اصول طراحی کامپایلر

درس ۴

تحلیل لغوی (۳)

Lexical Analysis (3)

کاظم فولادی قلعه
دانشکده مهندسی، پردیس فارابی
دانشگاه تهران

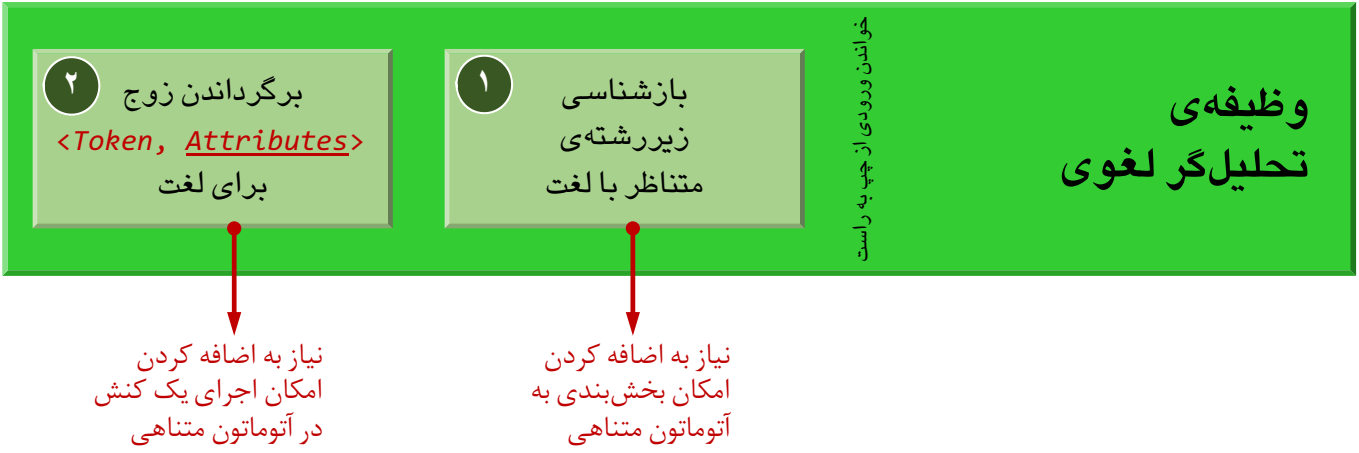
http://courses.fouladi.ir/compiler

تحلیل لغوی (۳)

