HOL4P4: Semantics for a Verified Data Plane

Euro P4 | 2022

Anoud Alshnakat
Didrik Lundberg
Roberto Guanciale
Mads Dam
Karl Palmskog

Div. of Theoretical Computer Science, KTH

I asked chatGPT....

Can P4 programs be prone to errors?

Yes, just like any other type of program, P4 programs can be prone to errors.

Can P4 programming errors lead to security vulnerabilities?

Yes, P4 programming errors can lead to security vulnerabilities.

Formal verification of data planes

Verification

➢ Complementary to testing

FM were applied to the abstract protocol level

➢ General purpose languages were hard to analyze (e.g. pointers)

FM at the implementation level

➢ P4 simplicity makes it feasible to analyse the program (no loop statements, pointers, ...etc)

HOL4P4 semantics for P4

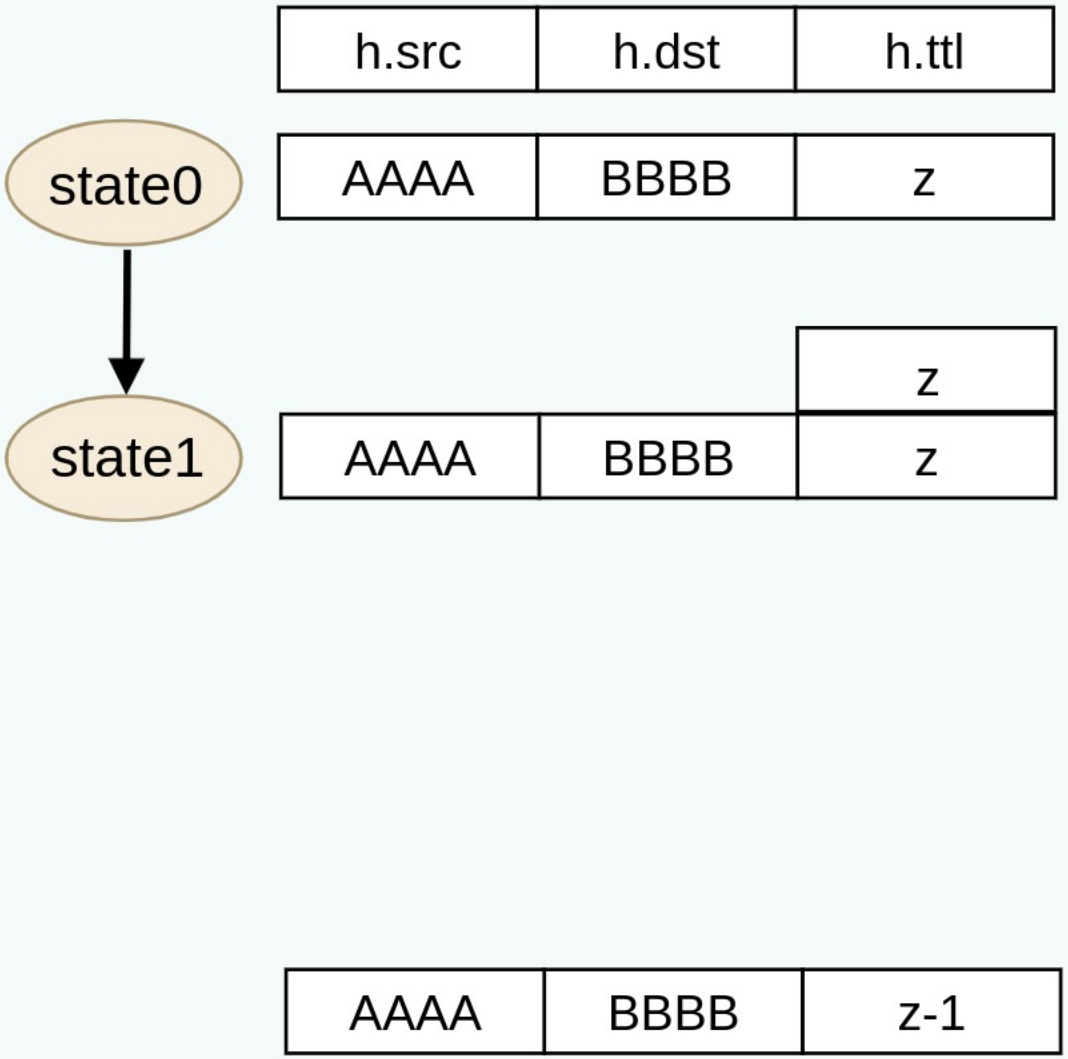
Key features:

- Heapless
- Small-step transition system
- Defined in HOL4 interactive theorem prover

