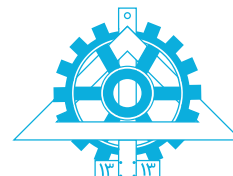




دانشگاه تهران
پردیس دانشکده‌های فنی
دانشکده مهندسی برق و کامپیوتر



روش‌های رسمی در مهندسی نرم‌افزار

تمرین کامپوتری Alloy

احمد پوری حسینی

seyedahmadpourihosseini@gmail.com

مهلت تحویل: نیمه‌شب جمعه ۲۲ آذر

هدف از تمرین، مدل‌سازی چند مسئله به زبان Alloy، چک کردن برخی خواص آن‌ها و همچنین آشنایی با API این ابزار به زبان جاوا است.

۱ مشخص کردن سلول زندانی‌ها

فرض کنید می‌خواهیم به هر زندانی در یک زندان، سلولی نسبت بدهیم. باید مراقب باشیم که دو زندانی که عضو باندهای متفاوتی هستند در یک سلول قرار نگیرند. می‌توانید از قالب زیر برای این کار استفاده کنید:

```
sig Gang{members: set Inmate}
sig Inmate {room: Cell}
sig Cell{}

pred safe {
  // your constraints here
}

pred show {
  // your constraints here
}

run show
```

۱. گزاره‌ی `safe` که بیان‌گر یک تخصیص امن است را کامل کرده، و با اجرای گزاره‌ی `show` که در خود از گزاره‌ی `safe` به نحو مناسب استفاده کرده است، یک نمونه تخصیص که گزاره‌ی `safe` را ارضا کند و یک نمونه تخصیص که گزاره‌ی `safe` را ارضا نکند تولید کنید.

۲. یک گزاره‌ی جدید به اسم `happy` بنویسید که بگوید که افرادی که عضو باندهی هستند فقط با اعضای همان باند هم سلولی هستند. یک تخصیص `safe` لزوماً `happy` نیست. با نوشتن یک `assertion` و استفاده از دستور `check` یک مثال نقض پیدا کنید و توضیح دهید علت این مسئله چیست.

۳. یک قید را در قالب یک `fact` بیان کنید که اطمینان حاصل کند که `safety` واقعاً منجر به `happiness` شود. با استفاده از گزاره‌ی `show` ابتدا اطمینان حاصل کنید که ناسازگاری در مدل ایجاد نکرده باشید، و بعد با استفاده از `assertion` قبلی اطمینان حاصل کنید که دیگر هیچ مثال نقضی وجود ندارد.

¹predicate

۴. با استفاده از Alloy API که می‌توانید یک نمونه‌ی استفاده از آن را در این لینک ببینید، بخش ۱ این سوال را مجدداً به زبان جاوا پیاده‌سازی کنید. توجه کنید که برای استفاده از این API کفایت که هنگام کامپایل و اجرای برنامه‌ی خود، فایل alloyx.x.jar را که دانلود کرده‌اید به classpath اضافه کنید.

۲ دفتر آدرس

در این سوال از شما خواسته می‌شود که تعدادی عبارت و قید برای یک دفتر آدرس دارای چند سطح بنویسید. یک دفتر آدرس یک پایگاه داده‌ی کوچک است که email ها را به اسامی کوچکی که استفاده از آن ها آسان تر از می‌نگارد. به این اسامی کوچکتر alias می‌گوییم و کاربر می‌تواند برای هر email یکی ایجاد کند. همچنین یک group مانند یک alias است ولی به جای ارتباط با یک نفر، ارتباط با چندین نفر را مشخص می‌کند. حال فرض کنید یک مجموعه‌ی Addr داریم که شامل تمامی آدرس email ها است و یک مجموعه‌ی Name داریم که از دو زیر مجموعه‌ی مجزای Alias و Group تشکیل شده. نگاشت از اسم‌ها به آدرس‌ها توسط رابطه‌ی address مشخص می‌شود، ولی یک اسم نه تنها می‌تواند به یک آدرس نگاشت شود، بلکه می‌تواند به یک اسم هم نگاشته شود. همین مسئله است که دفتر آدرس را چند سطحی می‌کند. در ابتدا، قیود زیر را یک انتظار داریم یک دفتر آدرس معتبر ارضاء کند را بنویسید:

۱. دفتر آدرس فاقد دور است. اگر یک اسم را مکرراً با نگاشت خود جایگزین کنیم، هیچ‌گاه به همان اسم نخواهیم رسید.
۲. تمامی alias ها در نهایت به حداقل یک آدرس نگاشته می‌شوند.
- حال، قیود شبیه‌سازی^۲ زیر، که ممکن است هنگام شناخت اولیه‌ی مدل به منظور دیدن نمونه‌های بهتر اضافه کنید را بنویسید:
۳. دفتر آدرس حداقل ۲ سطح دارد.
۴. گروه‌هایی هستند که خالی نباشند.
- در نهایت، برای هر یک از عبارت‌های زیر یک تابع بنویسید که مقدار آن عبارت را به عنوان خروجی برگرداند: راهنمایی می‌توانید از Alloy Evaluator^۳ برای تست این توابع استفاده کنید.
۵. مجموعه اسم‌هایی که حداقل در یک گروه عضو هستند.
۶. مجموعه گروه‌هایی که خالی هستند.
۷. نگاشت از alias ها به آدرس‌هایی که به آن اشاره می‌کنند، چه مستقیم چه غیرمستقیم.

نحوه تحویل

لطفاً فقط کدهای نهایی خود را در پرونده‌ای به نام SID.zip (که SID شماره دانشجویی شماست) پیش از اتمام مهلت تحویل در صفحه وب درس بارگذاری کنید. فایل آپلودی شما باید حاوی یک فایل als به ازای سوال دوم، و یک فایل als نیز به ازای هر بخش از سوال اول باشد، به جز بخش آخر آن که باید در یک فایل java نوشته شود. توجه داشته باشید که این تمرین تحویل حضوری دارد و در صورت حضور نیافتن در تحویل حضوری نمره‌ای از این تمرین دریافت نخواهید کرد.

²simulation constraints

³این ابزار را می‌توانید در صفحه‌ای که نتایج اجرا به شما نشان داده می‌شود، با کلیک بر آیکون Evaluator بیابید.