۱

پیاده‌سازی تحلیل‌گر لغوی

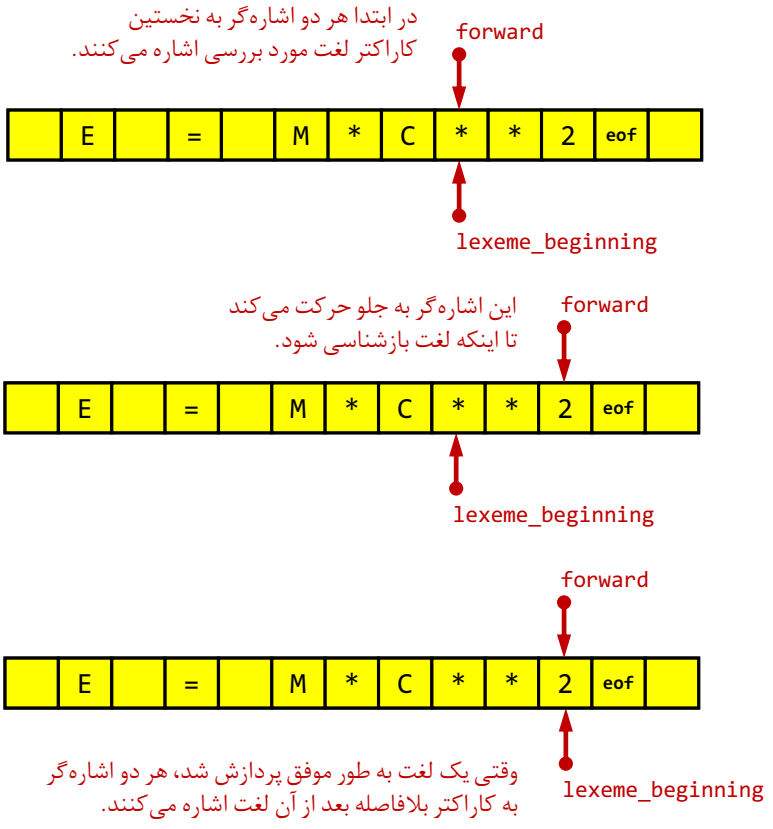
پیاده‌سازی تحلیل‌گر لغوی



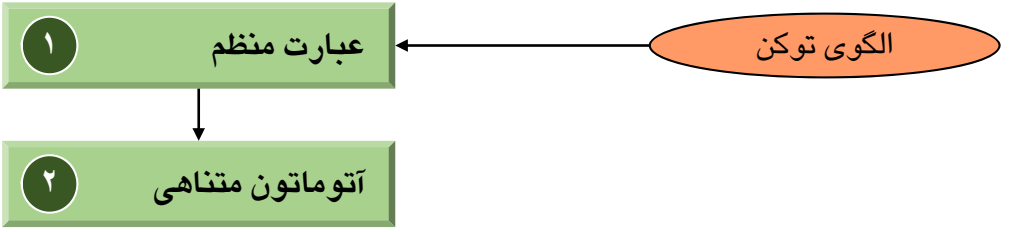
خواندن ورودی

تکنیک بافربندی

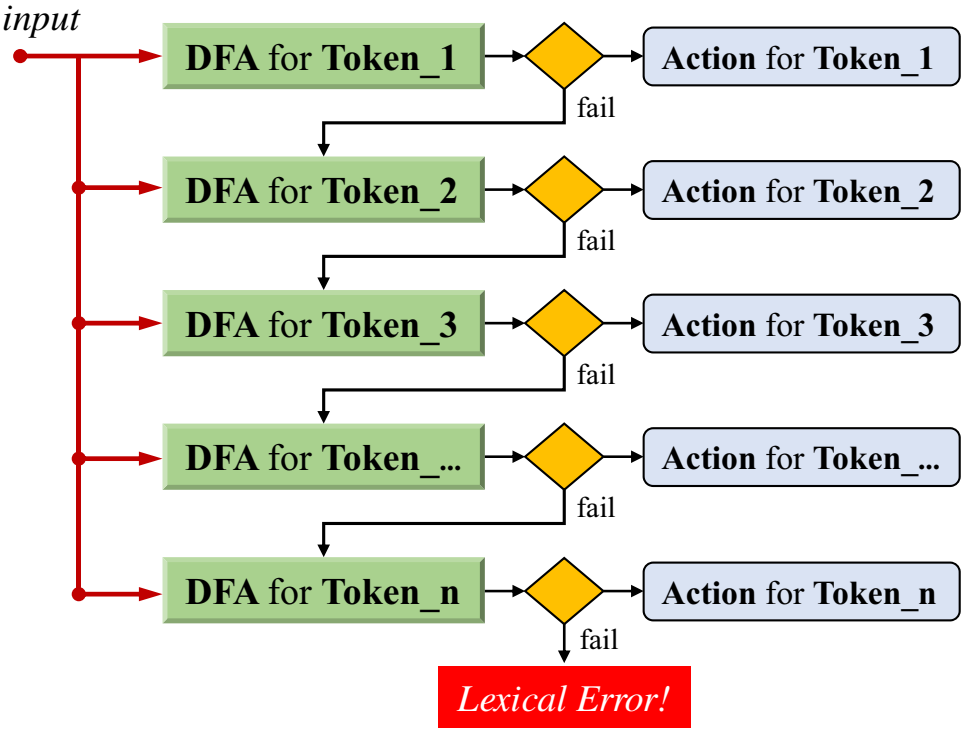
- **تحلیل لغوی** تنها مرحله از کامپایل است که باید ورودی را بخواند.
- خواندن ورودی می‌تواند زمان‌گیر باشد.
- تکنیک **بافربندی** برای بهبود کارآیی لازم است.



از عبارت منظم تا تحلیل‌گر لغوی



ساختار بازشناسی‌کننده‌ی توکن‌ها بر اساس آتوماتون متناهی



- اگر با دنبال کردن یک آتوماتون شکست خوردیم، اشاره‌گر forward را به عقب برمی‌گردانیم و آتوماتون بعدی را آزمایش می‌کنیم.
- اگر همه‌ی آتوماتون‌ها شکست خوردند، یک **خطای لغوی** آشکار می‌شود.
- در هنگام بازشناسی یک لغت، یک کنش (action) می‌تواند اجرا شود.
  مثلاً درج رکورد شناسه در جدول نمادها و ...

