Interesting Ideas in Programming Languages

Eric Skoglund

Internetstiftelsen

2020-12-17

Introduction

- Programmer in the registry dev team.

Introduction

- Programmer in the registry dev team.
- Interested in programming language theory. *Started* thesis on object oriented concurrent by default programming language.

Introduction

- Programmer in the registry dev team.
- Interested in programming language theory. *Started* thesis on object oriented concurrent by default programming language.

Looking at how different languages solve problems can inform you in your daily work (and it is fun).

Binary Pattern Matching

- switch
- destructuring
- (almost) everything

Binary Pattern Matching

- switch
- destructuring
- (almost) everything

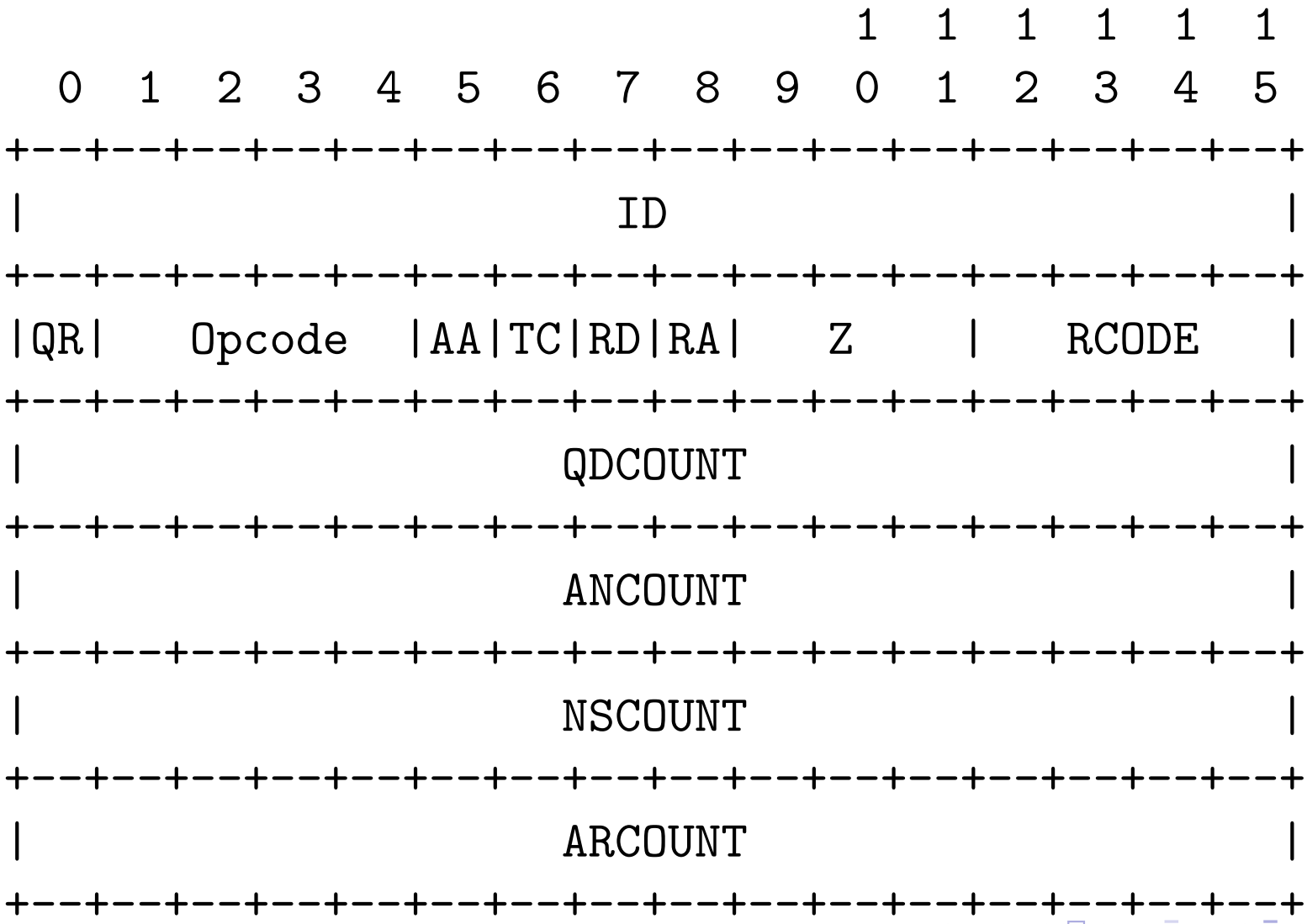
```
sum(L) -> sum(L, 0)
sum([H | T], N) -> sum(T, N+H);
sum([], N) -> N.
```

Binary Pattern Matching

Task: Parse a DNS packet (RFC 1035 header).

Binary Pattern Matching

Task: Parse a DNS packet (RFC 1035 header).



