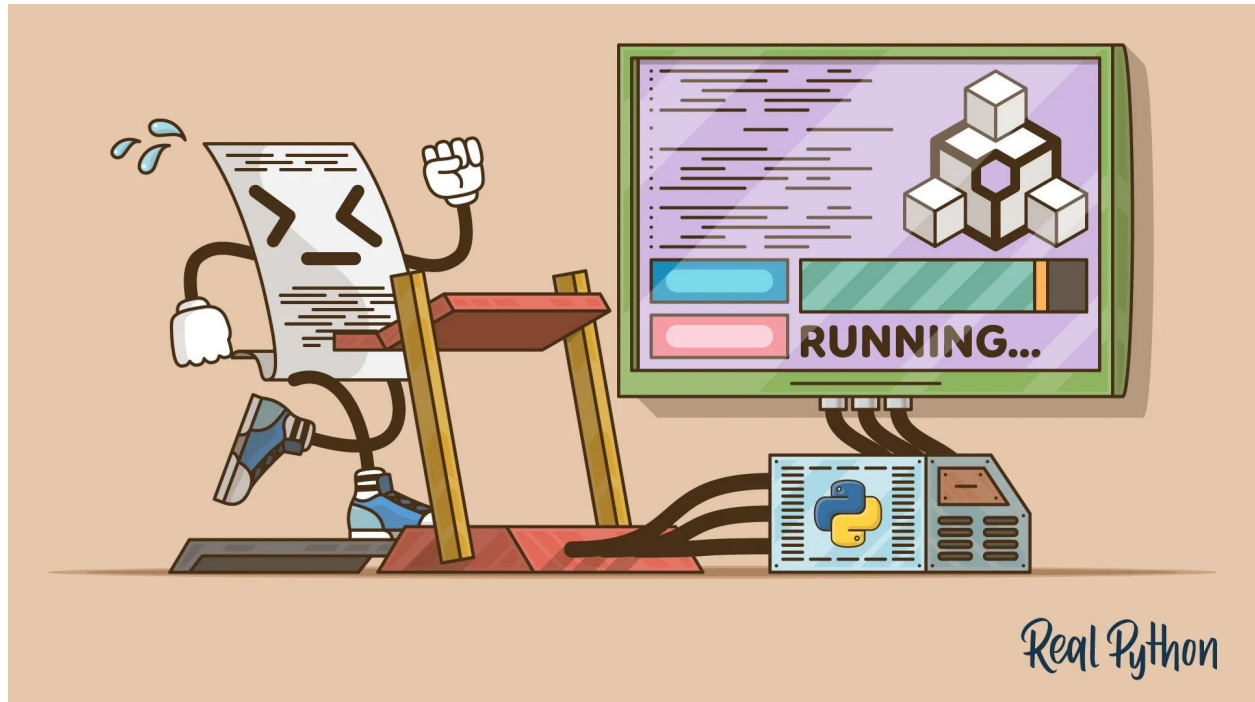


Tugas Besar IF2124 Teori Bahasa Formal dan Otomata

Compiler Bahasa Python

Tanggal Rilis : Senin, 8 November 2021 pukul 13.00 WIB

Tanggal Pengumpulan : Rabu, 24 November 2021 pukul 23.59 WIB



Sumber: <https://realpython.com/>

Deskripsi Permasalahan

Python adalah bahasa *interpreter* tingkat tinggi (*high-level*), dan juga *general-purpose*. Python diciptakan oleh Guido van Rossum dan dirilis pertama kali pada tahun 1991. Filosofi desain pemrograman Python mengutamakan *code readability* dengan penggunaan *whitespace*-nya. Python adalah bahasa multiparadigma karena mengimplementasi paradigma fungsional, imperatif, berorientasi objek, dan reflektif.

Dalam proses pembuatan program dari sebuah bahasa menjadi instruksi yang dapat dieksekusi oleh mesin, terdapat pemeriksaan sintaks atau kompilasi bahasa yang dibuat oleh programmer. Kompilasi ini bertujuan untuk memastikan instruksi yang dibuat oleh programmer mengikuti aturan yang sudah ditentukan oleh bahasa tersebut. Baik bahasa berjenis *interpreter*

maupun *compiler*, keduanya pasti melakukan pemeriksaan sintaks. Perbedaannya terletak pada apa yang dilakukan setelah proses pemeriksaan (kompilasi/*compile*) tersebut selesai dilakukan.

Dibutuhkan *grammar* bahasa dan algoritma *parser* untuk melakukan kompilasi. Sudah sangat banyak *grammar* dan algoritma yang dikembangkan untuk menghasilkan *compiler* dengan performa yang tinggi. Terdapat CFG, CNF^e, CNF^{+e}, 2NF, 2LF, dll untuk *grammar* yang dapat digunakan, dan terdapat LL(0), LL(1), CYK, Earley's Algorithm, LALR, GLR, Shift-reduce, SLR, LR(1), dll untuk algoritma yang dapat digunakan untuk melakukan *parsing*.

Pada tugas besar ini, implementasikanlah ***compiler* untuk Python** untuk *statement-statement* dan sintaks-sintaks bawaan Python. Gunakanlah konsep **CFG** untuk pengerjaan **compiler yang mengevaluasi syntax program**. Untuk **nama variabel** dalam program, gunakanlah **FA**.

Algoritma yang dipakai dibebaskan, namun tim asisten menyarankan menggunakan algoritma **CYK (Cocke-Younger-Kasami)**. Algoritma CYK harus menggunakan *grammar* CNF (Chomsky Normal Form) sebagai *grammar* masukannya. Oleh karena itu, jika ingin menggunakan CYK buatlah **terlebih dahulu *grammar* dalam CFG (Context Free Grammar), kemudian konversikan *grammar* CFG tersebut ke *grammar* CNF**.

