

کامپایلر  
پویشگر  
مقدمات و خودکاره متنهایی (معین) و عبارت منظم

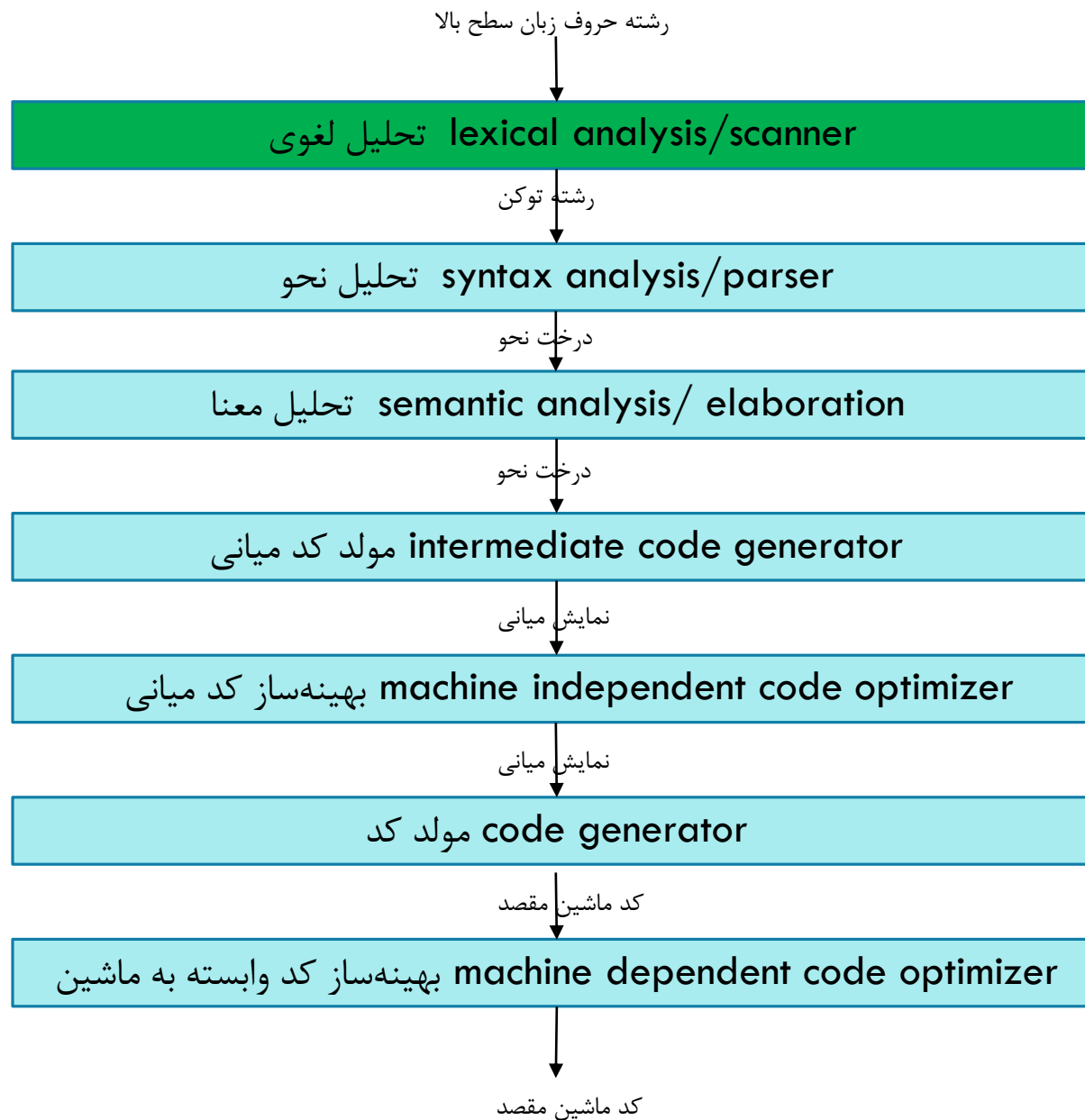
محسن هوشمند  
دانشکده تکنولوژی اطلاعات و علم رایانه  
دانشگاه تحصیلات تکمیلی علوم پایه زنجان

**Lex.i.cal:** of or relating to words or the vocabulary of a language as distinguished from its grammar and construction. *Webster's Dictionary*

**مفرد:** (زبان‌شناسی) در دستور زبان، ویژگی اسمی که تنها بر یک مصداق دلالت کند.

فرهنگ فارسی عمید

تک‌واژه



# شمارش نویسه‌ها

```
#include <stdio.h>
main()
{
    int num;
    num = 0;
    while (getchar() != EOF)
        ++num; /* add 1 to num */
    printf("%d\n", num);
}
```

Fig. 1.10. C program to count number of input characters.