Binary Pattern Matching - C

Parse header in C

```
msg->id = isc_buffer_getuint16(source);
tmpflags = isc_buffer_getuint16(source);
msg->opcode = ((tmpflags & DNS_MESSAGE_OPCODE_MASK)
                        >> DNS_MESSAGE_OPCODE_SHIFT);
msg->rcode = (dns_rcode_t)(tmpflags & DNS_MESSAGE_RCODE_MASK);
msg->flags = (tmpflags & DNS_MESSAGE_FLAG_MASK);
msg->counts[DNS_SECTION_QUESTION] = isc_buffer_getuint16(source);
msg->counts[DNS_SECTION_ANSWER] = isc_buffer_getuint16(source);
msg->counts[DNS_SECTION_AUTHORITY] = isc_buffer_getuint16(source);
msg->counts[DNS_SECTION_ADDITIONAL] = isc_buffer_getuint16(source);
```

Binary Pattern Matching - Erlang

```
<<ID:16, QR:1, Opcode:4,
  AA:1, TC:1, RD:1, RA:1, Z:3, RCODE:4
  QDCOUNT:16, ANCOUNT:16, NSCOUNT:16,
  ARCOUNT:16, DNSBody/binary>> = DNSPacket,
  ...
```

Property based testing

Beware of bugs in the above code; I have only proved it correct, not tried it. – Donald Knuth, 1977

Property based testing

Beware of bugs in the above code; I have only proved it correct, not tried it. – Donald Knuth, 1977

Types of tests

- Unit tests

Property based testing

Beware of bugs in the above code; I have only proved it correct, not tried it. – Donald Knuth, 1977

Types of tests

- Unit tests
- Integration tests

Property based testing

Beware of bugs in the above code; I have only proved it correct, not tried it. – Donald Knuth, 1977

Types of tests

- Unit tests
- Integration tests
- E2E tests
- Acceptance tests

Property based testing

Beware of bugs in the above code; I have only proved it correct, not tried it. – Donald Knuth, 1977

Types of tests

- Unit tests
- Integration tests
- E2E tests
- Acceptance tests
- Property based tests

Property based testing

```
# Takes a nonempty list of integers and finds
# the max value.
def max(ints) do
    ...
end
```

Property based testing

```
# Takes a nonempty list of integers and finds
# the max value.
def max(ints) do
    ...
end

test "find the max value" do
    assert max([0]) == 0
    assert max([1,1]) == 1
    assert max([1,2,3]) == 3
    assert max([3,2,1]) == 3
    assert max([-1, -100, -10000, ...]) == -1
    ...
end
```

Property based testing

```elixir
# Takes a nonempty list of integers and finds
# the max value.
def max(ints) do
    ...
end

property "find the max value" do
    forall x <- non_empty(list(integer())) do
       max(x) == List.last(Enum.sort(x))
    end
end
```

Propetry based testing

```
Failed: After 10 test(s).
[-1,6,3,1]
Shrinking ...(3 time(s))
[0,1]
...
```

Propetry based testing

```
Failed: After 10 test(s).
[-1,6,3,1]
Shrinking ...(3 time(s))
[0,1]
...

OK: Passed 100 test(s).
===>
```

Propetry based testing

Positives

- Large amount of tests (after writing good properties more tests are "free").
- Tests change each run.
- Shrinks to find the smallest failing input.
- Higher confidence that your code if bug free.

Negatives

- Big effort to write properties. Often not trivial.
- Takes longer time to run.

Property based testing

Is it used anywhere?

Project FIFO 460 lines of property tests: Covered 60k lines of code. Found 25 "important" bugs.

LevelDB 600 lines of propetry tests. Found sequences of 17 and 31 specific calls that could corrupt the database.

Type Driven Development

- Created by Edwin Brady - Creator of Idris

Type Driven Development

- Created by Edwin Brady - Creator of Idris
- TDD works by

Type Driven Development

- Created by Edwin Brady - Creator of Idris
- TDD works by

  Type Write a type for your function or inspect the type of a hole.

Type Driven Development

- Created by Edwin Brady - Creator of Idris
- TDD works by

  Type Write a type for your function or inspect the type of a hole.

  Define Create a (possible incomplete) implementation.

Type Driven Development

- Created by Edwin Brady - Creator of Idris
- TDD works by

  Type Write a type for your function or inspect the type of a hole.

  Define Create a (possible incomplete) implementation.

  Refine Improve the definition by filling a hole or making the type more precise.
- Works better with powerful type systems.

Type Driven Development

- Created by Edwin Brady - Creator of Idris
- TDD works by

  Type Write a type for your function or inspect the type of a **hole**.

  Define Create a (possible incomplete) implementation.

  Refine Improve the definition by filling a **hole** or making the type more precise.
- Works better with powerful type systems.

Type Driven Development

DEMO

End

There are interesting and useful things in the world of PLs, go explore!