Notes - 7/22

Friday, August 05, 2011 9:25 AM

Nice axiom in C#: we never infer types that weren't mentioned

Issue: inference questions around

```
Xnode
|- XElement
|- XText
|- XComment
```

If there is a branch

Data from Steve:

- Self host on new type system
- Overloading
- Modules
- Running 40,000 lines of JS through it
  - Wasn't a single case where full merge types are necessary
  - Was sufficient to say strict superset of subset

Questions: Number and string – currently that's string

Lattice and partial order:

-

getElementById:

- Okay to return IElement

Two scenarios:

-

$(fn): void
$(string):

Claim: Anything we infer should be denotable structurally, not declaratively

Question: Do we support generics? Nope.

```
interface IArray
{
  reduce(f: (acc: any, next: IElement) => any): any;
}
```

Function signatures:

- Shorthand still in
- Means exactly the same thing
- Requires parameter names
- Could allow ": foo" or "_:foo"

```
var x : O.I= {
  g: function(a): any  {
    return "hello";
  }
}
```

Decision: Common supertype – and there better be one

- ES5 concepts should pass through
  - ES5 flag

Tangent on WinJS load time

Implementing overloads:

- Go to 'any'

Inference rules:

- Object initializers: interface types based on inference of members
    - 3 incidences of var x = {}; x.y
- Array literal: mergeOrdered
-

Would love to do:

- Suggest examples
- Unit tests
- Write unit tests

By Monday:

-

Question: Var x = null;

1. null type
2. any
3. dissallow
4. nag

Base questions:

- don't' distinguish base types and interface types
- want to enforce that a class can only have a single class
- A name can refer to a class or an interface
- A name looks up a symbol
- A symbol has a type
- The symbol may be a class

Nice characteristcs:

- Compiles the internet, possibly with lots of warnings
-

Undefined: any

var x; // any
var x = undefined; // nag (cannot infer a type)+ any
var x = null; // nag (cannot infer a type) + any

From: Luke Hoban
Sent: Friday, July 22, 2011 9:27 AM
To: Strada Design Team
Subject: Strada Design Meeting Agenda - 7/22/2011

Agenda:

- Update on implementation from Steve
- Overload resolution algorithm (mail from Steve)
- Declaration of merge types (A|B syntax as type constructor, or only as type declaration?)
    - Don't think so
- Inference
- Extern and retrofitting types on global code
- Backlog:
    - Promises typing
    - Top level scope?
    - Uninitialized variables/inference/defaults
    - Enums
    - Global module
    - Foreach
    - Class-parameters as privates and inheritance
    - Function type compatibility (omit parameters, more parameters, varargs, etc.)
    - Any -> number (what gets emitted?)
    - "await"
- Anything else?