Developed to conform with P4 specification sheet

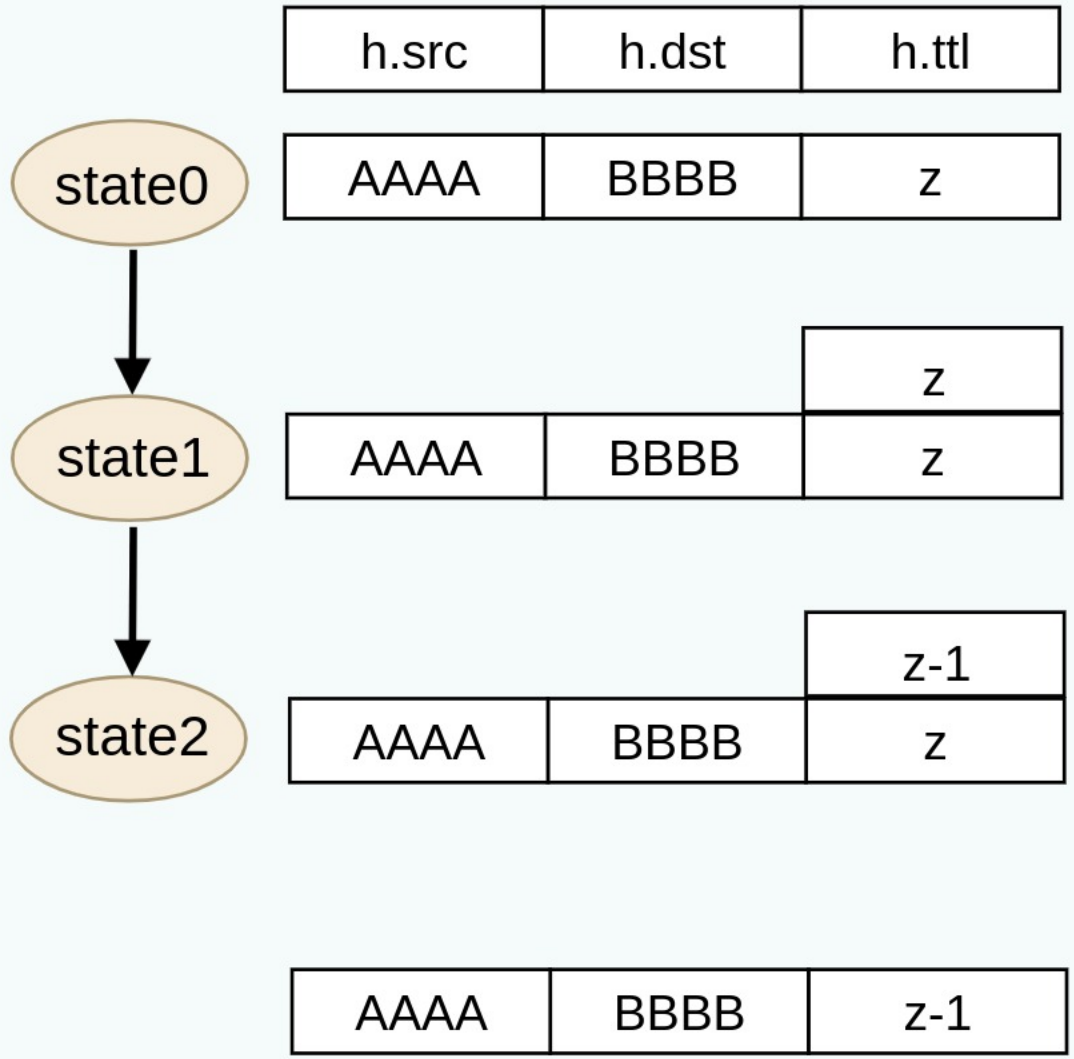
State transition

```
decrease_ttl ( inout bit< 32 > x ) {
  x = x - 1;
}

control ctrl ( inout PP h ) {
  decrease_ttl  (h.ttl);
  ...
}
```



