۹] Market Order
[۲۰] Iceberg
[۲۱] Fill and Kill
[۲۲] All or None

آنچه بررسی شد، بخش‌های محدودی از هسته معاملات بورس بود که در ادامه این تحقیق استفاده شده است. بدیهی است این سامانه دارای کارکردهای بسیار بیشتری همانند بررسی‌های پیش از معامله، انواع اعتبار زمانی و... است که بررسی همه آن‌ها از حوصله این بحث خارج است.

۳. روش پیشنهادی

۳,۱. مقدمه

برای فهمیدن بخش‌های مختلف سامانه هسته معاملات بورس که در این پژوهش مبنا قرار گرفته است، با کارشناسان دامنه مورد نظر مباحثه صورت گرفت. با توجه به این مباحثات، یکی از مهم‌ترین کارکردهای این سامانه، ثبت سفارش جدید، از نوع خرید یا فروش است. یک سفارش از بخش های زیر تشکیل شده است:

```
@dataclass
class Order:
    order_id: ...
    broker_id: ...
    shareholder_id: ...
    price: ...
    qty: ...
    side: ...
    min_qty: ...
    fak: ...
    disclosed_qty: ...
```

قطعه کد ۱ – بخش‌های یک سفارش

این بخش‌ها به ترتیب، شناسه سفارش، شناسه بازار، شناسه سهامدار، قیمت سفارش، حجم سفارش، خرید یا فروش بودن، حداقل کمیت، آیا از نوع انجام و ابطال است یا خیر و میزان حجم آشکار هستند.

نوع دیگری از سفارش‌ها وجود دارد که نسبتا پیچیده تر از سفارش‌های عادی است، این سفارش‌ها دارای حداقل کمیت هستند. همانطور که پیشتر نیز توضیح داده شد، تفاوت این نوع سفارش‌ها با سفارش‌های عادی این است که، سفارش‌های دارای حداقل کمیت، یک حداقل مقداری دارند که در صورت تطبیق نیافتن با سفارش دیگری، پذیرفته نمی‌شوند.

نوع دیگری نیز از سفارش‌ها وجود دارد که مکانیزم بسیار پیچیده‌ای دارد. این سفارش‌ها به نام سفارش‌های دوبخشی شناخته می‌شوند. همانطور که قبلا بررسی شد، این سفارش‌ها یک مقدار قابل دیدن و یک مقدار نهان دارند. معاملات بر اساس بخش قابل دیدن آن‌ها صورت می‌گیرد و وقتی این بخش تمام شد، سفارش با حجم باقی مانده مجدد در صف قرار می‌گیرد. همانطور که روشن است، این نوع سفارش‌ها در ترکیب با سایر سفارش‌ها و اولویت‌ها حالت‌های پیچیده به وجود می آورند. این نوع سفارش‌ها طبق تصمیم‌گیری گروه پژوهشی در افرازها بررسی شده اما پیاده‌سازی نشده است.

در ادامه افرازهای ایجاد شده برای سفارش‌ها و حالت‌های یاد شده را بررسی می‌کنیم و دلیل هر یک از تصمیم گیری‌ها را بیان می‌کنیم. سپس پیاده سازی این افرازها و ادغام شدن آن‌ها با سیستم اصلی و تابع سازواری را مورد بررسی قرار می‌دهیم.

برای افراز فضای ورودی در کارکردهای یاد شده از روش مبتنی بر کارکرد استفاده شده است. دلیل این انتخاب پیچیدگی این کارکردها است که اجزای مختلف آن‌ها باید در کنار هم بررسی شوند و روش مبتنی بر رابط نمی‌تواند ارتباط بین اجزا را به خوبی نشان دهد.

۳،۲. افراز فضای حالت سیستم

۳،۲،۱. سفارش جدید

سفارش‌ها در سامانه معاملات بورس می‌توانند از نوع خرید باشند یا فروش. در شروع کار این سفارش‌ها را جداگانه به اساس سه شرط زیر دسته بندی کرده بودیم:

- حالت تطبیق شدن: در این روش، تطبیق شدن سفارش خرید(یا فروش) با سفارش‌های فروش(یا خرید) که از قبل در صف هستند بررسی می‌شود.
- پس شرط: در این روش، شرایط صف و سفارش‌ها بعد از انجام معاملات بررسی می‌شود.
- تعداد تراکنش‌ها: به ازای هر تطبیق پذیری یک تراکنش ثبت می‌شود. در این روش تعداد تراکنش‌ها بر اثر انجام معاملات را مبنای افراز قرار می‌دهیم.

بعد از انجام تقسیم بندی براساس نکات بالا و بررسی نتایج در گروه پژوهشی، تصمیم بر آن شد که تغییراتی در این تقسیم بندی‌ها صورت گیرد. اول آن که از آنجا که ماهیت خرید و فروش در این کارکرد تفاوت چندانی ندارد، نیازی نیست که جداگانه بررسی شوند. دوم آن که تعداد تراکنش‌ها ارزش بیشتری نسبت به تقسیم بندی‌ها قبلی ایجاد نمی‌کند و بهتر است برای کاهش حجم کد و افزایش سرعت اجرا حذف گردد.

