CS/240/Project/A

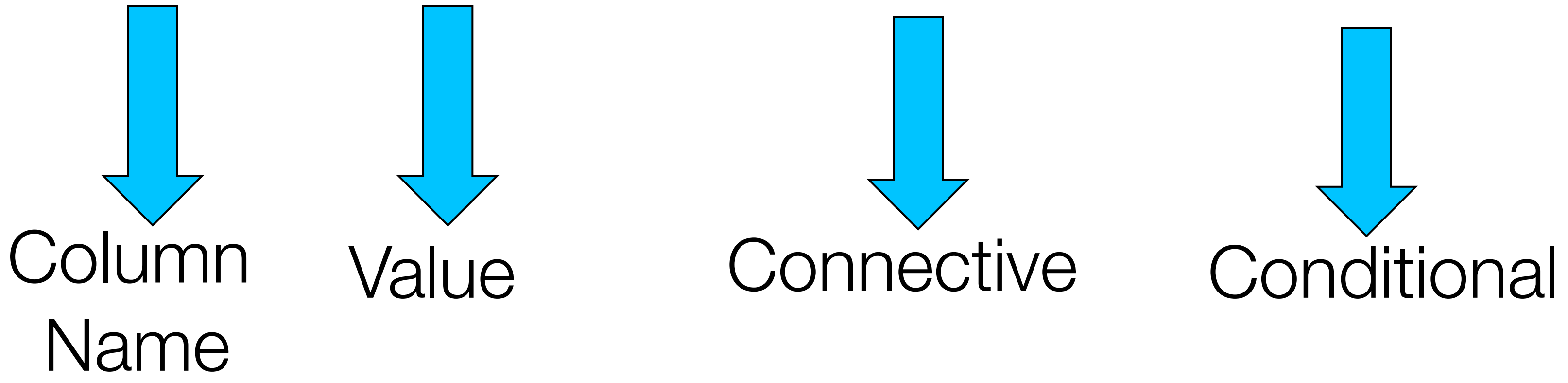
Making Data into Information

This lab will help you take a mass of raw data and start analyzing it

Additionally, it is now time to take on memory leaks:

What is a Query?

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



Data:

Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
TA	47906	Minion	.01

What is a Query?

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord

Column Name	Value	Connective	Conditional

Data:

Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
TA	47906	Minion	.01

What is a Query?

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord

Column Name | Value | Connective | Conditional

Data:

Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
TA	47906	Minion	.01

What is a Query?

Grammar:

<Query> := ^<Field>[<Connective><Field>]*$

<Field> := <Space>*<Column Name><Conditional><Value><Space>*

<Space> := ' '

<Column Name> := [a-zA-Z]+

<Conditional> := '=', '>', '<', '>=', '<='

<Value> := double OR a string

<Connective> := && or ||

Guarantees

- Column Names, Values will not violate the grammar
- The type of a column will be consistent
    - Get the type from the new tbl_type() function in the table API
    - If your column is at index 1, its type will be at index 1 in the array returned by tbl_type()

- We do NOT guarantee that connectives or conditionals will be well formed.

Problem Children

Where’s Waldo with malformed queries: spot the error!

```
AMOUNT>=50000 && ZIP=47906 & OCCUPATION=Overlord
```

Problem Children

Where’s Waldo with malformed queries: spot the error!

```
AMOUNT>=50000 && ZIP=47906 & OCCUPATION=Overlord
```

Problem Children

Where’s Waldo with malformed queries: spot the error!

```
AMOUNT>=50000 &&
```

Problem Children

Where's Waldo with malformed queries: spot the error!

```
AMOUNT>=50000 &&
```

Problem Children

Where’s Waldo with malformed queries: spot the error!

```
AMOUNT>==50000
```

Problem Children

Where’s Waldo with malformed queries: spot the error!

```
AMOUNT>==50000
```

How to make a Tree

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord

Multiple connectives:

- Count, then divide by 2.
- Connective at that index is the node to add

2 / 2 = 1 => Pick the connective at index 1 as root

How to make a Tree

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord

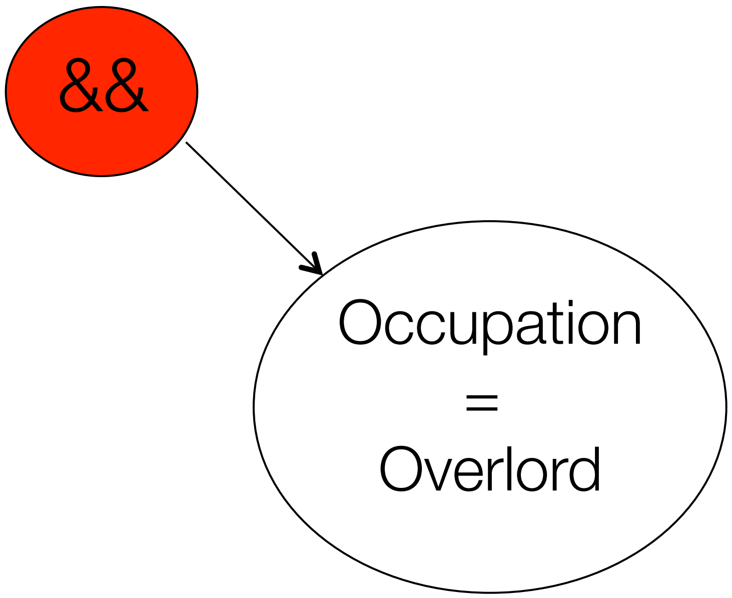
Multiple connectives:

- Count, then divide by 2.
- Connective at that index is the node to add

2 / 2 = 1 => Pick the connective at index 1 as root

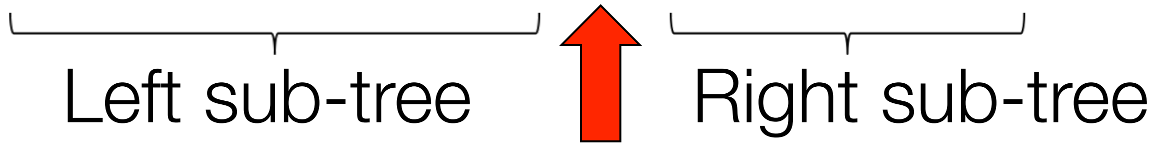
How to make a Tree

$$\underbrace{\text{AMOUNT>=50000 \&\& ZIP=47906}}_{\text{Left sub-tree}}\ \underset{\text{Root}}{\underset{\uparrow}{\text{\&\&}}}\ \underbrace{\text{OCCUPATION=Overlord}}_{\text{Right sub-tree}}$$