# شمارش کلمات

```
#include <stdio.h>

#define IN  1 /* inside a word */
#define OUT 0 /* outside a word */

/* count lines, words, and characters in input */
main()
{
    int c, nl, nw, nc, state;

    state = OUT;
    nl = nw = nc = 0;
    while ((c = getchar()) != EOF) {
        ++nc;
        if (c == '\n')
            ++nl;
        if (c == ' ' || c == '\n' || c == '\t')
            state = OUT;
        else if (state == OUT) {
            state = IN;
            ++nw;
        }
    }
    printf("%d %d %d\n", nl, nw, nc);
}
```

# مرور پویشگر

تبدیل رشته‌ای از حروف به رشته‌ای از لغت‌ها

دسته‌بندی نحوی هر لغت

تولید دنباله‌ای از تکه‌ها (توکن‌ها)

▪ تبدیل لغت به تکه

تنها مرحله در تماس با حروف ورودی برنامه از مراحل کامپایل‌گر

استفاده از «عبارت منظم»

▪ توصیف کلمات معتبر در زبان برنامه‌نویسی

روشی ریاضی جهت تولید پویشگر از عبارت منظم

▪ یا دستی و یا خودکار

# مرور پویشگر

معمولا در تعامل با جدول علامت

▪ مثال - هنگام کار با شناسه‌ها: افزودن لغت به جدول علامت

[اژدرها] معمولا با درخواست تجزیه‌گر از تحلیل‌گر لغوی

▪ موجب به کار افتادن لغت‌یاب

▪ خواندن نویسه‌ها از ورودی تا یافتن لغت بعدی و ایجاد تکه مربوطه به آن

حذف توضیحات و فواصل

شناسایی و وصل کردن پیام خطای یافت شده با موقعیت متناظر در کد مبدا

گسترش و اتصال ماکروها به کد

# مثال

چند تکه دارد؟

```
int max(x, y)
{
    /* find max between x and y*/
    int x , y;
    return (x>y?x:y) ;
}
```



# مقدمه

اولین مرحله از فرآیند سه مرحله‌ای درک ورودی

ورودی: رشته‌ای از نویسه‌ها

▪ تنها مرحله که با نویسه‌های برنامه سروکار دارد

خروجی: رشته‌ای از کلمات

گروه‌بندی نویسه‌ها با یکدیگر

▪ تشکیل کلمات

▪ ساده و نسبتاً سریع

استفاده از قوانینی جهت تعیین اعتبار کلمه در زبان مبدا

▪ توصیف ریاضی از نحو لغوی زبان و تولید تشخیص‌گر سریع

▪ گاهی اوقات دست

▪ سریع و قوی

# مقدمه - ا/د/امه

در صورت اعتبار هر کلمه ، تخصیص رده نحوی بدان

- مقوله نحوی syntactic category

- مقوله‌بندی کلمات منطبق بر استفاده دستوری آنها

تشخیص‌گر recognizer برنامه‌ای جهت استخراج کلمات خاص از رشته‌ای از نویسه‌ها

کتاب‌ها و درس‌ها تبلیغ روش خودکار

- اغلب کامپیولرهای تجاری و متن-باز

# مقدمه - ادامه

قوانینی تحت عنوان ریزنحو `microsyntax`

- یافتن کلمات
- جداکردن کلمات از یکدیگر
- در زبان فارسی

# مقدمه - ادامه

قوانینی تحت عنوان ریزنحو `microsyntax`

- یافتن کلمات
- جداکردن کلمات از یکدیگر
- در زبان فارسی
- فاصله
- استفاده از لغتنامه جهت تعیین اعتبار کلمه