در نهایت افرازها برای کارکرد سفارش جدید به این صورت درآمدند:

افراز بر اساس تطبیق سفارش خرید یا فروش	
سفارش خریدی (فروشی) که تازه رسیده ...	

با دقیقا یک سفارش فروش(خرید) در صف منطبق می‌شود.	۱
با بخشی از یک سفارش فروش(خرید) در صف منطبق می‌شود.	۲
با تعدادی سفارش فروش(خرید) در صف منطبق می‌شود.	۳
با هیچ سفارشی منطبق نمی‌شود.	۴
بخشی از آن منطبق می‌شود و بخشی باقی می‌ماند.	۵

جدول ۱ - افراز بر اساس تطبیق سفارش خرید یا فروش

در جدول ۱ افراز بر اساس تطبیق سفارش جدید نمایش داده شده است. در این افراز عملا حالت تعداد تراکنش‌ها پوشش داده می‌شود چرا که تعداد تراکنش‌ها می‌توانند مثلا هیچی، یکی یا بیشتر از یک باشد و در جدول ۱، مورد شماره ۴ معادل هیچی در تراکنش‌ها خواهد بود. همچنین مورد شماره ۳ حالت بیشتر از یکی را پوشش خواهد داد. همانطور که مشخص است، ردیف های ۱ و ۲ حالت یک تراکنش را در خود جای داده‌اند. بنابراین دیگر نیازی به بررسی افراز براساس تعداد تراکنش‌ها نداریم و این افراز اطلاعات ارزشمندی اضافه نخواهد کرد.

یکی دیگر از نکات قابل ذکر در جدول ۱ این است که از آنجا ماهیت سفارش خرید با فروش تفاوتی ندارد، جدا افراز کردن آنها کار مفیدی نخواهد بود، بنابراین تصمیم اولیه مبنی به جدا افراز کردن آنها بدین شکل تغییر کرد.

افراز بر اساس پس شرط سفارش‌های خرید یا فروش	
در پایان، حداقل یک سفارش فروش(خرید) دست نخورده باقی می‌ماند.	۱
در پایان، همه سفارش‌های فروش(خرید) تمام شده.	۲
در پایان، تنها بخشی از یک سفارش فروش(خرید) باقی مانده.	۳

جدول ۲ - افراز بر اساس پس شرط سفارش‌های خرید یا فروش

در جدول شماره ۲ افراز بر اساس حالت پایانی نمایش داده شده است. در این حالت صف معاملات را بعد از انجام همه معاملات بررسی می‌کنیم. همانطور که در بخش پیش زمینه ذکر شد، حالت پس شرط از مهم‌ترین افرازهای مبتنی بر کارکرد است که می‌تواند آزمون‌های ارزشمندی ایجاد کند.

۳,۲,۲. سفارش دارای حداقل کمیت

سفارش‌های دارای حداقل کمیت، اگر در بدو ورود به اندازه حداقل کمیت ذکر شده معامله نشوند، لغو می‌گردند. پس از آن، سفارش ذکر شده مانند یک سفارش محدود خواهد بود که پیشتر بررسی شد.

بر همین اساس، شرایط سفارش ورودی در زمان ورود و شرایط صف سفارش ها در آن لحظه است که می‌تواند تفاوت معناداری با حالت‌های پیشین برای این نوع سفارش‌ها ایجاد کند. در اولین بررسی، این منطق را بر اساس پیش شرط‌های حداقل کمیت و صف معاملات افراز کردیم. جدول زیر این تقسیم بندی را نشان می‌دهد:

افراز بر اساس پیش شرط سفارش های دارای حداقل کمیت	
سفارش خریدی (فروشی) که تازه رسیده و حداقل کمیت دارد....	
با هیچ سفارش فروشی(خریدی) منطبق نمی‌شود چون حداقل کمیت لازم وجود ندارد. (سفارشی که از نظر قیمت می‌تواند منطبق شود وجود دارد اما مقدارش کمتر از حداقل کمیت است) (این سفارش رد می‌شود)	۱
با هیچ سفارش فروشی(خریدی) منطبق نمی‌شود چون حداقل کمیت لازم وجود ندارد. (سفارشی که از نظر قیمت می‌تواند منطبق شود وجود ندارد)	۲
کامل منطبق می‌شود. (حداقل کمیت با حجم برابر است یا مجموع سفارش‌های دیگر به قدری بزرگ است که کامل می‌تواند سفارش وارد شده را پوشش دهد.)	۳
به اندازه حداقل کمیت منطبق می‌شود و بقیه آن باقی می‌ماند.	۴
بیشتر از اندازه حداقل کمیت منطبق می‌شود ولی بخشی نیز باقی می‌ماند.	۵