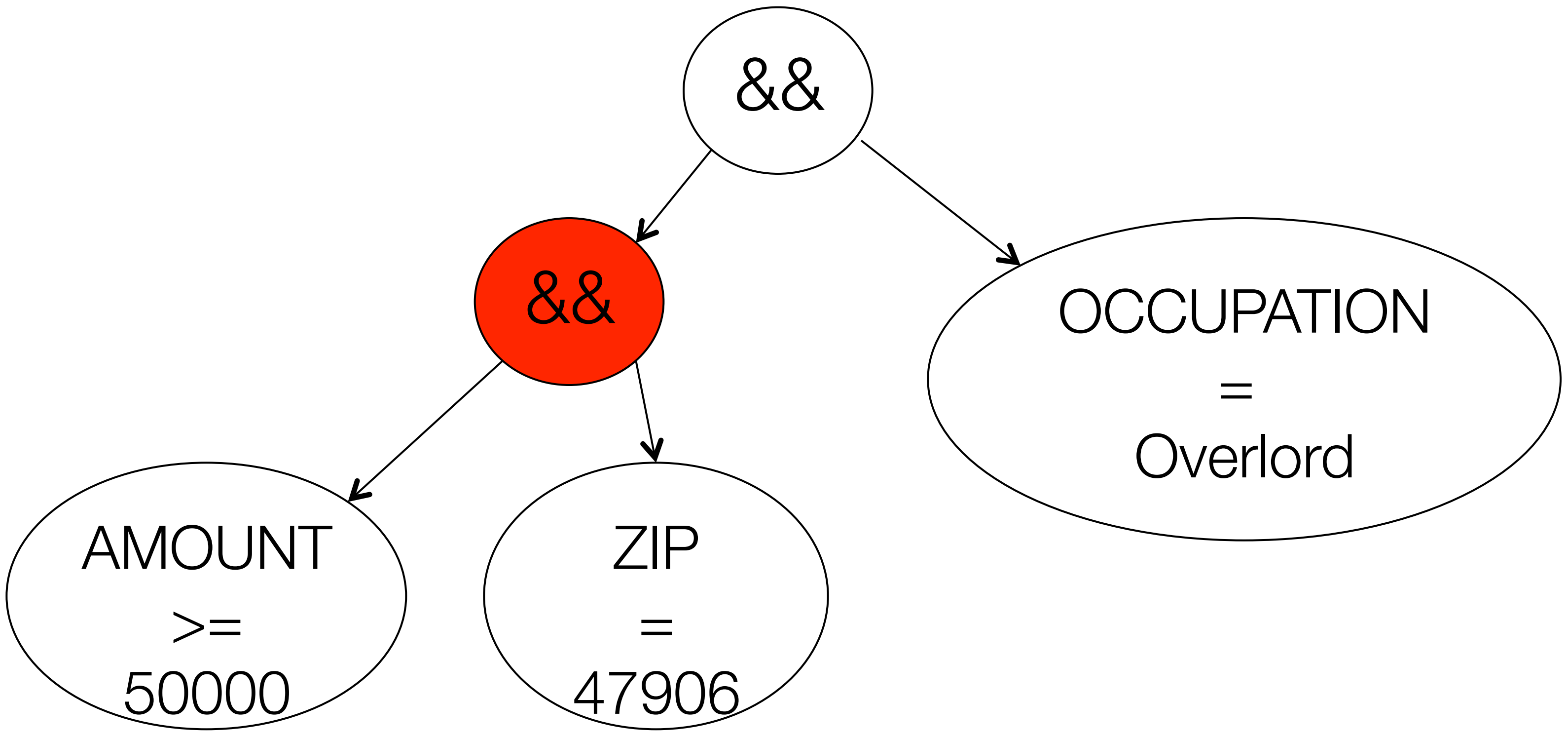


How to make a Tree

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



One connective: => Add Connective and Field nodes



007: Super Snoop

- Now you have a query tree, what can you do with it?
- Use it to snoop through credit card data!
- Look for occupations, names, etc. in any combination you like!

- Use your new powers wisely

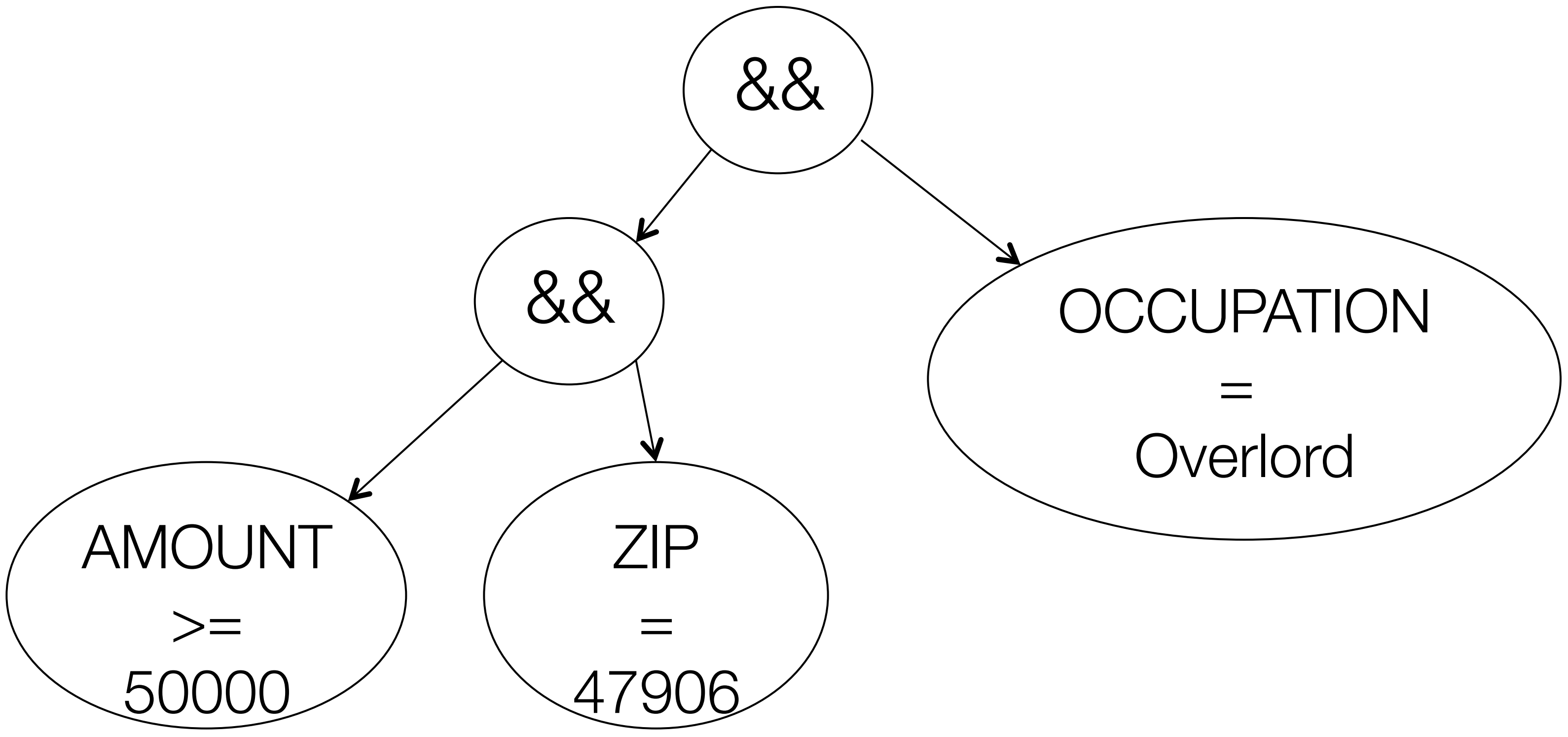
007: Super Snoop (Snoopiest?)