# مقدمه - ادامه

قوانینی تحت عنوان ریزنحو microsyntax

- یافتن کلمات
- جداکردن کلمات از یکدیگر
- در زبان فارسی
  - فاصله
  - استفاده از لغتنامه جهت تعیین اعتبار کلمه
- در زبان برنامه‌نویسی

# مقدمه - ادامه

قوانینی تحت عنوان ریزنحو *microsyntax*

- یافتن کلمات
- جداکردن کلمات از یکدیگر
- در زبان فارسی
  - فاصله
- استفاده از لغتنامه جهت تعیین اعتبار کلمه
- در زبان برنامه‌نویسی
  - فاصله‌ها
  - نشانه‌گذاری (نشانه‌های نگارشی)

# مقدمه - ادامه

قوانینی تحت عنوان ریزنحو *microsyntax*

- یافتن کلمات
- جداکردن کلمات از یکدیگر
- در زبان فارسی
  - فاصله
  - استفاده از لغتنامه جهت تعیین اعتبار کلمه
- در زبان برنامه‌نویسی
  - فاصله‌ها
  - نشانه‌گذاری (نشانه‌های نگارشی)
- مثال-الگول
  - شناسه: تک‌حرف به همراه یک یا چند نویسه عددی و الفبائی *fee* یا *f1e* معتبر ولی *12fee* نامعتبر
  - تفاوت با زبان گفتاری: تعیین اعتبار کلمه با قوانین به جای گردش در لغتنامه

# مقدمه - ادامه

کلیدواژه **keywords** یا کلمات رزرو **reserved words**

- کلماتی که برای هدف نحوی خاصی لحاظ می‌شوند
- به عنوان شناسه بکار نمی‌روند
- مثال **while** و **static** در سی و جاوا
- کلیدواژه و نگارواژه‌ها (**punctuation marks**) تشکیل رده نحوی مخصوص به خود
- تشخیص کلیدواژه
  - یا جستجو در لغت نامه
  - یا جای‌گذاری در قوانین ریزنحو

پویشگر کارا بهره از ساختار ساده واژوی

- تدوین ویژگی‌ها
- یا موردپذیرش مولد پویشگر یا تولید دستی
- $O(1)$  زمان برای هر نویسه



# تکه و الگو و واژه

تکه: جفتی شامل نام و مقدار (اختیاری)

▪ نام: نشان دهنده نوع لغت

▪ مثال - کلیدواژه خاص، دنباله‌ای از نویسه‌های ورودی مشخص کننده شناسه

الگو: توصیفی از شکلی که لغت‌های هر تکه می‌توانند بگیرند.

▪ مثال - کلیدواژه: خود دنباله نویسه‌های شکل دهنده کلیدواژه!

▪ مثال - شناسه‌ها: ساختاری که بسیاری از رشته‌ها بر اساس آن منطبق می‌شوند - به چه معنی؟

لغت: دنباله‌ای از نویسه‌ها در برنامه مبدا که با الگویی از یک تکه منطبق می‌شود و لغت کاو [؟] آن را نمونه‌ای از توکن مذکور می‌شناساند.

# مثال کار لغت کاو

لغت‌ها

```
while  
(  
count  
<=  
100  
)  
{  
count  
++  
;  
...
```

کد سی

```
while (count <= 100) { /** some loop */  
count++;  
// Body of while continues  
...
```

تکه‌سازی

# نمونه‌ای از تکه‌ها

نمونه لغت‌ها	توصیف غیرریاضی	تکه
if	نویسه‌های i و f	if
else	نویسه‌های e و a و s و e	else
<=, !=	< یا > یا <= یا >= یا == یا !=	comparison
Pi, score, d2, printf	حرفی که با دنباله حروف یا ارقام	id
3.14159, 0, 6.02e23	هر مقدار عددی	number
"AzhdarhaDragon!!"	هر چیزی محصور بین دو علامت ""	literal

# رده‌های معمول تکه‌ها

- ۱- یک تکه برای هر کلیدواژه- الگوی کلیدواژه خود کلیدواژه است.
- ۲- تکه‌های عملگرها- یا هر یک منفرد و متمایز یا همگی در یک رده
- ۳- یک تکه نمایشگر تمامی شناسه‌ها
- ۴- یک یا چند تکه برای نمایش ثابت‌ها چون لیترال و اعداد
- ۵- تکه‌هایی برای نشانه نگارشی- چون ویرگول و نقطه‌ویرگول و پرانتز راست و پرانتز چپ