جدول ۳ - اولین نسخه افراز بر اساس پیش شرط سفارش‌های دارای حداقل کمیت

در جدول شماره ۳ اولین نسخه افراز بر اساس پیش شرط سفارش‌های دارای حداقل کمیت نمایش داده شده است. همانطور که دیده می‌شود، این افراز علاوه بر پیش شرط حداقل کمیت، پیش شرط صف و حتی بخشی از اتفاقات بعدی را نیز در نظر می‌گیرد. این تقسیم بندی گرچه کامل به نظر می‌اید، بیش از حد به جزئیات پرداخته است. با تقسیم بندی ساده تر و با ترکیب آن با افرازهای قبلی می‌توان به نتیجه‌ای مشابه دست یافت. توجه به این نکته ضروری است که هر چقدر افرازها پیچیده‌تر باشد، کد نوشته شده برای در نظر گرفتن آن‌ها طولانی تر و کندتر خواهد شد.

بنابراین افراز بالا را فقط برای شرایط حداقل کمیت در بدو ورود در نظر می‌گیریم و افراز را به شکل زیر تغییر می‌دهیم:

افراز بر اساس پیش شرط سفارش های دارای حداقل کمیت	
سفارش خریدی (فروشی) که تازه رسیده و حداقل کمیت دارد ...	
مقدار حداقل کمیت برابر با حجم دارد.	۱
مقدار حداقل کمیت بین صفر تا حجم دارد.	۲
مقدار حداقل کمیت برابر صفر دارد.	۳

جدول ۴ - نسخه نهایی افراز بر اساس پیش شرط سفارش‌های دارای حداقل کمیت

در جدول شماره ۴، حالت ساده‌تر افراز بر اساس پیش شرط سفارش‌های دارای حداقل کمیت نمایش داده شده است. این افراز فقط براساس مقدار حجم سفارش و حداقل کمیت آن صورت گرفته است و بسیار ساده‌تر از تقسیم بندی پیشین است.

در این افراز حالت‌های زیر به صورت ضمنی ایجاد می‌گردد:

- میزان حداقل کمیت اقناع می‌شود.
- میزان حداقل کمیت اقناع نمی‌شود و سفارش رد می‌شود.
- حداقل کمیت صفر است که یعنی این سفارش یک سفارش محدود عادی است که پیشتر بررسی شد.

۳,۲,۳. سفارش دو بخشی

سفارش‌های دو بخشی دارای دو حجم نهان و آشکار هستند. حجم آشکار برای همه معامله کنندگان نمایش داده می‌شود و پس از اتمام، از حجم نهان کم شده و به آن اضافه می‌شود. این سفارش‌ها در ترکیب با صف سفارش‌های دیگر حالت‌های بسیار پیچیده‌ای را می‌توانند ایجاد کنند. از این رو یافتن یک افراز مناسب برای آن ها کار ساده‌ای نیست.

در یک افراز می‌توان سفارش دو بخشی را بر اساس قرارگیری در صف تقسیم بندی کرد. در این حالت بررسی می‌کنیم که سفارش یادشده براساس مقدار آشکار خود و سفارش‌های دیگر در صف و معامله انجام شده در بدو ورود در کجای صف قرار می‌گیرد. در این افراز فرض شده که حتما یک تطبیق برای سفارش دو بخشی تازه وارد صورت می‌گیرد.

براساس آنچه گفته شد جدول زیر را تشکیل می‌دهیم:

افراز بر اساس قرارگیری سفارش دو بخشی در صف	
سفارش خریدی (فروشی) که تازه رسیده و دوبخشی است....	
بالای صف قرار می‌گیرد، مقدار آشکار آن منطبق شده و به پایین می‌رود. (یک سفارش دیگر اولویت می‌گیرد)	۱
بالای صف قرار می‌گیرد، کمتر از مقدار آشکار آن منطبق می‌شود و همانجا می‌ماند.	۲
بالای صف قرار می‌گیرد، مقدار آشکار آن منطبق شده و در بالا باقی می‌ماند. (مجدد اولویت می‌گیرد)	۳

جدول ۵ -افراز سفارش دو بخشی بر اساس قرارگیری در صف

جدول ۵ افراز سفارش دو بخشی بر اساس معامله اول و قرارگیری در صف را نشان می‌دهد. در ردیف اول، سفارش دو بخشی اولویت زمانی خود را از دست داده و پایین رفته است. در ردیف دوم از آنجا که مقدار منطبق شده از حجم آشکار آن کمتر است، بالای صف می‌ماند و در ردیف آخر با وجود معامله بخش آشکار باز بالای صف باقی می‌ماند.

در این افراز فرض شده سفارش دو بخشی از ابتدا بالای صف قرار دارد که این فرض بسیاری از حالت‌ها را نادیده می‌گیرد. البته بر همین اساس می‌توان حالت‌های دیگر را نیز اضافه نمود.

