What is “Source Code?”

Contact:
Modern Markets Initiative
1440 G St NW
Washington, DC 20005
info@modernmarkets.org
Introduction

• Computer programs are the actual instructions that express algorithms which control the behavior of a machine.

• Computers don’t understand informal human syntax so scientists have created “programming languages,” formal grammars that allow people to express ideas in a way computers can understand.

• Source code is human-readable text, written in any of thousands of programming languages, that is translated into instructions that are directly understandable by the computer.
What’s an algorithm?

- Computer scientists use algorithms to specify a series of steps that accomplish a specific task. Here’s a very simple one that determines if you’re an adult.

```
Start

"what is your birth year"
set birthYear

"what is the current year"
set currentYear

age = currentYear - birthYear

"You are" age "years old"

age >= 18

"You are an adult"

True

End

False
```
One of the most important functions is the object identification system, which is responsible for “seeing” what’s around the car, for example, a traffic cone, person, car, truck or bicycle, and using this information to predict an object’s future behavior.

The type of object is determined by considering various characteristics like size, shape, speed or the amount of heat coming from the detected object. Once an object is identified, the car can react appropriately. For example, stop, slow down, turn, etc.

Automobile companies spend huge sums of money developing this type of algo, since improving on state-of-the-art can mean the difference between a company’s success and failure.
Algo Example: Object Identification Algorithm for Self-Driving Car

Start

Collect information from cameras, heat sensors, GPS, etc.

Is object wider than 4 feet?

Yes

Is object height > 10 feet?

Yes

Object has "Harley-Davidson" logo?

Yes

Is object height > 6 feet?

Yes

Object is a car!

No

Object is a bicycle!

No

Object is a motorcycle!

No

Object has large wheels?

Yes

Object has mud flaps?

No

Object is an SUV!

No

Object is a truck!

No

Unknown object!

Yes

Object has mud flaps?

No

Object is a bicycle!

Yes

Object is a motorcycle!

No

Object is a car!

End
Algo example: Fundamental Investor

- Ben Graham’s classic book, “The Intelligent Investor,” describes a series of tests to gauge whether companies are sound investments. They include:
  - Favorable earnings/price yield relative to AAA bonds.
  - Total debt should be less than the company book value.
  - Current Ratio > 2.0.
- Investors can use a computer to scan through thousands of stocks, precisely and quickly, and buy stocks matching those criteria.
- Most investors would never reveal their criteria to regulators. Doing so would mean that their investing methodology could be duplicated.
Example Trading Algorithm for Fundamental Investor

1. Start
2. Collect company information from SEC filings, news reports, analyst reports, etc.
3. E/P yield >= 2 * AAA bond yld?
   - Yes
   - No
4. Cur P/E >= .4 * highest P/E over past 5 yrs?
   - Yes
   - No
5. Total debt < book value?
   - Yes
   - No
6. Current ratio > 2.0?
   - Yes
   - No
7. More stocks to check?
   - Yes
   - No
8. Add stock to purchase database
9. End
My First Program

• Let’s say we want to tell the computer to display the text, “Hello, World!” on the screen. Using the Python programming language, the source code would look like this:

```python
print(“Hello, World!”)
```

• A programmer would type this in, and the computer would output:

```
Hello, World
```

• Congratulations, you’ve just written your first computer program!
What About Math?

• We can easily write source code to do calculations:

```python
print(1+1)
2
print((50 - 5.0*6) / 4)
5.0
```

• And we can store numbers in buckets called variables:

```python
previous_shares_owned = 100
shares_purchased = 200
current_shares_owned = previous_shares_owned + shares_purchased
print(current_shares_owned)
300
```
def collectExternalData():
    # Insert code to collect fundamental data from external sources, store it in stockUniverse variable
    return stockUniverse

def findGrahamStocks():
    extData = collectExternalData()  # function returns a list of all company fundamental data

    # Scan each company that data is available for
    for company in extData:
        if company.getEYield() > (AAABondYield * 2):
            if company.getTotalDebt() < company.getBookValue():
                if company.getPERatio < (company.getHistoricalPERatio() * 0.4):
                    if company.getCurrentRatio() > 2:
                        buyList.insert(company.getSymbol)
    return buyList

print("Graham Stock List=", findGrahamStocks())
The most common HFT strategy is market making using computers.

