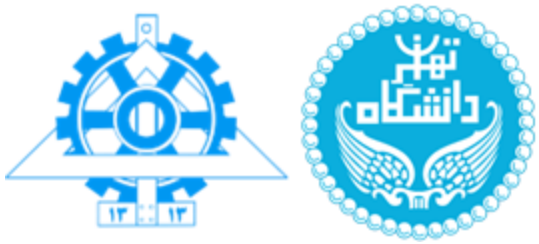

به نام خدا

تمرین کامپیوتری سوم درس طراحی کامپایلر

بهار ۱۴۰۴

فهرست مطالب

مقدمه

در فاز دوم پروژه، خطاهای NameAnalysis بررسی شد و همچنین برخی بهینه‌سازی‌ها روی ast اعمال شد. فاز دیگر در طراحی یک کامپایلر، بررسی خطاهای مربوط به type است. در این فاز از پروژه‌ی درس قصد داریم تعدادی از type error هایی که ممکن است در زبان CPY رخ دهد را بررسی کنیم. همچنین برخی آسیب‌پذیری‌های احتمالی را بررسی می‌کنیم.

برای انجام پروژه، ویزیتور مربوط به TypeChecker و VulnAnalysis را بسازید و ویزیتورهای آن را تکمیل کنید. در صورت تشخیص خطا، یک instance جدید از خطای مربوطه به آرایه typeErrors اضافه کنید و در نهایت آنها را پرینت بگیرید.

آسیب‌پذیری‌های مورد بررسی (فقط دانشجویان مهندسی کامپیوتر)

آسیب‌پذیری‌هایی که انتظار می‌رود در این فاز پیاده‌سازی کنید به شرح زیر است:

1. عدم آزادسازی حافظه تخصیص‌یافته با malloc

این خطا نشان‌دهنده‌ی نشت حافظه (Memory Leak) است که در آن حافظه‌ای با malloc رزرو شده اما با free آزاد نشده است.

```
int main() {
    int* ptr = malloc(sizeof(int) * 10);
    return 0;
}

```

```
Line:2 -> memory not deallocated
```

2. استفاده از متغیر مقداردهی‌نشده (UninitializedVariable)

این خطا نشان می‌دهد که متغیری قبل از مقداردهی اولیه مورد استفاده قرار گرفته که می‌تواند منجر به رفتار غیرقابل پیش‌بینی در برنامه شود.

```c
int main() {
    int x;
    int y = x + 1;
    printf("%d\n", y);
    return 0;
}
```

```
Line:3 -> uninitialized variable used
```

3. تخصیص حافظه با ورودی کاربر (UserControlledMalloc)

این آسیب‌پذیری زمانی رخ می‌دهد که مقدار ورودی از کاربر، مستقیماً یا غیرمستقیم برای تخصیص حافظه با تابع malloc استفاده می‌شود. حتی اگر مقدار بررسی شود، کنترل آن در دست کاربر باقی می‌ماند و باز هم شامل این آسیب‌پذیری می‌شود.

```c
int main() {
    int n;
    scanf("%d", &n);
    int* arr = malloc(n * sizeof(int));
    free(arr);
    return 0;
}
```

```
Line:4 -> user-controlled value used in malloc
```

خطاهای type check

خطاهایی که انتظار می‌رود در این فاز پیاده‌سازی کنید به شرح زیر است:

1. ناسازگاری تایپ آرگومان‌ها با پارامترهای تعریف‌شده (ArgumentTypeMismatch)

این خطا نشان می‌دهد که نوع آرگومان ارسال‌شده به تابع با نوع پارامتر تعریف‌شده در تابع مطابقت ندارد که باعث بروز خطای نوع (type error) در زمان کامپایل می‌شود.

```c
void printInt(int x) {
    printf("%d\n", x);
}

int main() {
    printInt("Hello");
    return 0;
}
```

```
Line:6 -> argument type mismatch
```

2. یکسان نبودن type عملوندهای مربوط به یک عملگر (NonSameOperands)

این خطا زمانی رخ می‌دهد که عملوندهای یک عملگر دارای type های ناسازگار باشند؛ مانند جمع عدد صحیح با رشته که معتبر نیست.

```c
int main() {
    int x = 5;
    float y = x + "abc";
    return 0;
}

```

```
Line:3 -> type mismatch in expression
```

3. ناسازگاری type مقدار بازگشتی با type اعلام‌شده تابع (ReturnTypeMismatch)

این خطا زمانی رخ می‌دهد که مقدار بازگشتی از تابع با نوع اعلام‌شدهٔ آن مطابقت نداشته باشد؛ مانند بازگرداندن رشته به‌جای عدد اعشاری.

```
float getPi() {
    return "3.14";
}

int main() {
    float pi = getPi();
    return 0;
}
```

```
Line:2 -> return type mismatch
```

موفق باشید.