در تقسیم بندی بعدی، افراز را براساس پس شرط در معاملات انجام شده انجام می‌دهیم:

افراز بر اساس پس شرط سفارش دو بخشی	
سفارش دو بخشی در پایان ...	
حداقل یک معامله با مقدار کمتر از مقدار آشکار اولیه دارد.	۱
بیش از مقدار آشکار اولیه معامله شده است.	۲
معامله نشده است.	۳

جدول ۶ -افراز سفارش دو بخشی بر اساس پس شرط

جدول ۶، افراز سفارش‌های دو بخشی را بر اساس پس شرط معاملات صورت گرفته نمایش می‌دهد. برنامه نویسی برای این افراز ساده‌تر از افراز یادشده در جدول ۵ است. چرا که فقط حالت پایانی را باید در نظر بگیریم.

در این بخش افرازهای مختلف و تصمیم گیری‌های مربوط به بخشی از کارکردهای سامانه مبنا، یعنی هسته معاملات بورس مورد بررسی قرار گرفت. در ادامه کد نوشته شده براساس این افرازها را مورد بررسی قرار می‌دهیم.

۳،۳. برنامه نوشته شده برای آزمودن هر افراز

۳،۳،۱. آزمایه‌ها

برای نوشتن کد مربوط به هر افراز، نخست باید ساختار یک آزمایه را بررسی کنیم. آزمایه‌ها در سامانه مبنا دارای تعدادی دستور هستند که براساس آن‌ها سفارش‌های مختلف، پیش شرط‌ها و دیگر بخش‌ها ایجاد می‌شود. سپس نتیجه هر دستور مشخص می‌گردد. بر همین اساس کلاس آزمایه و دستور را برای افرازها به شکل زیر در نظر میگیریم:

```
class Command:
    command = None
    args = None
```

```
    is_order = False
    is_in_queue = False
    is_trade = False
    order_type = None

    def create_command(self, line_command):

        self.command, *self.args = line_command.split()
        if self.command == NEW_ORDER_REQ or self.command == ORDER_QUEUE:
            if self.command == NEW_ORDER_REQ:
                self.is_order = True
            else:
                self.is_in_queue = True

            self.order_type, self.order_id, self.broker_id,
self.shareholder_id, self.price, \
            self.quantity, self.side, self.min_quantity, \
            self.is_fak, self.peak_size = self.args

            self.quantity = int(self.quantity)
            self.min_quantity = int(self.min_quantity)
            self.price = int(self.price)

        elif self.command == TRADE_CMD:
            self.is_trade = True
            self.price, self.quantity, self.buy_id, self.sell_id = self.args

            self.quantity = int(self.quantity)
            self.price = int(self.price)
```

قطعه کد ۲ - کلاس دستور

در کلاس دستور همه انواع دستورات از یکدیگر جدا نمی‌شوند و براساس آنچه برای کد افرازها نیاز داریم فقط دستورات مربوط به معامله و سفارش را جدا می‌کنیم و از پیچیدگی بیشتر پرهیز می‌کنیم. دستورات در این کلاس براساس آرگومان‌های خود ذخیره می‌شوند تا بعدا استفاده شوند.

```
class Testcase:
    all_commands: list = []

    def parse(self, lines):
        for line in lines:
            if line.isspace() or line == '':
                continue
            new_command = Command()
```

```
            new_command.create_command(line)
            self.all_commands.append(new_command)

    def get_all_orders(self):
        all_orders = []
        for cmd in self.all_commands:
            if cmd.is_order:
                all_orders.append(cmd)
        return all_orders

    def get_all_trades(self):
        all_trades = []
        for cmd in self.all_commands:
            if cmd.is_trade:
                all_trades.append(cmd)
        return all_trades

    def get_final_queue(self):
        final_queue = []
        last_orders_idx = None
        for cmd_idx in range(len(self.all_commands)):
            if self.all_commands[cmd_idx].command == 'Orders':
                last_orders_idx = cmd_idx

        if not last_orders_idx:
            return None

        for cmd_idx in range(last_orders_idx, len(self.all_commands)):
            cmd = self.all_commands[cmd_idx]
            if cmd.is_in_queue:
                final_queue.append(cmd)
        return final_queue
```

قطعه کد ۳ - کلاس آزمایه

همانطور که ذکر شد، یک آزمایه از تعدادی دستور تشکیل شده است. بر همین اساس کلاس آزمایه لیستی از دستورات را نگه می‌دارد. این دستورات، با متد پارس کردن ایجاد شده و در لیست قرار می‌گیرند. براساس افرازها، به همه سفارش‌های موجود در آزمایه، همه معاملات صورت گرفته و وضعیت نهایی صف نیاز داریم که به همین دلیل سه متد بعدی ایجاد شده است.

حال که به یک آزمایه، دستورات آن و ویژگی های دسترسی داریم، می‌توانیم کد مربوط به افرازها را بنویسیم.