انواع ابهام در تحلیل لغوی

تطابق
یک لغت با
چند عبارت منظم

تطابق
بخشی از یک لغت با
یک عبارت منظم

برخورد با ابهام «تطابق یک لغت باچند عبارت منظم»

$$R_1 \rightarrow abb$$
$$id \rightarrow letter(letter \mid digit)^*$$

مثال: لغت abb هم با عبارت منظم R_1 و هم با عبارت منظم id تطابق می‌یابد.

قاعده

اگر یک لغت باچند عبارت منظم تطابق یافت،
اولویت با عبارت منظمی است که در لیست زودتر دیده می‌شود.

* پیاده‌سازی با ترتیب آتوماتون‌ها

برخورد با ابهام «تطابق بخشی از یک لغت با یک عبارت منظم»

رشته‌ی position
و عبارت منظم

$$id \rightarrow letter(letter \mid digit)^*$$

لغت‌های زیر همگی با id تطابق می‌یابند:

p
po
pos
...
position

قاعده

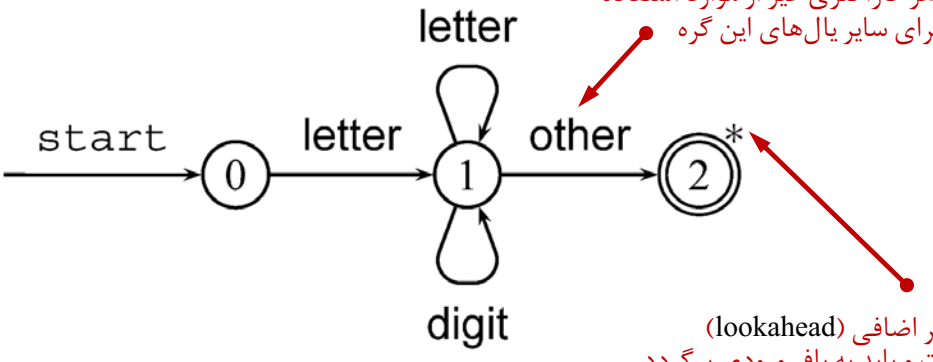
استراتژی حداکثر طول
لغت مربوط به هر توکن، آن است که بیشترین طول ممکن را دارد.

* پیاده‌سازی با دو روش: (۱) استفاده از lookahead و (۲) تغییر پاسخ آتوماتون به عدم پذیرش

برخورد با ابهام «تطابق بخشی از یک لغت با یک عبارت منظم»

استراتژی حداکثر طول: پیاده‌سازی اول: استفاده از آتوماتون با lookahead

استفاده از آتوماتون با lookahead

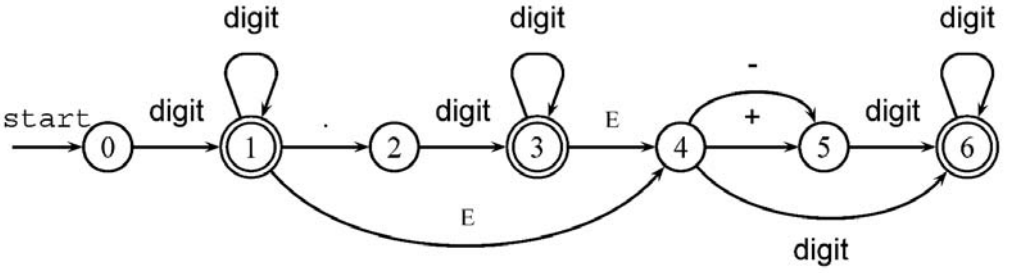


برخورد با ابهام «تطابق بخشی از یک لغت با یک عبارت منظم»

تغییر پاسخ آتوماتون به عدم پذیرش

با رسیدن به حالت پذیرش، توقف نمی‌کنیم و همچنان ادامه می‌دهیم.
هر گاه به شکست برخوردیم، به **آخرین حالت پذیرش** برمی‌گردیم.