State transition

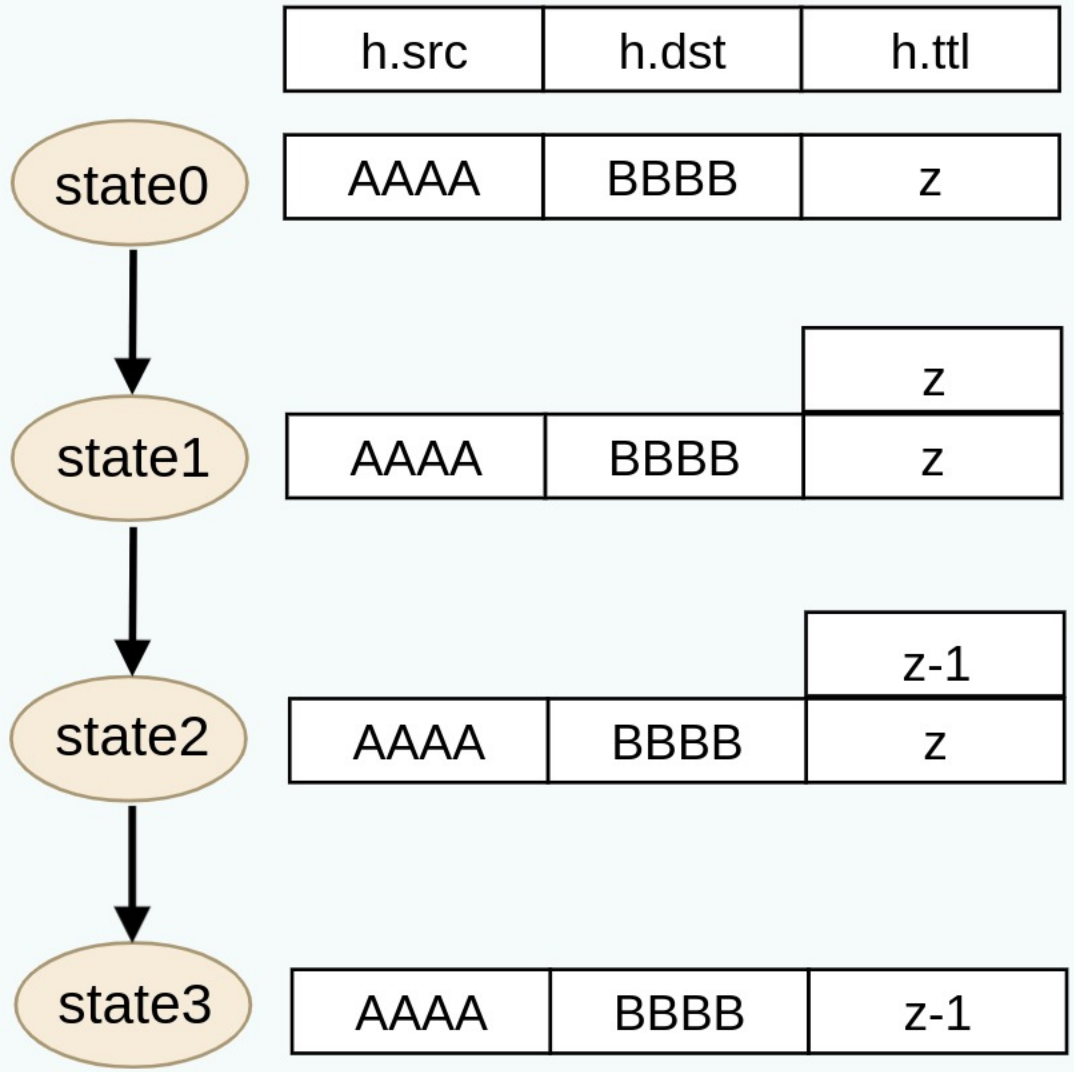
```
decrease_ttl ( inout bit< 32 > x ) {
  x = x - 1;
}

control ctrl ( inout PP h ) {
  decrease_ttl  (h.ttl);
  ...
}
```



State transition

```
decrease_ttl ( inout bit< 32 > x ) {
  x = x - 1;
}

control ctrl ( inout PP h ) {
  decrease_ttl  (h.ttl);
  ...
```

