

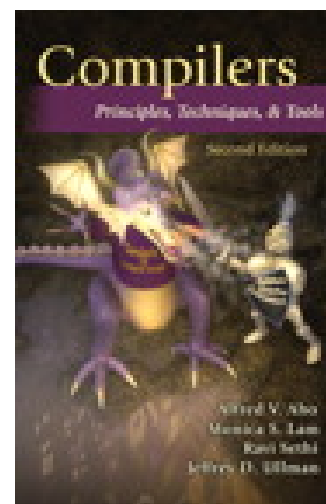
قسمت اول - مقدمه

پائیز ۱۳۸۶ - دانشگاه آزاد مشهد

احمد خادم زاده (khademzadeh@mshdiau.ac.ir)

اصول طراحی کامپیوتر

Compilers: Principles, Techniques, and Tools,
2/E, Alfred V. Aho, Monica S. Lam, Ravi Sethi, Jeffrey D. Ullman
Publisher: Addison-Wesley



ساختار کلاس

- ساعات کلاس، منابع و مراجع، تکالیف، تاریخ امتحان میان ترم و سایر اطلاعات لازم را می توانید در صفحه مربوط به درس در سایت زیر ببینید:

<http://Khademzadeh.mshdiau.ac.ir>

عناوین درس

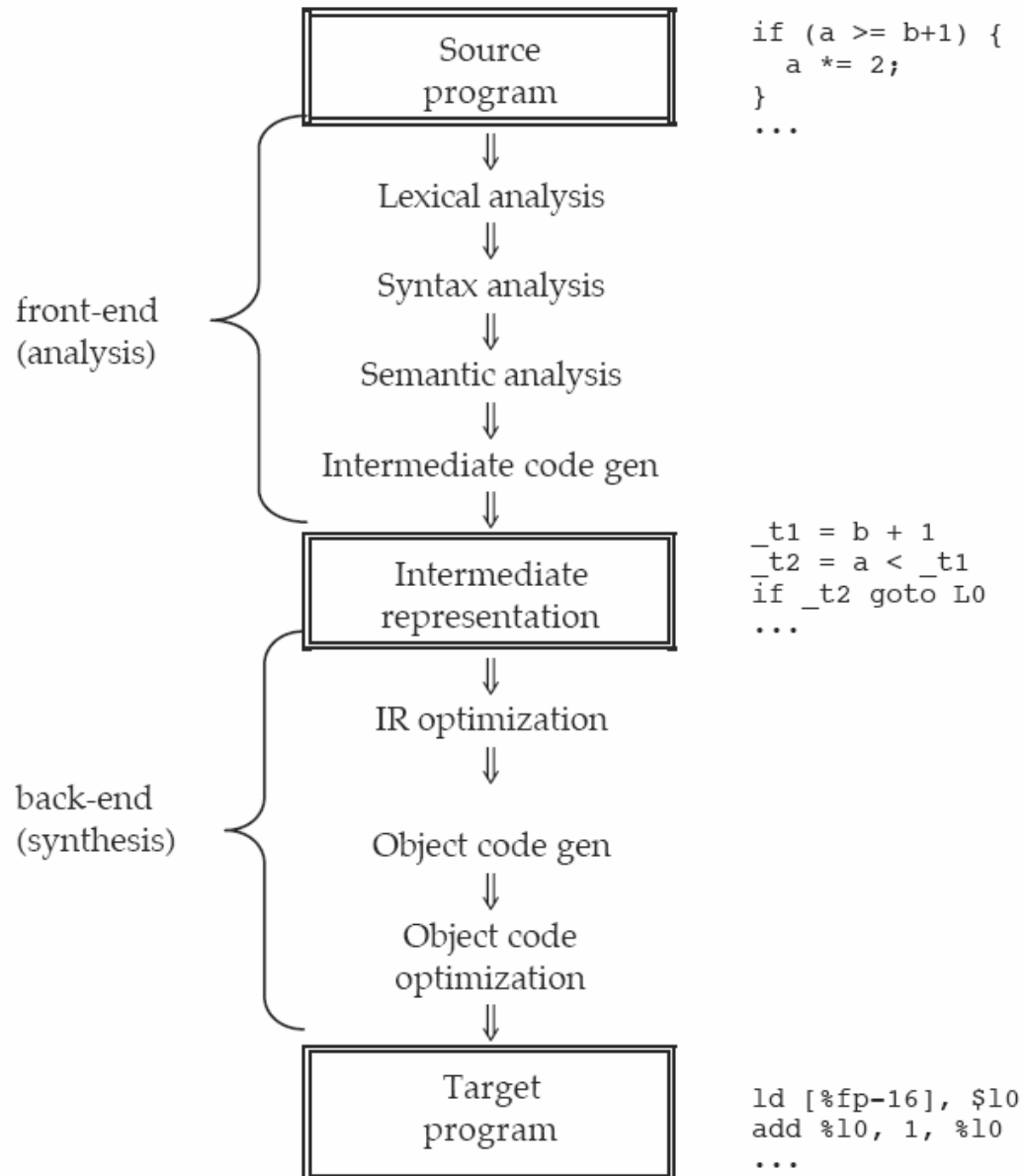
- 1 Lexical analysis, regular expressions
- 2 Formal grammars, Top-down parsing, LL(1)
- 3 Bottom-up parsing, shift-reduce, SLR(1), LR and LALR parsing
- 4 Error repair/recovery, syntax directed translation, Attribute grammars
- 5 Semantic analysis, type systems
- 6 Intermediate representation, three address code
- 7 Runtime environments, code generation

استخوان بندی کامپایلر

کامپایلر چیست؟

- مترجم
- برنامه مبدا
- برنامه مقصد
- گزارش خطا

کامپایلر چگونه کار می کند



تجزیه (analysis) در front end

- تحلیل لغوی (lexical analysis) یا پوش (scanning)
- تحلیل گرامری (نحوی) (syntax analysis) یا تجزیه (parsing)
- تحلیل معنایی (semantic analysis)
- تولید کد میانی (Intermediate code generation)

در ادامه هر یک از این چهار مرحله را بررسی می کنیم.