Modify your good friend snoop.c so that it outputs rows that match your query tree.

When does a row match?

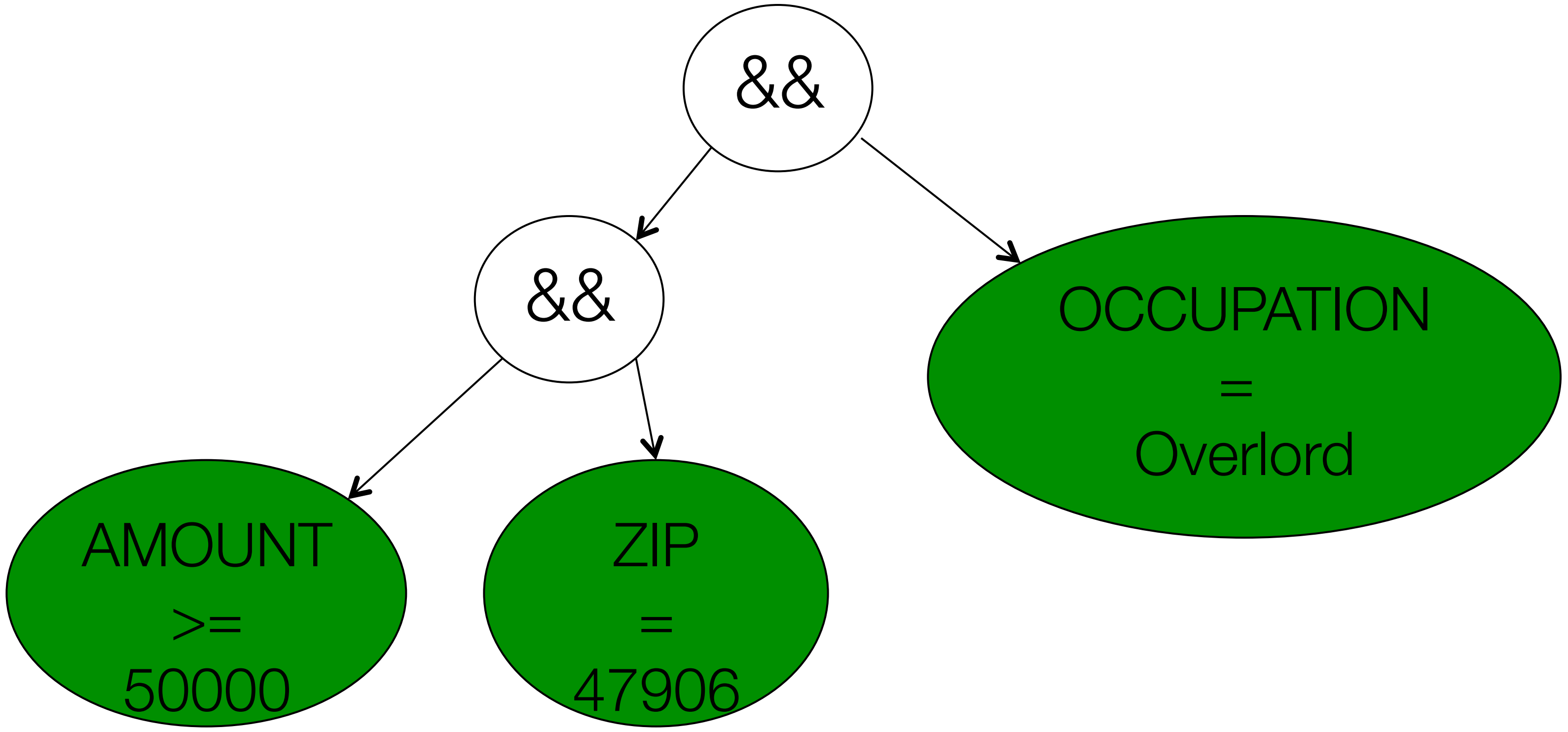
Evaluate each leaf, and then each node based on the value of its children.