۳,۳,۲. سفارش جدید

در این بخش کد مربوط به افرازهای سفارش جدید را بررسی میکنیم. ابتدا کد هر ردیف مربوط به جدول ۱، افراز بر اساس تطبیق سفارش خرید یا فروش، بررسی می‌گردد.

سفارش خریدی (فروشی) که تازه رسیده ...

- با دقیقا یک سفارش فروش(خرید) در صف منطبق می‌شود.

```
def p1_1_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()
    test_case_trades = testcase.get_all_trades()

    for order in test_case_orders:
        for trade in test_case_trades:
            if order.order_id == trade.buy_id or order.order_id ==
trade.sell_id:
                if order.quantity == trade.quantity:
                    return True

    return False
```

قطعه کد ۴ - کد افراز جدول ۱ ردیف ۱

- با بخشی از یک سفارش فروش(خرید) در صف منطبق می‌شود.

```
def p1_2_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()
    test_case_trades = testcase.get_all_trades()

    for order in test_case_orders:
        for trade in test_case_trades:
            if order.order_id == trade.buy_id or order.order_id ==
trade.sell_id:
                if order.quantity < trade.quantity:
                    return True

    return False
```

قطعه کد ۵ - کد افراز جدول ۱ ردیف ۲

- با تعدادی سفارش فروش(خرید) در صف منطبق می‌شود.

```
def p1_3_isp(testcase: Testcase):
```

```
    test_case_orders = testcase.get_all_orders()
    test_case_trades = testcase.get_all_trades()

    for order in test_case_orders:
        trade_count = 0
        final_quantity = 0
        for trade in test_case_trades:
            if order.order_id == trade.buy_id or order.order_id ==
trade.sell_id:
                if order.quantity > trade.quantity:
                    trade_count += 1
                    final_quantity += trade.quantity

        if trade_count > 1 and final_quantity == order.quantity:
            return True

    return False
```

قطعه کد ۶ - کد افراز جدول ۱ ردیف ۳

- با هیچ سفارشی منطبق نمی‌شود.

```
def p1_4_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()
    test_case_trades = testcase.get_all_trades()

    for order in test_case_orders:
        trade_count = 0
        for trade in test_case_trades:
            if order.order_id == trade.buy_id or order.order_id ==
trade.sell_id:
                trade_count = + 1

        if trade_count == 0:
            return True

    return False
```

قطعه کد ۷ - کد افراز جدول ۱ ردیف ۴

- بخشی از آن منطبق می‌شود و بخشی باقی می‌ماند.

```
def p1_5_isp(testcase: Testcase):
```

```
    test_case_orders = testcase.get_all_orders()
    test_case_trades = testcase.get_all_trades()

    for order in test_case_orders:
        trade_count = 0
        final_quantity = 0
        for trade in test_case_trades:
            if order.order_id == trade.buy_id or order.order_id ==
trade.sell_id:
                if order.quantity > trade.quantity:
                    trade_count += 1
                    final_quantity += trade.quantity

        if trade_count > 0 and final_quantity < order.quantity:
            return True

    return False
```

قطعه کد ۸ - کد افراز جدول ۱ ردیف ۵

حال به بررسی هر ردیف مربوط به جدول ۲، افراز بر اساس پس شرط سفارش‌های خرید یا فروش، می‌پردازیم.

- در پایان، حداقل یک سفارش فروش(خرید) دست نخورده باقی می‌ماند.

```
def p2_1_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()
    test_case_final_queue = testcase.get_final_queue()

    for order in test_case_orders:
        for in_queue in test_case_final_queue:
            if order.order_id == in_queue.order_id and order.quantity ==
in_queue.quantity:
                return True

    return False
```

قطعه کد ۹ - کد افراز جدول ۲ ردیف ۱

- در پایان، همه سفارش‌های فروش(خرید) تمام شده.

```
def p2_2_isp(testcase: Testcase):
    test_case_final_queue = testcase.get_final_queue()
    return True if not test_case_final_queue else False
```

قطعه کد ۱۰ - کد افراز جدول ۲ ردیف ۲

- در پایان، تنها بخشی از یک سفارش فروش(خرید) باقی مانده.

```
def p2_3_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()
    test_case_final_queue = testcase.get_final_queue()

    for order in test_case_orders:
        for in_queue in test_case_final_queue:
            if order.order_id == in_queue.order_id and order.quantity >
in_queue.quantity:
                return True

    return False
```

قطعه کد ۱۱ - کد افراز جدول ۲ ردیف ۳

۳،۳،۳. سفارش دارای حداقل کمیت

در این بخش کد مربوط به افرازهای سفارش‌های دارای حداقل کمیت را بررسی می‌کنیم. بدین منظور کد مربوط به هر ردیف جدول جدول ۴، نسخه نهایی افراز بر اساس پیش شرط سفارش‌های دارای حداقل کمیت، بررسی می‌گردد.