**Lexical elements:** The Jack language includes five types of terminal elements (tokens):

keyword: 'class' | 'constructor' | 'function' |  
'method' | 'field' | 'static' | 'var' |  
'int' | 'char' | 'boolean' | 'void' | 'true' |  
'false' | 'null' | 'this' | 'let' | 'do' |  
'if' | 'else' | 'while' | 'return'

symbol: '{' | '}' | '(' | ')' | '[' | ']' | '.' |  
' ,' | ';' | '+' | '-' | '\*' | '/' | '&' |  
'|' | '<' | '>' | '=' | '~'

integerConstant: A decimal number in the range 0 .. 32767.

StringConstant: "" A sequence of Unicode characters not including double quote or  
newline ""

identifier: A sequence of letters, digits, and underscore ('\_') not starting with a  
digit.

# تکه

زبان سی دارای بیش از صد نوع تکه

۴۴ کلیدواژه چون `double` و `if` و `return` و `struct` و جز اینها

شناسه‌ها

عدد صحیح

عدد ممیزدار

رشته

نشان‌ها

دو نوع توضیح

# صفت تکه

بیش از یک لغت منطبق با الگو

▪ نیاز با افزودن اطلاع بیشتر جهت استفاده در مراحل بعدی

مثال- الگوی «عدد» هر دو عدد 0 و 1 را قبول می‌کند.

▪ مولد کد نیاز به دانستن اینکه چه لغتی در کد مبدا مورد استفاده است

لغت کاو در بسیاری از موارد هم برگرداننده نام تکه و هم مقدار (صفت) تکه

▪ نام تکه مورد استفاده در تصمیمات تجزیه‌گر

▪ صفت تکه مورد استفاده در ترجمه تکه پس از تجزیه

بنیاد بر این فرض که هر تکه حداکثر یک صفت

▪ صفت می‌تواند مرکب باشد

▪ مثال- شناسه- نگهداری لغت، نوع، موقعیت آن در متن کد مبدا (جهت صدور خطا در وقت مقتضی) در جدول علامت

## صفت تکه - ادامه

$E = M * C ** 2$

مثال فرترن

<id, pointer to symbol-table entry for E>  
<assign\_op>  
<id, pointer to symbol-table entry for M>  
<mult\_op>  
<id, pointer to symbol-table entry for C>  
<exp\_op>  
<number, integer value 2>



# تشخیص کلمات

ساده‌ترین الگوریتم تشخیص کلمات

▪ تدوین نویسه به نویسه

مثال - تشخیص کلیدواژه **new**

روال **NextChar** (نویسه‌بعد) برگرداندن نویسه بعدی

▪ بررسی نویسه به نویسه

```
c ← NextChar();
if (c = 'n')
  then begin;
    c ← NextChar();
    if (c = 'e')
      then begin;
        c ← NextChar();
        if (c = 'w')
          then report success;
          else try something else;
        end;
      else try something else;
    end;
  else try something else;
```

[کوپر]

# تشخیص کلمات

ساده‌ترین الگوریتم تشخیص کلمات

▪ تدوین نویسه به نویسه

مثال - تشخیص کلیدواژه **new**

روال **NextChar** (نویسه‌بعد) برگرداندن نویسه بعدی

▪ بررسی نویسه به نویسه

امکان نمایش با نمودار انتقال

```
c ← NextChar();
if (c = 'n')
  then begin;
    c ← NextChar();
    if (c = 'e')
      then begin;
        c ← NextChar();
        if (c = 'w')
          then report success;
          else try something else;
        end;
      else try something else;
    end;
  else try something else;
end;
```

[کوپر]

