
Cyclone

A Very Short Introduction

Dan Grossman

Cornell University Computer Science

Cyclone in One Slide

*A safe, convenient, and modern language/compiler
at the C level of abstraction*

- Safe: Memory safety, abstract types, no core dumps
- C-level: User-controlled data representation, easy inter-operability, resource-management control
- Convenient: “looks like C, acts like C”, but may need more type annotations
- Modern: discriminated unions, pattern-matching, exceptions, polymorphism, advanced type system, ...

Why Bother?

- Safety-critical systems that must control data representation
 - e.g. low-level plug-ins
- Performance tuning without throwing away the prototype
- Exploit massive infrastructure and knowledge-base of C

Silly Example

```
void * swap(void **x) | `a swap(`a @{2} `r x)
{                       | {
    void * t = x[0];    |     `a t = x[0];
    x[0] = x[1];        |     x[0] = x[1];
    x[1] = t;          |     x[1] = t;
    return x[0];       |     return x[0];
}                       | }
```

- Compiler needs info to check safety of body and call sites
- Call sites do not change !

Status

- Bootstrapped compiler (approx. 1 minute)
- Libraries
- Documentation (more in progress)
- Easy to install and build (gcc, gnumake, a couple environment variables)
- URL announcement soon, meanwhile mail us:
 - Greg Morrisett (jgm@cs.cornell.edu)
 - Trevor Jim (trevor@research.att.com)
 - Dan Grossman (danieljg@cs.cornell.edu)

Yeah, We Do That... (With Caveats)

pointer arithmetic, unions, varargs, nested structs, address of stack variables, function pointers, polymorphism, casts to supertypes, declaration w/o initialization, type inference, array-bounds checks, extensible datatypes, namespace control, link-compatibility checking, ...

What do you want to see? (We'll write code)