سفارش خریدی (فروشی) که تازه رسیده و حداقل کمیت دارد ...

- مقدار حداقل کمیت برابر با حجم دارد.

```
def p3_1_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()

    for order in test_case_orders:
        if order.min_quantity == order.quantity:
            return True

    return False
```

قطعه کد ۱۲ - کد افراز جدول ۴ ردیف ۱

- مقدار حداقل کمیت بین صفر تا حجم دارد.

```
def p3_2_isp(testcase: Testcase):
```

```
    test_case_orders = testcase.get_all_orders()

    for order in test_case_orders:
        if 0 < order.min_quantity < order.quantity:
            return True

    return False
```

قطعه کد ۱۳ - کد افراز جدول ۴ ردیف ۲

- مقدار حداقل کمیت برابر صفر دارد.

```
def p3_3_isp(testcase: Testcase):
    test_case_orders = testcase.get_all_orders()

    for order in test_case_orders:
        if order.min_quantity == 0:
            return True

    return False
```

قطعه کد ۱۴ - کد افراز جدول ۴ ردیف ۳

۳,۳,٤. ادغام شدن با سیستم موجود

برای این کار تک تک آزمایه‌های ایجاد شده را توسط توابع بالا (افرازها) باید بررسی کنیم و مشخص کنیم چه تعداد از آن‌ها مقدار صحیح باز می‌گردانند. این عدد بر تعداد کل افرازها، مبنای نمره دهی برای تابع سازواری قرار می‌گیرد. این کد در گروه پژوهشی نوشته و بررسی گردید.

٤. ارزیابی

در این بخش ابتدا آزمایه‌های از پیش تولید شده توسط سامانه آزمون هسته معاملات را بررسی می‌کنیم که ببینیم این آزمون چقدر در پوشش فضای ورودی سامانه موفق بوده است. برای این کار تعدادی مجموعه آزمون از پیش تهیه شده را انتخاب می‌کنیم و بررسی می‌کنیم که آزمایه‌ها در آن چه تعداد از افرازهای بحث شده در فصل قبل را پوشش می‌دهند.

نتیجه این بررسی برای سه مجموعه آزمون را مشاهده می‌کنید:

توضیح جداول: از آنجا در هر مجموعه تست، تعداد بسیار زیادی آزمایه وجود دارد، جدول‌ها در دو بخش گزارش شده‌اند. ستون سمت چپ، برابر با نام تابع معادل افراز فضای ورودی است که در فصل ۳ شرح داده شد. ستون آخر (در بخش دوم هر جدول) تعداد آزمایه‌هایی را نشان می‌دهد که در آن مجموعه تست، افراز مشخص شده در ستون اول را پوشش داده اند.

مجموعه آزمون ۱	آزمایه ۱	آزمایه ۲	آزمایه ۳	آزمایه ۴	آزمایه ۵	آزمایه ۶	آزمایه ۷	آزمایه ۸	آزمایه ۹	آزمایه ۱۰	آزمایه ۱۱
p1_1_isp	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p1_2_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_3_isp	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_4_isp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
p1_5_isp	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p2_1_isp	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE
p2_2_isp	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE
p2_3_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p3_1_isp	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p3_2_isp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE
p3_3_isp	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE

جدول ۷ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۱ (بخش اول)

آزمایه ۱۲	آزمایه ۱۳	آزمایه ۱۴	آزمایه ۱۵	آزمایه ۱۶	آزمایه ۱۷	آزمایه ۱۸	آزمایه ۱۹	آزمایه ۲۰	آزمایه ۲۱	مجموع
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	2

FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	2
TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	19
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	3
TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	4
FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	16
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	1
TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	5
TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	20
TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	9

جدول ۸ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۱ (بخش دوم)

مجموعه آزمون۲	آزمایه ۱	آزمایه ۲	آزمایه ۳	آزمایه ۴	آزمایه ۵	آزمایه ۶	آزمایه ۷	آزمایه ۸	آزمایه ۹	آزمایه ۱۰	آزمایه ۱۱
p1_1_isp	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p1_2_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_3_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE
p1_4_isp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE
p1_5_isp	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p2_1_isp	FALSE	FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE
p2_2_isp	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE
p2_3_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p3_1_isp	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE
p3_2_isp	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE
p3_3_isp	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	FALSE

جدول ۹ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۲ (بخش اول)

آزمایه ۱۲	آزمایه ۱۳	آزمایه ۱۴	آزمایه ۱۵	آزمایه ۱۶	آزمایه ۱۷	آزمایه ۱۸	آزمایه ۱۹	آزمایه ۲۰	مجموع
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	4
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0

FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	2
TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	FALSE	16
FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	4
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	3
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	15
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	1
TRUE	TRUE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	4
TRUE	TRUE	TRUE	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	13
FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	TRUE	FALSE	12

جدول ۱۰ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۲ (بخش دوم)