تحلیل لغوی

- چپ به راست
- نشانه یا نقش (token)
- کلمه (lexeme)

مثالی از تحلیل لغوی

```
int a;  
a = a + 2;
```

int	T_INT	(reserved word)
a	T_IDENTIFIER	(variable name)
;	T_SPECIAL	(special symbol with value of ";")
a	T_IDENTIFIER	(variable name)
=	T_OP	(operator with value of "=")
a	T_IDENTIFIER	(variable name)
+	T_OP	(operator with value of "+")
2	T_INTCONSTANT	(integer constant with value of 2)
;	T_SPECIAL	(special symbol with value of ";")

تجزیه

- گروه بندی نشانه ها
- گرامر های مستقل از متن – CFG
- LHS و RHS
- خروجی این مرحله
- درخت تجزیه – اشتقاق (derivation)
- نشان دهنده چیست ؟

مثالی از تجزیه

Expression \rightarrow Expression + Expression |
Expression – Expression |

...

Variable |
Constant |

...

Variable \rightarrow T_IDENTIFIER
Constant \rightarrow T_INTCONSTANT |
T_DOUBLECONSTANT

مثالی از تجزیه (ادامه)

X + 19

Expression \rightarrow Expression + Expression

\rightarrow Variable + Expression

\rightarrow T_IDENTIFIER + Expression

\rightarrow T_IDENTIFIER + Constant

\rightarrow T_IDENTIFIER + T_INTCONSTANT

تحلیل معنایی

- تشخیص خطاهای معنایی مثل ؟
 - متغیرهای تعریف نشده
 - فراخوانی تابع با عدم رعایت نوع یا تعداد پارامترها
 - نقض دسترسی
 - عدم تطابق نوع (جمع آرایه و عدد)

مثالی از تحلیل معنایی

```
int arr[2], c;  
c = arr * 10;
```

- خطای لغوی؟
- خطای نحوی؟
- خطای معنایی؟

تولید کد میانی

- دو خاصیت کد میانی
- فرم نوشتاری کد میانی
- رایج ترین: کد سه آدرسه (TAC)
- مثال :

$$a = b * c + b * d$$

$$_t1 = b * c$$

$$_t2 = b * d$$

$$_t3 = _t1 + _t2$$

$$a = _t3$$

تولید کد میانی – ادامه

▪ مثال دیگر

```
if (a <= b)
    a = a - c;
c = b * c;
```

```
_t1 = a > b
if _t1 goto L0
_t2 = a - c
a = _t2
L0: _t3 = b * c
c = _t3
```

ترکیب (synthesize) در back end

- بهینه سازی کد میانی (Intermediate code optimization)
- تولید کد مقصد (Object code generation)
- بهینه سازی کد مقصد (Object code optimization)

در ادامه هر یک از این سه مرحله را بررسی می کنیم.

بهینه سازی کد میانی

- ورودی ؟
- خروجی ؟
- سریعتر ، کوتاه تر و کاراتر
- ویژگی های کد خروجی ؟
- کد های غیر قابل دسترس
- متغیر های استفاده نشده
- ضرب در یک و جمع با صفر
- حلقه ها
- زیر عبارات مشترک

بهینه سازی کد میانی - ادامه

- عیب - (حالت پیش فرض)
- مثالی از بهینه سازی کد میانی

```
_t1 = b * c  
_t2 = _t1 + 0  
_t3 = b * c  
_t4 = _t2 + _t3  
a = _t4
```

```
_t1 = b * c  
_t2 = _t1 + _t1  
a = _t2
```

تولید کد مقصد

- خروجی این مرحله
- انتخاب دستورات
- مثال

```
_t1 = b * c  
_t2 = _t1 + _t1  
a = _t2
```

```
ld [%fp-16], %l1      # load  
ld [%fp-20], %l2     # load  
smul %l1, %l2, %l3   # mult  
add %l3, %l3, %l0    # add  
st %l0, [%fp-24]    # store
```

بهینه سازی کد مقصد

- ورودی و خروجی ؟
- استفاده کارا از ثبات های پردازنده
- قابلیت های خاص سخت افزاری

جدول نمادها (symbol table)

- کاربرد در هر دو بخش
- محتویات
- شروع ساخت
- نقش هر مرحله در ساخت و استفاده از این جدول

پوشش خطا

- در چندین مرحله انجام می شود
- اغلب در کدام مراحل ؟
- عملکرد کامپایلر در زمان بر خورد با خطا؟

کامپایلر تک گذره و چند گذره

■ آیا هر مرحله در یک گذر انجام می شود؟

■ Multi-pass

■ One-pass