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



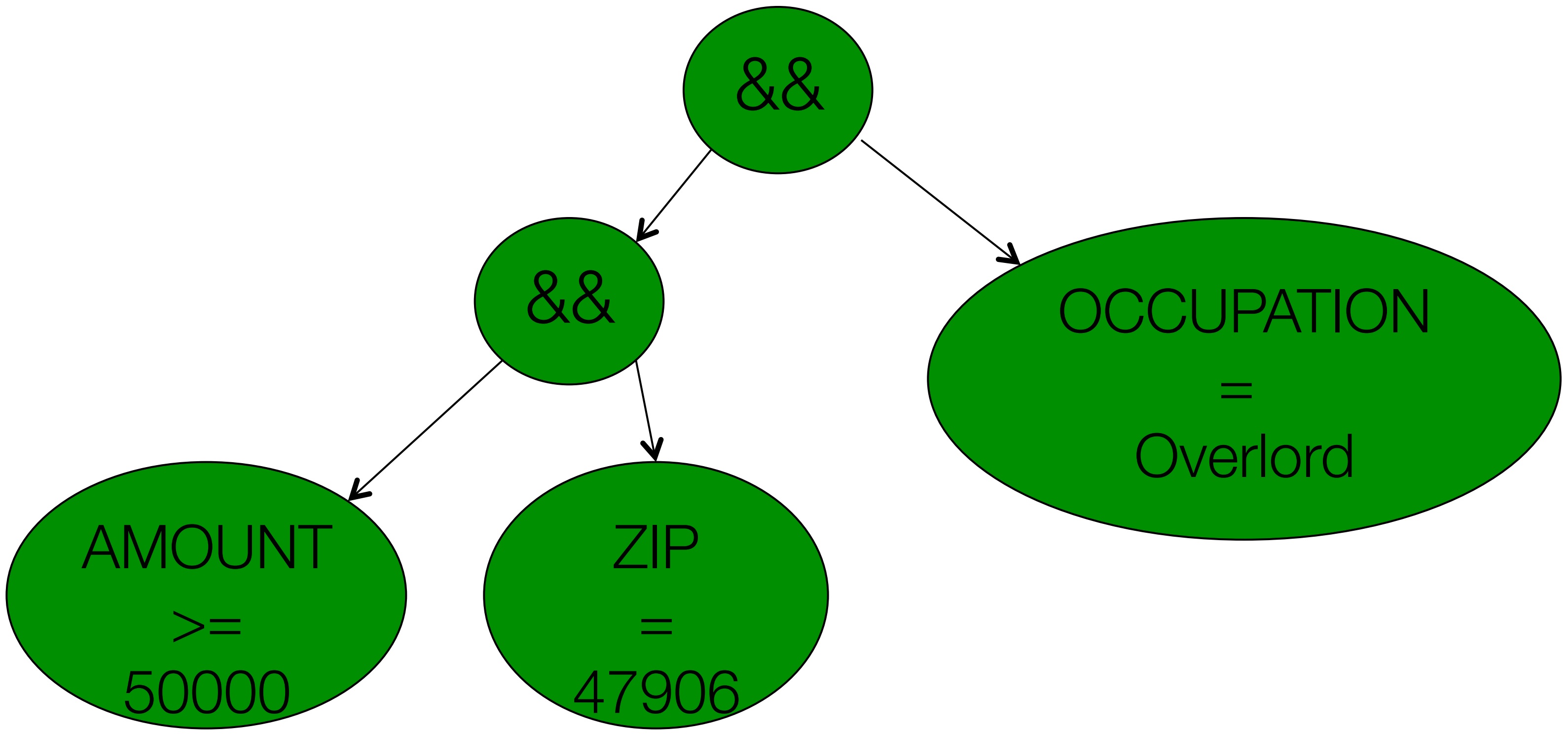
Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