Market makers are generally price-agnostic; their algos assume that the market price is the “right” one.

A really simple Market Making algo’s source code could narrow or tighten the National Best Bid/Offer spread depending on its current position.

- If long stock, improve the ask price (wants to sell).
- If short stock, improve the bid price (wants to buy).

This is how HFT Market Makers improve prices for investors!
Algo example: Automated Market Making

def setQuotes(stock_symbol, current_position, shares, nbbo_bid_price, nbbo_ask_price):
    if current_position <= MAXIMUM_POSITION_LIMIT:  # Risk mgmt: Don’t bid if we’re already max long
        if current_position < 0: my_bid_price = nbbo_bid_price + 0.01  # Improve the NBB if we’re short
        else: my_bid_price = nbbo_bid_price
        buy_order_id = submit_trade(ACTION_BUY, shares, stock_symbol, ORDER_TYPE_LIMIT, my_bid_price, EXCH_NASDAQ)
    if current_position >= MINIMUM_POSITION_LIMIT:  # Risk mgmt: Don’t offer if we’re already max short
        if current_position > 0: my_ask_price = nbbo_ask_price - 0.01  # Improve the NBO if we’re long
        else: my_ask_price = nbbo_ask_price
        sell_order_id = submit_trade(ACTION_BUY, shares, stock_symbol, ORDER_TYPE_LIMIT, my_ask_price, EXCH_NASDAQ)

def eventTradeReport():  # When a trade is made...
    if executed_trade_id == buy_order_id:
        current_position = current_position + executed_quantity
    elif executed_trade_id == sell_order_id:
        current_position = current_position - executed_quantity
    setQuotes(execute traded_symbol, current_position, 500, nbbo_bid_price, nbbo_ask_price)

def eventNBBOChanged():  # When the NBBO changes...
    setQuotes(nbbo Symbol, current_position, 500, nbbo_bid_price, nbbo_ask_price)

def eventMarketOpened():  # At 9:30:00 AM...
    setQuotes(nbbo Symbol, current_position, 500, nbbo_bid_price, nbbo_ask_price)

def eventMarketClosed():  # At 4:00:00 PM...
    setQuotes(nbbo Symbol, 0, 0, 0, 0)
Magnitude of Source Code

• Usable commercial programs run from about 50,000 lines for a smart phone app to 2 billion for all internet services provided by Google.

• However, number of lines isn’t necessarily an indicator of complexity.
  
  • In time-critical applications like self-driving cars, HFT market making and fighter jets, programmers devote resources to "optimizing" code--making it smaller and more efficient--because less code generally executes faster.

  • Also, there’s no single standard for measuring lines of code. For example, some include comment lines, some include third-party and open-source modules, etc.
I have presented incredibly simple algorithms in this presentation. The “real thing” comprises millions of lines of source code for each company.

Typical HFT algorithms are 5-10 million lines. How does that compare?

Source: InformationIsBeautiful.net
Summary

- Dangers posed by access to source code
  - If a competitor learns how a company's algorithms work, he can steal their IP through duplication and compete unfairly against the inventor.
  - If a cyber-criminal learns how they work, he could wreak havoc on the markets by causing liquidity to exit and facilitating possible crash scenarios.
- Giving regulators access to source code would not make roads (or markets) safer. What would?
  - Regulators should mandate “best practices” for code development, testing and deployment -- and levy fines against companies that don’t comply.
  - Regulators should have access to "kill switches," the ability to prevent a bad system from executing (i.e., banning an unsafe car or phone, preventing trade order entry, etc.)
What is “Source Code?”