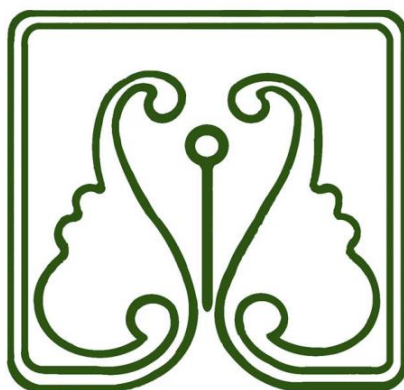


پروژه اصول طراحی کامپایلر



دانشگاه شاهرود

فاز سوم: تحلیلگر معنایی

دکتر فرید فیضی - ترم ۱۴۰۳۲

تحلیل معنایی وظیفه اطمینان از صحت معنایی اعلان‌ها و دستورات یک برنامه است، یعنی اینکه معنای آنها واضح و سازگار با نحوه استفاده از ساختارهای کنترلی و انواع داده‌ها باشد. تحلیل معنایی سومین مرحله از فرآیند ساخت کامپایلر است. در حالی که تحلیل لغوی و نحوی تضمین می‌کند که کد از دستور زبان پیروی می‌کند، تحلیل معنایی، معناداری و سازگاری در چارچوب قوانین زبان را بررسی می‌کند. این مرحله شامل موارد زیر است:

- بررسی تایپ: اطمینان از انجام عملیات بر روی انواع داده‌های سازگار.
- بررسی اسکوپ‌ها: مدیریت اعلان متغیرها و توابع در اسکوپ‌های مختلف.
- تشخیص خطای معنایی: شناسایی مسائلی مانند متغیرهای اعلان نشده یا عدم تطابق تایپ.

در این مرحله، شما تجزیه‌گر خود را برای انجام بررسی‌های معنایی ارتقا می‌دهید و اطمینان حاصل می‌کنید که کد نه تنها از نحو صحیح پیروی می‌کند، بلکه منطقی نیز می‌باشد. تحلیل گر معنایی شما شامل موارد زیر می‌باشد:

ساخت جدول نماد (Symbol Table)

یک جدول نماد برای مدیریت شناسه‌ها (متغیرها، توابع، کلاس‌ها) و اطلاعات مرتبط با آنها پیاده‌سازی کنید.

- مدیریت اسکوپ: از یک رویکرد مبتنی بر پشته برای مدیریت محدوده‌های تو در تو استفاده کنید. هنگام ورود به یک بلوک جدید، یک محدوده جدید را به پشته اضافه کنید و هنگام خروج آن را باز کنید.

- ردیابی شناسه: برای هر شناسه، موارد زیر را ذخیره کنید:

- نام
- نوع (int, float, double, char, bool, string)
- سطح اسکوپ
- ویژگی‌های اضافی در صورت نیاز

سیستم بررسی تایپ

اطمینان حاصل کنید که عبارات و عملیات شامل تایپ‌های سازگار هستند.

- ارزیابی عبارت: قوانین گرامری را برای داشتن اطلاعات تایپ اصلاح کنید. به عنوان مثال، هنگام تجزیه یک عملیات جمع، بررسی کنید که هر دو عملوند از نوع عددی (int یا float) باشند.
- دستورات انتساب: هنگام تجزیه یک انتساب، نوع اعلام شده متغیر را از جدول نماد بازیابی کنید و آن را با نوع عبارتی که انتساب داده می‌شود مقایسه کنید. (برای مثال
- فراخوانی تابع: بررسی کنید که تعداد و نوع آرگومان‌ها با تعریف تابع مطابقت داشته باشد.

تشخیص خطای معنایی

بررسی‌هایی را برای شناسایی خطاهای معنایی پیاده‌سازی کنید. مطمئن شوید که خطاها شامل شماره خط و توضیحات واضح باشند:

- شناسه‌های تعریف نشده: گزارش خطا (Error) هنگام استفاده از یک متغیر یا تابع بدون تعریف قبلی.
- تعریف‌های تکراری: شناسایی و گزارش چندین تعریف از یک شناسه در یک اسکوپ. برای مثال اگر در یک فانکشن دو متغیر با اسم یکسان تعریف شده، باید یک اخطار (Warning) خروجی داده شود.
- عملیات نامعتبر: شناسایی عملیاتی که مجاز نیستند، برای مثال کاربر مجاز نیست که یک داده با تایپ String را در یک متغیر که تایپ آن int تعریف شده ذخیره کند.

بخش امتیازی

برای دریافت نمره کامل بخش امتیازی، پیاده‌سازی همه بخش‌ها الزامی نیست. حداقل ۲ بخش از موارد زیر را انتخاب و پیاده‌سازی کنید:

نقض کنترل دسترسی: اگر زبان شما از visibility modifiers (مثلاً public, private) پشتیبانی می‌کند، اطمینان حاصل کنید که آنها رعایت می‌شوند. برای مثال اگر در یک کلاس یک متغیر private تعریف شده باشد، خارج از کلاس نباید قابلیت دسترسی به آن متغیر وجود داشته باشد.

تبدیل تایپ: در صورت امکان، قوانینی را برای تبدیل تایپ ضمنی یا صریح (implicit یا explicit) پیاده‌سازی کنید.

Shadowing detection: اگر شناسه‌ای در یک اسکوپ، شناسه‌ای را در یک اسکوپ بیرونی یا همان اسکوپ shadow کند، یک اخطار (Warning) باید خروجی داده شود. اطلاعات بیشتر در مورد shadowing در بخش منابع کمکی آورده شده.

بررسی سازگاری تایپ داده برگشتی در توابع: اطمینان حاصل کنید که دستور return در یک تابع، مقداری از تایپ مشخص شده برای تابع را برمی‌گرداند.

ساخت درخت AST: درخت AST را تولید و نمایش دهید.

عدم وجود وابستگی حلقوی: اگر دو کلاس داشته باشیم و در کلاس اول، متغیری از جنس کلاس دوم و در کلاس دوم نیز متغیری از جنس کلاس اول داشته باشیم، وابستگی حلقوی وجود دارد و باید هشدار چاپ شود.

تشخیص متغیرهای استفاده نشده: پس از تجزیه و تحلیل معنایی، متغیرهایی را که اعلان شده‌اند اما هرگز استفاده نشده‌اند، شناسایی کنید.

نکات مهم

- یک فایل pdf به نام report به عنوان گزارش کار از عملکرد پروژه شامل توضیحات مورد نیاز درباره کد را به همراه تصاویری از مراحل اجرا در فایل فشرده پروژه قرار دهید.
- پروژه در قالب گروه های دو نفره تحویل داده شود. تمامی اعضای گروه باید در ارائه ی پروژه حضور فعال داشته باشند.
- مسئولیت اشکال در فایل ارسالی و یا عدم اجرا صحیح پروژه به هر شکلی، بر عهده دانشجو خواهد بود.
- پروژه ی ارسالی با یکدیگر و با پروژه های موجود مقایسه خواهند شد. مسئولیت مشابهت مشکوک/غیرقابل توجیه برعهده ی دانشجو خواهد بود

منابع کمکی

[JFlex User's Manual](#)

[CUP User's Manual](#)

Shadowing in programming