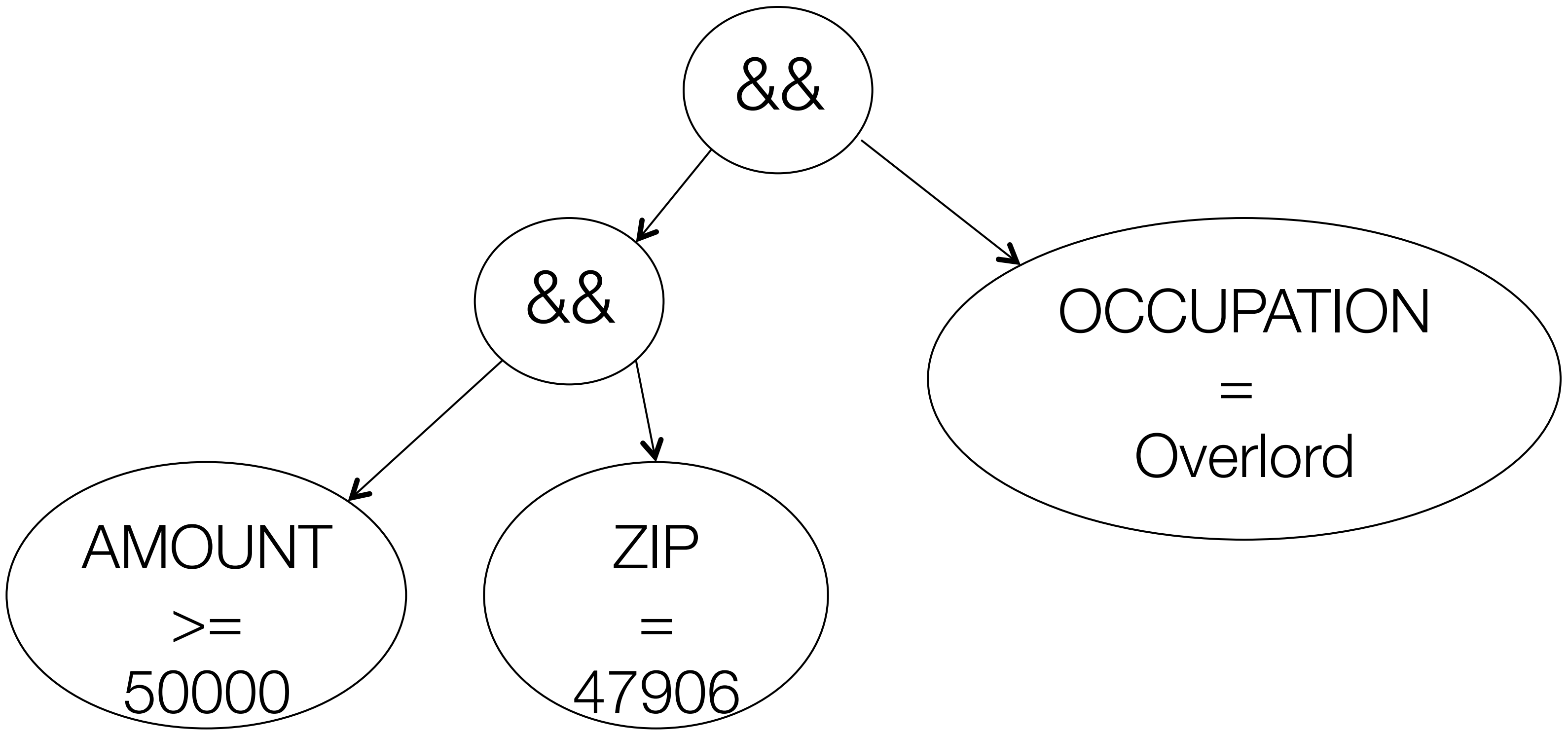
Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