مجموعه آزمون۳	آزمایه ۱	آزمایه ۲	آزمایه ۳	آزمایه ۴	آزمایه ۵	آزمایه ۶	آزمایه ۷	آزمایه ۸
p1_1_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_2_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_3_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p1_4_isp	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
p1_5_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p2_1_isp	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE
p2_2_isp	FALSE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE	TRUE
p2_3_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p3_1_isp	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE
p3_2_isp	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	TRUE	TRUE
p3_3_isp	TRUE	FALSE	FALSE	TRUE	FALSE	FALSE	TRUE	TRUE

جدول ۱۱ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۳ (بخش اول)

آزمایه ۹	آزمایه ۱۰	آزمایه ۱۱	آزمایه ۱۲	آزمایه ۱۳	آزمایه ۱۴	آزمایه ۱۵	آزمایه ۱۶	مجموع
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	1
FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	FALSE	0
FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	FALSE	FALSE	2
TRUE	TRUE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	14

FALSE	FALSE	FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	3
FALSE	TRUE	TRUE	FALSE	TRUE	FALSE	FALSE	FALSE	5
TRUE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	TRUE	10
FALSE	FALSE	FALSE	FALSE	TRUE	FALSE	TRUE	FALSE	2
FALSE	FALSE	FALSE	TRUE	FALSE	FALSE	FALSE	FALSE	1
TRUE	FALSE	FALSE	FALSE	TRUE	TRUE	TRUE	TRUE	12
FALSE	TRUE	TRUE	TRUE	TRUE	FALSE	TRUE	FALSE	9

جدول ۱۲ - نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون ۳ (بخش دوم)

خلاصه جداول بالا در جدول زیر گرد آمده:

	مجموعه آزمون۱	مجموعه آزمون۲	مجموعه آزمون۳
p1_1_isp	2	4	1
p1_2_isp	0	0	0
p1_3_isp	2	2	2
p1_4_isp	19	16	14
p1_5_isp	3	4	3
p2_1_isp	4	3	5
p2_2_isp	16	15	10
p2_3_isp	1	1	2
p3_1_isp	5	4	1
p3_2_isp	20	13	12
p3_3_isp	9	12	9

جدول ۱۳ - خلاصه نتایج بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون‌ها

در جدول بالا خلاصه نتایج حاصل از بررسی میزان پوشش افرازهای فضای ورودی توسط مجموعه آزمون‌های ۱ تا ۳ قابل مشاهده است. همانطور که دیده می‌شود، ردیف دوم از جدول ۱ در هیچ یک از آزمایه‌ها پوشش داده نشده است. اما بقیه افرازها کاملا پوشش داده شده‌اند.

نکته قابل توجه دیگر در جداول ۷ تا ۱۲ این است که در بسیاری از افرازها، پوشش پراکنده است. یعنی هر آزمایه بر روی یک بخش خاص متمرکز است و همه بخش‌ها را نمی‌پوشاند.

در گام دوم این ارزیابی، به این سوال می‌پردازیم که اضافه کردن افراز فضای ورودی برای کارکردهای مشخص شده در تابع سازواری آزمون جست و جو محور تا چه حد می‌تواند پوشش کد مبنا را بهبود ببخشد. برای این کار یک بار هدف بیشینه کردن پوشش فضای ورودی را به عنوان هدف دوم تابع سازواری در کنار بیشینه کردن پوشش کد تعریف می‌کنیم و یک بار پوشش فضای ورودی را به عنوان محدودیت تابع سازواری در نظر می‌گیریم.

در جدول زیر نتیجه حاصل از حالت بدون افراز و با افراز فضای ورودی گزارش شده است:

	حالت پایه (بدون افراز فضای ورودی) - درصد	حالت چندهدفه - درصد	حالت محدودیت - درصد
کد هسکل	۷۲	۷۴	۷۰
هسته معاملات مداوم	۷۲	۵۴	۵۲
سفارش محدود	۱۰۰	۵۰	۱۰۰
سفارش دو بخشی	۷۵	۸۱	۳۷

جدول ۱۴ - نتایج حاصل از اجرای تابع سازواری های مختلف

همانطور که در جدول بالا دیده می‌شود با افزودن افراز فضای ورودی چه به عنوان محدودیت چه هدف دوم، پوشش هسته کاهش پیدا می‌کند. در حالت اول، تابع سازواری چندهدفه می‌شود و مجبور است همزمان هر دو مقدار را بیشینه کند و به همین دلیل یکی از هدف‌ها بیشینه می‌شود و هدف دیگه در نقطه نیمه بهینه قرار می‌گیرد که پوشش را کاهش می‌دهد.

در حالت محدودیت، پوشش تمام فضای ورودی به عنوان یک محدودیت تعریف می‌شود. یعنی تابع سازواری اول باید این پوشش را تامین کند سپس پوشش کد را بیشینه کند. این کار باعث می‌شود که از مسیرهای مشخصی که توسط افراز فضای ورودی تعریف شده به هدف نزدیک شود. از آنجا که این افرازها کامل نیست پوشش کاهش پیدا می‌کند.