# تشخیص کلمات

ساده‌ترین الگوریتم تشخیص کلمات

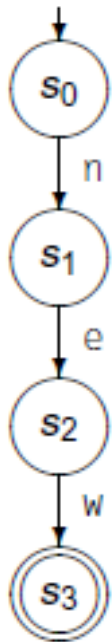
▪ تدوین نویسه به نویسه

مثال - تشخیص کلیدواژه **new**

روال **NextChar** (نویسه بعد) برگرداندن نویسه بعدی

▪ بررسی نویسه به نویسه

امکان نمایش با نمودار انتقال

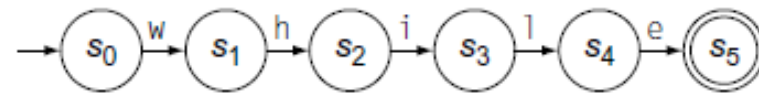


```
c ← NextChar();
if (c = 'n')
  then begin;
    c ← NextChar();
    if (c = 'e')
      then begin;
        c ← NextChar();
        if (c = 'w')
          then report success;
          else try something else;
        end;
      else try something else;
    end;
  else try something else;
end;
```

[کوپر]

# تشخیص کلمات - ادامه

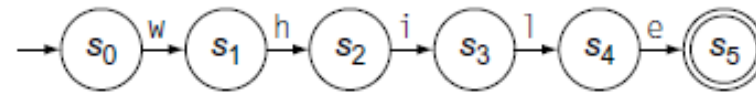
مثال - تشخیص کلیدواژه while



دارای پنج ساختار «اگر-آن گاه-وگرنه»

# تشخیص کلمات - ادامه

مثال - تشخیص کلیدواژه while



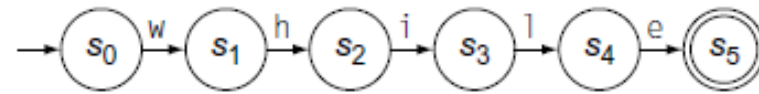
دارای پنج ساختار «اگر-آن گاه-وگرنه»

جهت تشخیص چند کلمه

▪ اتصال چند یال خروجی به حالتها

# تشخیص کلمات - ادامه

مثال - تشخیص کلیدواژه while



دارای پنج ساختار «اگر-آن گاه-وگرنه»

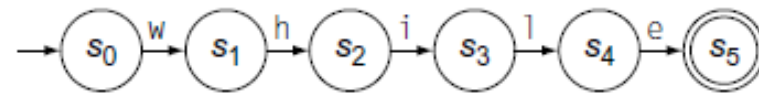
جهت تشخیص چند کلمه

▪ اتصال چند یال خروجی به حالتها

مثال - تشخیص not و new

# تشخیص کلمات - ادامه

مثال - تشخیص کلیدواژه while

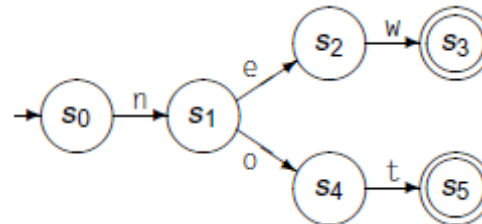


دارای پنج ساختار «اگر-آن گاه-وگرنه»

جهت تشخیص چند کلمه

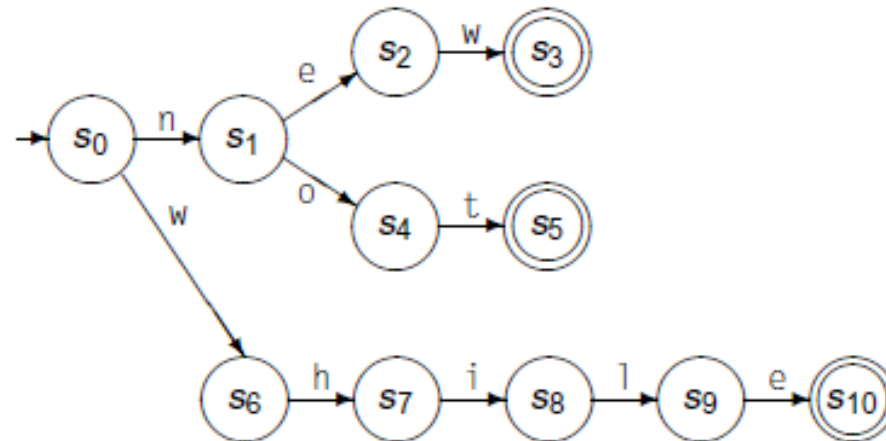
▪ اتصال چند یال خروجی به حالتها

مثال - تشخیص new و not



# تشخیص کلمات - ادامه

ترکیب چند کلیدواژه





# منابع

[اژدرها]

[ببر سبز]

[کوپر]

[فیشر]

[نیسان]