مثال: آتوماتون اعداد (61.23Express)

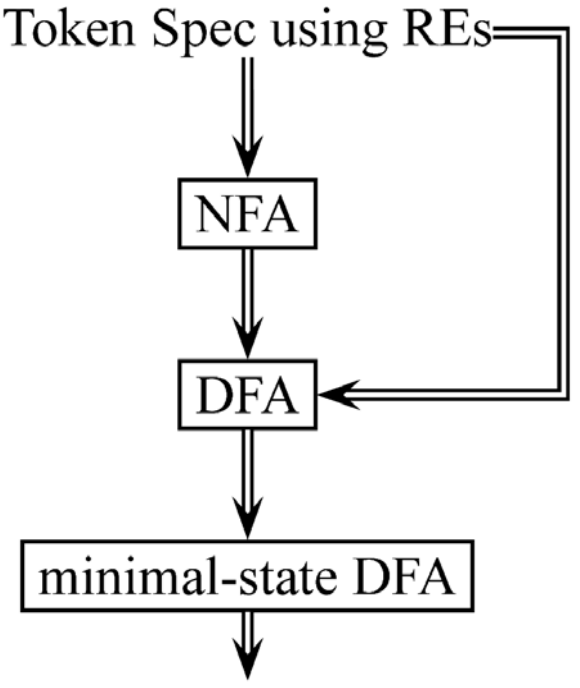


یک ملاحظه برای بهبود کارآیی

در پیاده‌سازی، ترتیب آتوماتون‌ها اهمیت دارد
(آتوماتون‌ها یکی پس از دیگری در یک دنباله آزمایش می‌شوند):

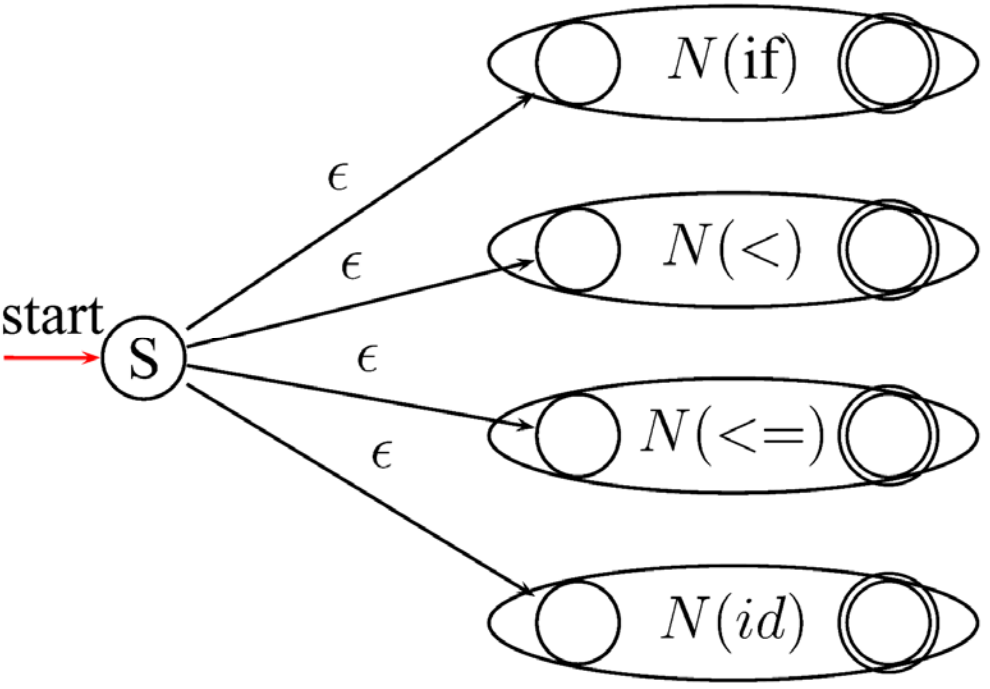
اول آتوماتون‌هایی را قرار می‌دهیم که
متناظر با پر رخدادترین توکن‌ها هستند
(مثل فواصل خالی)

فرآیند ساخت بازشناسی‌کننده‌ی هر توکن



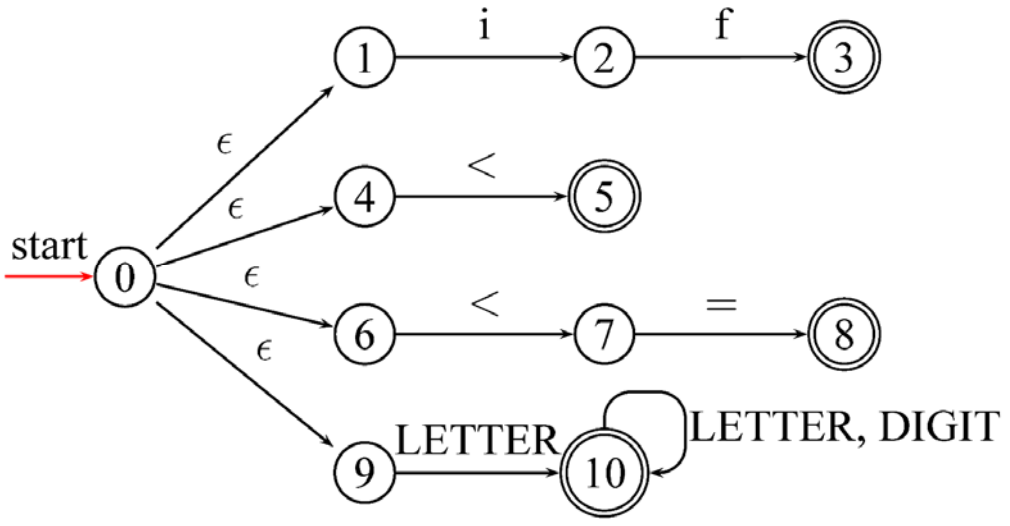
مثال: ساخت بازشناسی‌کننده برای چهار توکن