برای بهبود پوشش کد باید همه فضای ورودی افراز گردد و پوشش کارکردهای محدودی از کد باعث بهبودی خاصی نمی‌شود. با افراز فضای ورودی همه بخش‌ها می‌توانیم تابع سازواری را هدایت کنیم تا از بخش‌های کمتر پیمایش شده نیز آزمایه ایجاد کند.

۵. جمع بندی و نتیجه گیری

همانطور که در بخش مقدمه و تعریف مسئله ذکر شد، هدف اصلی و انگیزه آغاز این پژوهش، بهبود پوشش کد و فضای ورودی سامانه مبنا با استفاده از روش افراز فضای ورودی بخش‌هایی از سامانه مبنا بود. سامانه مبنا در این تحقیق، هسته معاملات بورس بود که از قبل دارای پیشگوی آزمون[۲۳]، توصیف تابعی به زبان هسکل و سامانه آزمون جست و جو محور با تابع سازواری به هدف بیشینه کردن پوشش کد بود.

با توجه به اینکه تولید خودکار آزمایه ها از روی توصیف تابعی صورت می پذیرفت، صرف به دست آوردن پوشش مناسب روی آن نمی توانستیم مطمئن باشیم کد مبنا به خوبی پوشش داده شده است. به همین دلیل پژوهش حاضر با تمرکز بر روی افراز فضای ورودی تعریف شد تا بتوانیم علاوه بر پوشش کد، به پوشش مناسبی از فضای ورودی نیز دست یابیم.

به این منظور، گام‌های مشخصی برای تعریف و پیاده سازی افرازهای مختلف از بخش های مختلف سامانه مبنا تعریف شد که به تفصیل در فصل ۳ بررسی گردید. در این مسیر، انتخاب‌ها و آزمون و خطاهای متعدد منجر به شکل‌گیری تجربه‌ها، اصول و الگوهایی برای مدل‌سازی‌های مشابه گردید. از رئوس نتایج این پژوهش که در قسمت‌های مختلفی از فصول به آن‌ها اشاره شد، می‌توان موارد زیر را برشمرد:

- دستیابی به افرازهای نهایی از بخش‌های مختلف سامانه مبنا
- درک بهتر این سامانه با مباحثه با متخصصان دامنه که پایه‌ای برای تحقیقات آتی می‌گردد.
- بررسی مجموعه آزمون‌های قبلی ایجاد شده توسط سامانه آزمون جست و جو محور از نظر پوشش بخش‌های مختلف فضای ورودی
- تعریف پوشش افرازهای یاد شده در تابع سازواری آزمون جست و جو محور و بررسی نتایج حاصل از آزمون های ایجاد شده توسط تابع جدید

در ادامه این پژوهش، مهم‌ترین کاری که به عنوان تحقیقات آتی و کارهای آینده می‌تواند تعریف شود، تکمیل افراز فضای ورودی برای همه بخش‌های کارکردی سامانه مبنا است. همچنین می‌توان روش پیاده شده در این تحقیق را در سامانه دیگر، مثلا انتخاب واحد دانشگاه پیاده سازی کرد. همچنین برای اندازه گیری پوشش فضای ورودی می‌توان از روش های ترکیبیاتی پوشش استفاده کرد تا تاثیر قسمت‌های مختلف بر روی یکدیگر را بهتر بسنجیم.

---

[۲۳] Test Oracle

٦. مراجع

[۱] A. Zakeriyan, R. Khosravi, H. Safari and E. Khamespanah, "Towards Automatic Test Case Generation for Industrial Software Systems Based on Functional Specifications," in *FSEN*, ۲۰۲۱.

[۲] P. Ammann and J. Offutt, Introduction to Software Testing, Cambridge University Press, ۲۰۱۶.

[۳] S. Anand, E. K. Burke, T. Y. Chen, J. Clark, M. B. Cohen, W. Grieskamp, M. Harman, M. J. Harrold, P. McMinn, A. Bertolino, J. Li and H. Zhu, "An orchestrated survey of methodologies for automated software test case generation," *Journal of Systems and Software,* vol. ۸۶, no. ۸, pp. ۱۹۷۸-۲۰۰۱, ۲۰۱۳.

[۴] R. Kuhn, D. Wallace and A. Gallo Jr, "Software Fault Interactions and Implications for Software Testing," *IEEE Transactions on Software Engineering,* vol. ۳۰, no. ۶, pp. ۴۱۸ - ۴۲۱, ۲۰۰۴.

[۵] ا. حاج یاسینی, ر. کرامتی نیا, ا. خامس پناه و آ. ذاکریان, *پروژه طراحی و تدوین معماری سامانه هسته معاملات بازار نقدی بورس*, ۱۳۹۶.