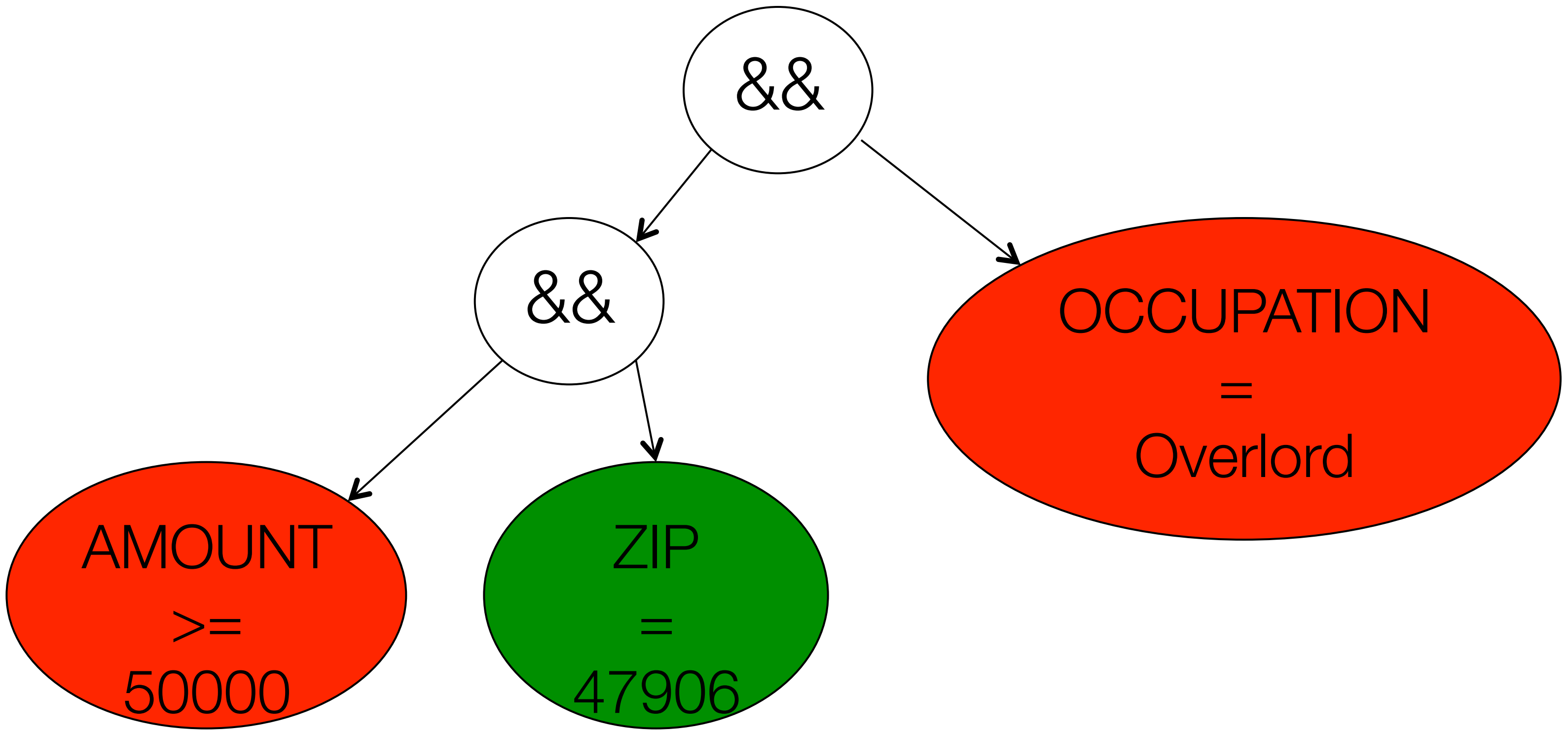
Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