NFA



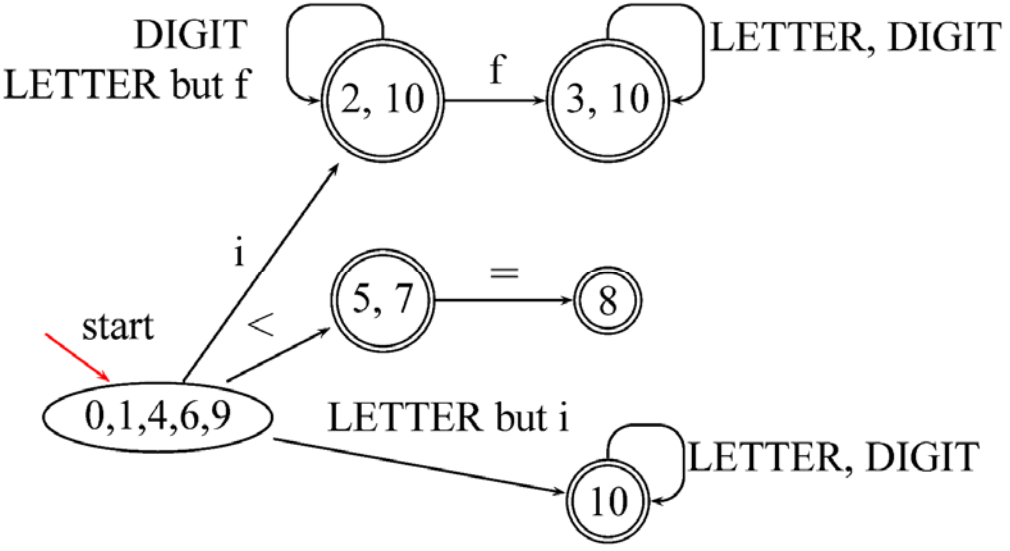
مثال: ساخت بازشناسی‌کننده برای چهار توکن

تکمیل NFA



مثال: ساخت بازشناسی‌کننده برای چهار توکن

تبدیل NFA به DFA و می‌نیمم‌سازی DFA



مسائل خاص برای تحلیل‌گر لغوی

لزوم خواندن کاراکترهای اضافی از ورودی

تحلیل‌گر لغوی برای بازشناسی یک توکن
گاهی مجبور می‌شود که چند کاراکتر اضافی را نیز از ورودی بخواند.

کاراکترهای اضافی خوانده شده، باید پس از بازشناسی توکن به بافر ورودی برگردانده شوند.

مثال:

- رزروشده نبودن کلیدواژه‌ها در برخی زبان‌ها
- بی‌اهمیت بودن فاصله‌های خالی در برخی زبان‌ها
- پردازش ثابت‌های رشته‌ای
- محدودیت در طول شناسه‌ها (بستار متناهی)

مسائل خاص برای تحلیل‌گر لغوی

لزوم خواندن کاراکترهای اضافی از ورودی: (۱) رزروشده نبودن **کلیدواژه‌ها** در برخی زبان‌ها

رزروشده نبودن کلیدواژه‌ها در برخی زبان‌ها

مانند زبان PL/1

مثال:

```
if then then then = else;
else else = then;
```

مسائل خاص برای تحلیل‌گر لغوی

لزوم خواندن کاراکترهای اضافی از ورودی: (۲) بی‌اهمیت بودن فاصله‌های خالی در برخی زبان‌ها

بی‌اهمیت بودن فاصله‌های خالی در برخی زبان‌ها

مانند زبان FORTRAN و Algol68

فاصله‌های خالی نادیده گرفته می‌شوند.