Interleaving and concurrency

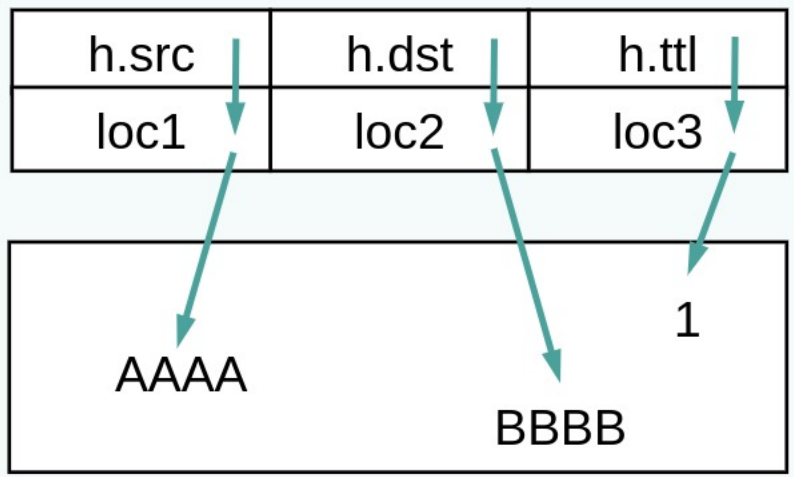


Heapless semantics

Heapless memory

P4 lacks referencing
and pointers

Heap memory

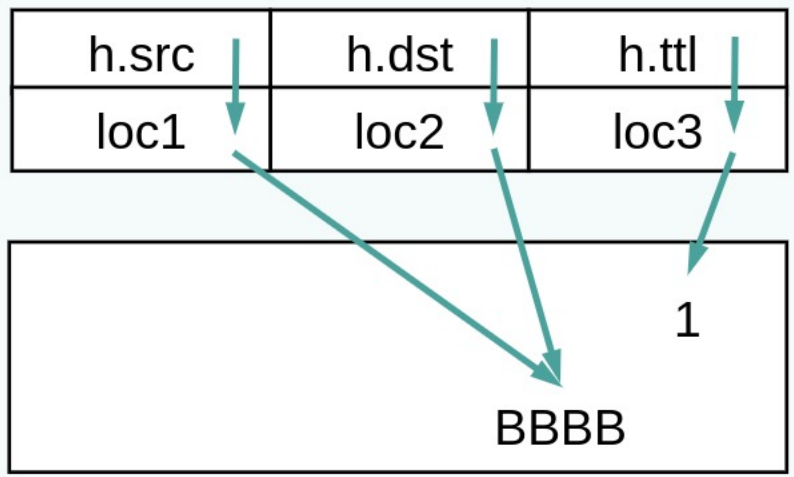


Heapless semantics

Heapless memory

P4 lacks referencing and pointers

Heap memory

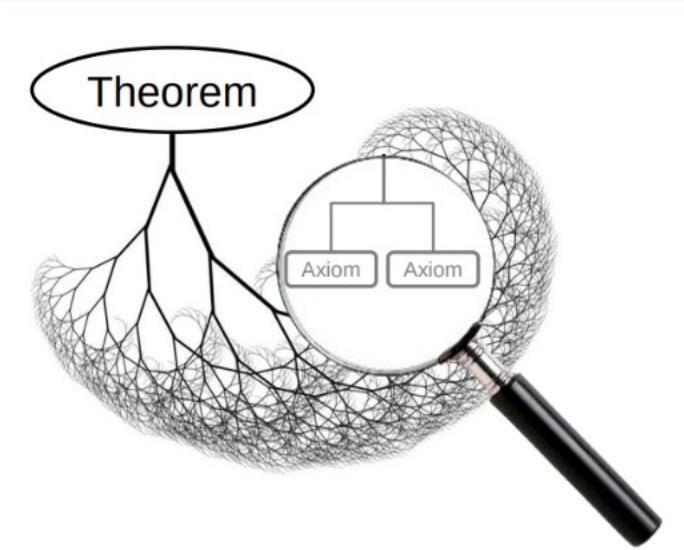


- Aliasing complicates proofs
- Garbage collection

PoC verification tool

Uses the execution of the system symbolically

- Implemented in HOL4 theorem prover
- Can produce machine-checkable proof certificates
    - Properties proved for all possible inputs
- Evaluated on the VSS example



Conclusion

- Formal semantics for P4
  - Small step
  - Heapless
  - In HOL4 interactive theorem prover
- PoC tool that can generate machine-checkable proof certificates

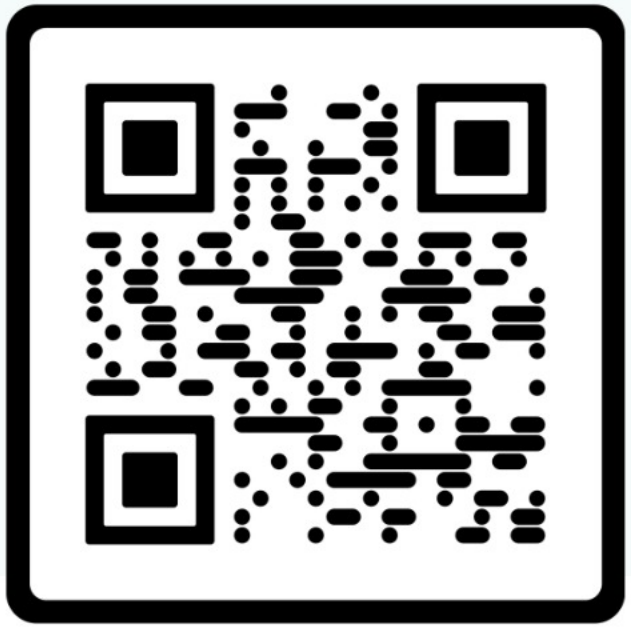
Future work

- Complete the foundations
    - Type system
    - Include all P4 constructs
    - Parse P4 programs to HOL4 terms
    - Modeling more architectures
- Validate the semantics via differential testing
- Develop certifying verification tools of P4 programs

Thank you!

Anoud Alshnakat
anoud@kth.se

Didrik Lundberg
didrikl@kth.se



github.com/kth-step/HOL4P4