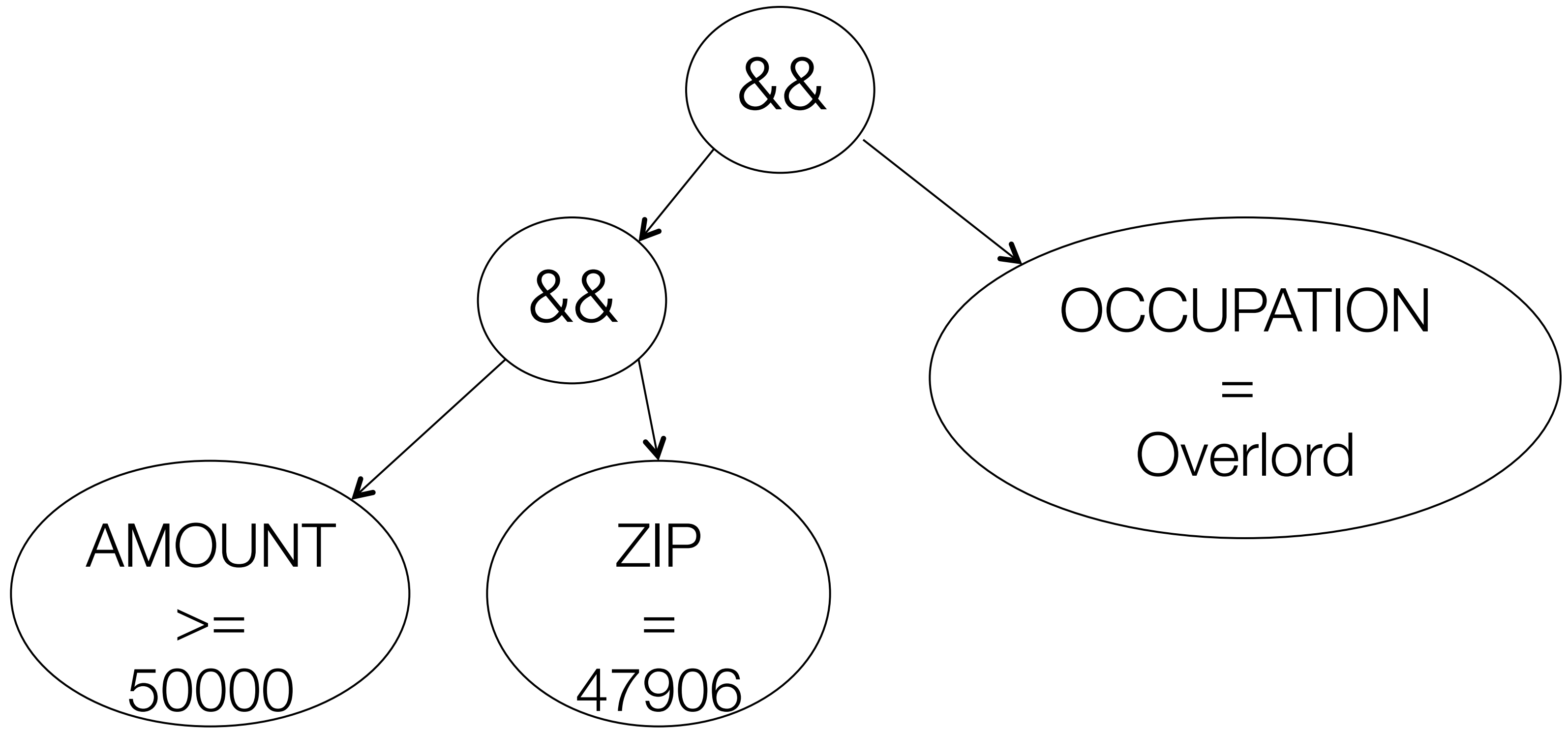
Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

AMOUNT>=50000 && ZIP=47906 && OCCUPATION=Overlord



Name	Zip	Occupation	Amount
Vitek	47906	Overlord	100000000.00
Nathan	47906	Minion	.01

Memory Leaks

Run your code using valgrind

Live Demonstration: leaky.c and fixed.c

```
Valgrind --leak-check=full ./leaky hello
Valgrind --leak-check=full ./fixed hello
```