مثال:

```
do 10 i = 1,25
```
← حلقه با شمارنده‌ی `i`

```
do 10 i = 1.25
```
← انتساب به متغیر `do10i`

مسائل خاص برای تحلیل‌گر لغوی

لزوم خواندن کاراکترهای اضافی از ورودی: (۳) پردازش ثابت‌های رشته‌ای

پردازش ثابت‌های رشته‌ای

کاراکترهای خاص در ثابت‌های رشته‌ای باید به گونه‌ی دیگری تفسیر شوند.

مثال:

کاراکتر خط جدید \n
کاراکتر تب \t
نمادهای نقل قول " "
جداکننده‌ی توضیحات /* */ (comment delimiter)

مسائل خاص برای تحلیل‌گر لغوی

لزوم خواندن کاراکترهای اضافی از ورودی: (۴) محدودیت در طول شناسه‌ها **(بستار متناهی)**

محدودیت در طول شناسه‌ها (بستار متناهی)

در برخی زبان‌ها، طول شناسه‌ها محدودیت دارد:

مثلاً FORTRAN 66 (حداکثر ۶ کاراکتر برای هر شناسه)

خطای لغوی

محل رخداد خطا	نام خطا	مثال
تحلیل لغوی	**خطای لغوی (lexical error)**	شروع کردن یک شناسه با عدد
تحلیل نحوی	**خطای نحوی (syntax error)**	عدم تطابق پرانترها در یک عبارت ریاضی
تحلیل معنایی	**خطای معنایی (semantic error)**	انتساب یک مقدار اعشاری به متغیر صحیح
تولید کد میانی		کمبود حافظه بر روی دیسک
بهینه‌سازی کد میانی		کمبود حافظه بر روی دیسک
تولید کد نهایی		کمبود حافظه بر روی دیسک

خطای لغوی: خطایی که در سطح تحلیل لغوی شناسایی می‌شود.

* هرگاه یک لغت با الگوی هیچ توکنی تطبیق پیدا نکند، خطای لغوی رخ داده است.
* تعداد کمی از خطاها در سطح تحلیل لغوی قابل تشخیص است (به دلیل دید محلی).

استراتژی‌های عبور از خطاهای لغوی

استراتژی حالت وحشت (Panic Mode)	از ورودی مابقی کاراکترها را یکی یکی حذف می‌کنیم تا زمانی که تحلیل‌گر قادر به یافتن یک توکن جدید شود.
استراتژی‌های تک‌کاراکتری	• حذف یک کاراکتر بیگانه • درج یک کاراکتر جا افتاده • جایگزینی یک کاراکتر غلط با کاراکتر صحیح • تغییر مکان دو کاراکتر همسایه تبدیل `=$:` به `:=` تبدیل `:` به `:=` تبدیل `::` به `:=` تبدیل `fi` به `if`
استراتژی کم‌ترین فاصله	محاسبه‌ی حداقل تعداد تغییر برای تبدیل برنامه‌ی خطا به یک برنامه‌ی صحیح * صرفاً برای داشتن یک معیار نظری است و به دلیل هزینه‌ی بالا پیاده‌سازی نمی‌شود.

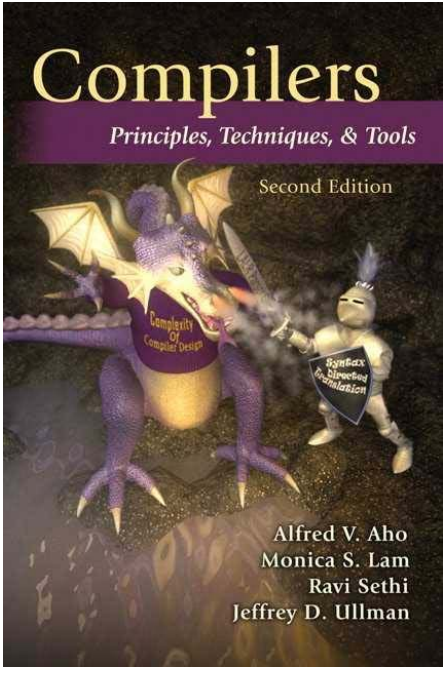
اصول طراحی کامپایلر

تحلیل لغوی (۳)

۲

منابع

منبع اصلی



A. V. Aho, M. S. Lam, R. Sethi, J. D. Ullman,
Compilers: Principles, Techniques and Tools,
Second Edition, Addison-Wesley, 2007.

Chapter 3