

domol1

April 3, 2020

1 DOMOL-1

```
[1]: import pandas as pd
import string
```

```
[12]: states = pd.read_csv('domol1_states.csv', index_col=0)

for d in string.digits:
    states[d] = states['Szamjegy']

for l in string.ascii_lowercase + string.ascii_uppercase:
    states[l] = states['Betu']

states.drop(columns=['Szamjegy', 'Betu'], inplace=True)
states.rename(columns={'Szokoz': ' '}, inplace=True)

states[['Visszalepes', 'Olvas']] = states[['Visszalepes', 'Olvas']] == 'igen'

states.head()
```

```
[12]:
```

	Allapot	Visszalepes	Olvas	\$	Egyeb	{	}	(*	...	Q	\
Id										...		
1	Kezdőállapot	False	True	19	21	19	6	19	8	19	...	2
2	Azonosítóban	False	True	3	3	3	3	3	3	3	...	2
3	Azonosító vége	True	False	1	1	1	1	1	1	1	...	1
4	Számban	False	True	5	5	5	5	5	5	5	...	5
5	Szám vége	True	False	1	1	1	1	1	1	1	...	1

	R	S	T	U	V	W	X	Y	Z
Id									
1	2	2	2	2	2	2	2	2	2
2	2	2	2	2	2	2	2	2	2
3	1	1	1	1	1	1	1	1	1
4	5	5	5	5	5	5	5	5	5
5	1	1	1	1	1	1	1	1	1

[5 rows x 77 columns]

```

[52]: class Scanner():
    def __init__(self, program):
        self.state_id = 1
        self.program = program
        self.current = 0
        self.lookahead = 0
        self.tokens = []
        self.old_state = None

    @property
    def state(self):
        return states.loc[self.state_id]

    def read(self):
        self.lookahead += 1

    def update_state(self):
        self.old_state = self.state['Allapot']
        char = self.program[self.lookahead]
        self.state_id = self.state[char] if char in states.columns else self.
→state['Egyeb']

    def store(self):
        self.tokens.append((self.program[self.current:self.lookahead], self.
→old_state))
        self.current = self.lookahead

    def get_tokens(self):
        while True:
            self.update_state()
            if self.state['Allapot'] == 'Hibakezelő':
                raise ValueError("'" + self.program[self.lookahead:] + "'")

            if self.state['Allapot'] == 'Kezdőállapot' != self.old_state:
                self.store()
            elif not self.state['Visszalepes']:
                self.read()

            if self.state['Allapot'] == 'Stop':
                self.store()
                return self.tokens[:]

    def print_tokens(self):
        tokens = self.get_tokens()
        token_max_len, token_class_max_len = map(lambda x: max(map(len, x))+1,
→zip(* tokens))

```

```

    for (token, token_class) in tokens:
        print("| " + token.ljust(token_max_len) + "| " + token_class.
→ljust(token_class_max_len) + "|")

```

```
[53]: Scanner("$").get_tokens()
```

```
[53]: [('$', 'Kezdőállapot')]
```

```
[54]: Scanner("{&.&}a(* * *):={{<>}a56>=a$").print_tokens()
```

{&.&}	{} komm. vég	
a	Azonosító vége	
(* * *)	(**)komm.vég	
:=	:= token	
{{<>}}	{} komm. vég	
a56	Azonosító vége	
>=	>= token	
a	Azonosító vége	
\$	Kezdőállapot	

```
[55]: Scanner("DOMOSI1234{}123DOMOSI123:=(***)<>DOMSOI1$").print_tokens()
```

DOMOSI1234	Azonosító vége	
{}	{} komm. vég	
123	Szám vége	
DOMOSI123	Azonosító vége	
:=	:= token	
(***)	(**)komm.vég	
<>	<> token	
DOMSOI1	Azonosító vége	
\$	Kezdőállapot	

```
[ ]:
```