Berikut adalah daftar kata kunci bawaan Python yang harus terdaftar dalam *grammar* (yang dicoret tidak perlu diimplementasi). Rincian mengenai implementasi dan contohnya dapat dilihat pada [pranala ini](#).

False	class	finally	is	return
None	continue	for	lambda	try
True	def	from	nonlocal	while
and	del	global	not	with
as	elif	if	or	yield
assert	else	import	pass	
break	except	in	raise	

Hal-hal ini tidak perlu kalian masukkan ke dalam *grammar* atau diimplementasikan:

1. Arti semantik dari *input* (mis. walaupun kelas Foo belum pernah didefinisikan, `bar = Foo()` atau `bar.a_method()` diperbolehkan)
2. Arti semantik dari *method* (mis. jumlah parameternya)
3. Regex dalam bentuk apapun, seperti r-string (mis. `r'123'`)
4. F-string (mis. `f'aku suka tbfo'`)
5. *End of statement* dengan menggunakan titik koma (;)
6. *Syntactic sugar*
7. Decorator (@), dan *backslash* (\)
8. Karakter-karakter di luar cakupan ASCII.
9. Tab(INDENT/DEDENT)

Pengerjaan

Pengerjaan dilakukan secara berkelompok, setiap kelompok terdiri dari maksimal 3 mahasiswa, mahasiswa diperkenankan membentuk kelompok masing masing dan mengisi data kelompok di sheets berikut :

<https://docs.google.com/spreadsheets/d/10FTPI8DVEaKu22H90p2-twzi4ZX4vC1FAG4c4aJfvqw/edit#gid=0>

Spesifikasi Program

1. Tugas dikerjakan menggunakan bahasa pemrograman **Python**.
2. Program menggunakan Antarmuka berbasis CLI (Command Line Interface).
3. Mahasiswa akan menggunakan konsep **CFG** dan **FA** untuk membuat sebuah parser bahasa pemrograman (python).
4. Program akan **menerima suatu teks file atau string** yang merupakan kode dari sebuah program, lalu program akan memanfaatkan CFG untuk mengevaluasi kebenaran syntax dari kode yang dijadikan input tadi. Program juga menggunakan FA untuk mengecek tiap expression (contoh :, FA akan menolak '123dasda' sebagai nama variabel, karena dalam python nama variabel tidak diperbolehkan didahului angka, contoh lain : ekspresi '55+' salah karena penempatan operand + harus dijepit di antara 2 expression valid lain)
5. Alur umum dari program yang akan dibangun adalah sebagai berikut.

- a. Menerima **input berupa file eksternal** berisi string yang merupakan kode sebuah program python
 - b. Melakukan **evaluasi sintaks** dengan CFG
 - c. Melakukan **evaluasi nama-nama variabel** yang ada dengan FA
 - d. Memberikan keluaran hasil evaluasi program antara **“Accepted”** jika input diterima atau **“Syntax Error”** jika input tidak diterima
 - e. **(BONUS)** Memberi tahu letak dan detail kesalahan syntax jika ada
6. Buatlah Laporan terkait pengerjaan tugas kalian, laporan wajib memiliki :
- a. **Teori dasar** FA, CFG, serta penjelasan per-syntax-an python yang perlu diperhatikan selama pembuatan CFG dan FA.
 - b. **Hasil** FA dan CFG yang dibuat.
 - c. **Implementasi dan Pengujian**
 - i. Spesifikasi teknis program, termasuk di dalamnya struktur data, fungsi dan prosedur (header fungsi dan prosedur saja, tidak perlu source code), antarmuka, dan lain-lain yang dianggap perlu.
 - ii. *Screenshot* yang memperlihatkan contoh dari berbagai kasus yang muncul (buatlah minimal 3 kemungkinan tipe kasus uji yang mungkin) dan analisis hasilnya
 - d. Link ke **repository github**.
 - e. Pembagian **Tugas**.

Contoh Input dan Output Program

Diberikan 2 buah file .py.

1. inputAcc.py

```
inputAcc.py
def do_something(x):
    ''' This is a sample multiline comment
    '''
    if x == 0:
        return 0
    elif x + 4 == 1:
        if True:
            return 3
        else:
```

```
        return 2
    elif x == 32:
        return 4
    else:
        return "Doodoo"
```

2. inputReject.py

inputReject.py

```
def do_something(x):
    ''' This is a sample multiline comment
    '''
    x + 2 = 3
    if x == 0 + 1:
        return 0
    elif x + 4 == 1:
        else:
            return 2
    elif x == 32:
        return 4
    else:
        return "Doodoo"
```

3.

py parserprogram.py "inputAcc.py"

Accepted

4.

py parserprogram.py "inputReject.py"

Syntax Error

(Output tambahan jika mengerjakan bonus)
Terjadi kesalahan ekspresi pada line 4 : "x + 2 = 3"

Pengumpulan

- Waktu pengerjaan tugas adalah 16 Hari, batas akhir pengumpulan ialah tanggal **24 November jam 23.59 WIB**.
- Pengumpulan dilakukan dengan **github**, buatlah repository github PRIVATE untuk pengerjaan kalian, lalu set ke public **paling lambat tepat 24 jam setelah deadline**.
- Kecurangan dan plagiarisme akan mengakibatkan **sanksi ke kedua belah pihak**.
- Demo akan dilakukan setelah pengumpulan. Metode dan peraturan demo akan diberitahukan nanti.