

An evil-genius guide to computer programming

Daniel Lemire

Montreal 

blog: <https://lemire.me>

twitter: [@lemire](https://twitter.com/lemire)

GitHub: <https://github.com/lemire/>

```
from z3 import *
#We received 500 dollars from the Quebec government :) :) We need to do the shopping!!
shoes, pullover = Ints('shoes pullover')
solve(shoes>=1, # we want to buy minimum a shoe
      pullover>=5, # we want to buy minimum 5 pullovers

      #shoes cost 100 dollars and pullover cost 30 dollars
      100 * shoes + 30 * pullover == 500)
```

```
p1 = Bool('p1')
p2 = Bool('p2')
p3 = Bool('p3')
p4 = Bool('p4')
p5 = Bool('p5')
p6 = Bool('p6')
p7 = Bool('p7')

solve(Implies(p1, p2),Implies(p2, p3), Implies(p4, Not(p6)),Implies(p7, p5), p2, p3, p5)
```

```
from z3 import *
```

```
jacques = Int("jacques")  
francoise = Int("francoise")
```

```
jackie = Int("jackie")  
john = Int("john")
```

```
serge = Int("serge")  
jane = Int("jane")
```

```
simone = Int("simone")  
yves = Int("yves")
```

```
s.add(jacques >= 1)
s.add(francoise >= 1)
s.add(jackie >= 1)
s.add(john >= 1)
s.add(serge >= 1)
s.add(jane >= 1)
s.add(simone >= 1)
s.add(yves >= 1)
```

```
s.add(jacques <= 8)
s.add(francoise <= 8)
s.add(jackie <= 8)
s.add(john <= 8)
s.add(serge <= 8)
s.add(jane <= 8)
s.add(simone <= 8)
s.add(yves <= 8)
```

```
s.add(jacques != francoise)
s.add(jacques != jackie)
s.add(jacques != john)
s.add(jacques != serge)
s.add(jacques != jane)
s.add(jacques != simone)
s.add(jacques != yves)
```

```
s.add(francoise != jackie)
s.add(francoise != john)
s.add(francoise != serge)
s.add(francoise != jane)
...
```

```
s.add(jacques - francoise != 1)
s.add(jacques - francoise != -1)
```

```
s.add(jackie - john != 1)
s.add(jackie - john != -1)
```

```
s.add(serge - jane != 1)
s.add(serge - jane != -1)
```

```
s.add(simone - yves != 1)
s.add(simone - yves != -1)
```

```
if(s.check() == z3.sat):  
    m = s.model()  
    print(m)
```


Week 3

Conquer the Web

Managing error conditions

- Return an error code from the function
- 'raise an exception'

```
try:  
    something  
except some error:  
    do something  
finally:  
    do that always
```

```
file = open('file_path', 'w')  
try:  
    file.write('hello world')  
finally:  
    file.close()
```

```
try:  
    something  
except:  
    do something
```

```
with open('file_path', 'w') as file:  
    file.write('hello world !')
```

```
def get_if_exist(data, key):  
    if key in data:  
        return data[key]  
    return None
```

String aggregation

- `"ab" + "ac"`
- `"sb" + str(1)`
- `""`

HTML

```
<html>  
<body>  
</body>  
</html>
```

- html → body

HTML

```
<html>  
<body>  
  <p> Hello World </p>  
</body>  
</html>
```

Ordered tree

HTML

```
<html>
<body>
  <p> <a href="https://google.com">Hello World</a> </p>
</body>
</html>
```

HTML

TCP/UDP

- UDP: fast, naive, data can be lost → media streaming
- TCP: connection, error checks → HTTP, web

HTTP/HTTPS

- Get
- Post
- Head, Put, Delete, Connect, Options, Trace, Patch

Most common query is GET. A single web page can be dozens of GET queries.

```
def search(keyword):  
    result = getjson("https://api.duckduckgo.com/?q="+keyword+"&format=json")  
    results = []  
    for key in result["RelatedTopics"]:  
        if "Result" in key:  
            results.append(key["Result"])  
    return results  
  
print(search("Hamburger"))
```



```
from flask import Flask
app = Flask(__name__)

@app.route('/')
def hello_world():
    return '<html><body>Hello World!</body></html>'

if __name__ == '__main__':
    app.run()
```

```
from urllib.request import urlopen
from urllib.error import HTTPError

import timeit

def grab():
    try:
        data = urlopen("http://127.0.0.1:5000").read().decode("utf-8")
        return data
    except HTTPError:
        return None

t = timeit.timeit(grab, number=1000)
```

```
<html>
  <body>
    <form action = "http://localhost:5000/uploader" method = "POST"
      enctype = "multipart/form-data">
      <input type = "file" name = "file" />
      <input type = "submit"/>
    </form>
  </body>
</html>
```

```
import exifread
def get_exif_data(image_file):
    with open(image_file, 'rb') as f:
        exif_tags = exifread.process_file(f)
    return exif_tags
```

- 'GPS GPSLatitude' → [45, 31, 2391/50]
- 'GPS GPSLatitudeRef' → N
- 'GPS GPSLongitude' → [73, 35, 5607/100]
- 'GPS GPSLongitudeRef' → W

```
import exifread
def get_exif_data(image_file):
    with open(image_file, 'rb') as f:
        exif_tags = exifread.process_file(f)
    return exif_tags
```

```
# credit : https://pythonbasics.org/flask-upload-file/
from flask import Flask, render_template, request
from werkzeug.utils import secure_filename

app = Flask(__name__)

@app.route('/upload')
def upload_file_render():
    return render_template('upload.html')
```

```
@app.route('/uploader', methods = ['GET', 'POST'])
def upload_file():
    if request.method == 'POST':
        f = request.files['file']
        sn = secure_filename(f.filename)
        f.save(sn)
        lat, long = get_exif_location(get_exif_data(sn))
        if lat is None:
            return render_template('upload.html')
        link = "https://www.openstreetmap.org/?mlat="+str(lat)+"&mlon="+str(long)+"&zoom=15"
        return "<html><body><a href=\""+link+"\">map</a></body></html>"
```



```
import threading

x = 0

def increment():
    global x
    for i in range(500000):
        x += 1

def main():
    global x
    x = 0

    t1 = threading.Thread(target=increment)
    t2 = threading.Thread(target=increment)

    t1.start()
    t2.start()

    t1.join()
    t2.join()
    print(x)

main()
```

```
def log(long,lat):  
    with sqlite3.connect("img.db") as con:  
        con.execute("CREATE TABLE IF NOT EXISTS geo (date TEXT, long NUMERIC, lat NUMERIC)")  
        dt = datetime.now()  
        con.execute("INSERT INTO geo (date,long,lat) values (\\""+str(dt)+"\\",\\""+str(long)+"\\", \\""+str(lat)+"\\") ")
```

<https://replit.com/@lemire/MonthlyBriefExponents#main.py>

Homework

Build a small Python web application

<https://github.com/lemire/talks/tree/master/2022/evil/week3>