

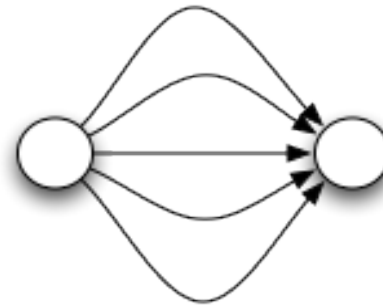
Lattakia (A language for Lattices)

Wael Salloum
Hebatallah Elfardy
Katherine Scott
Li Yifan

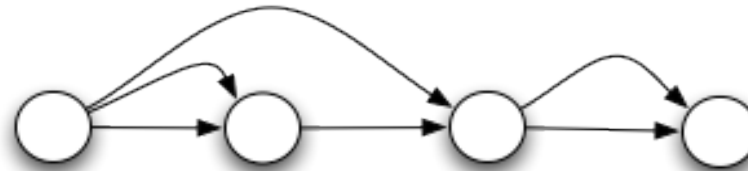
What's a lattice?



Sequence Lattice (seqlat)

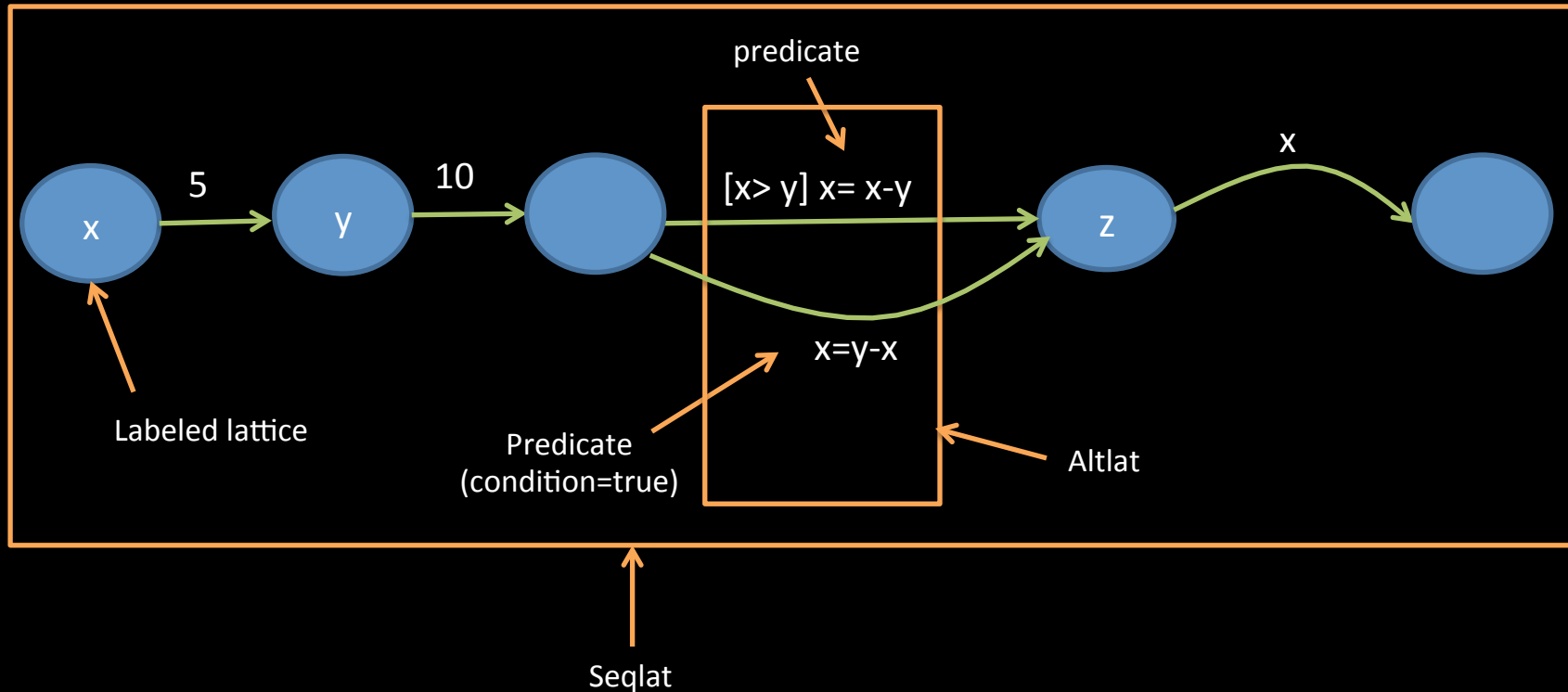


Alternative Lattice (allat)



Lattice (combination of seqlat and allat)

A more detailed example



```
..Code for the above lattice
x: 5;
y: 10;
[x > y ] x = x - y | y = y + 5;
z: x;
```



Lattakia in a Nutshell

```
**  
A MULTI LINE COMMENT  
**  
.. A single line comment  
foo = (1;2;3); .. a sequence lattice of integers  
bar = (1.0|2.0|3.0); .. an alternative lattice of floats  
baz = (a:1|b:2|c:3); .. a labeled alt lat  
  
print(baz.a); .. access a and print it  
print(foo{0}) .. Prints 1  
print(bar[0]) .. Prints 1.00  
  
def fnord(x;y) = ([x>=y] print(x); | print(y); ); .. a function definition  
herp = 10;  
derp = 20;  
  
let fnord(herp;derp); .. Function application prints 10  
  
←- MORE HERE →
```

Quick Sort Example

```
**
Perform quicksort on a seqat of integer values.
**
def quicksort(input) =
(
  let (less = (); greater = ()); .. create two local variables
  [input.length <= 1] return = input | .. if we have a single value return

  let pivot = input[0]; .. create a pivot
  let input[0] = epsilon;

  foreach(x; input; .. for each value in the array
  [x < pivot] less = (less; x) .. create a less than lattice
  | greater = (greater; x) .. and a greater than lattice
  );
  .. now recursively sort the new lattices and return a lattice
  (quicksort(less); pivot; quicksort(greater));
).return; .. return the input

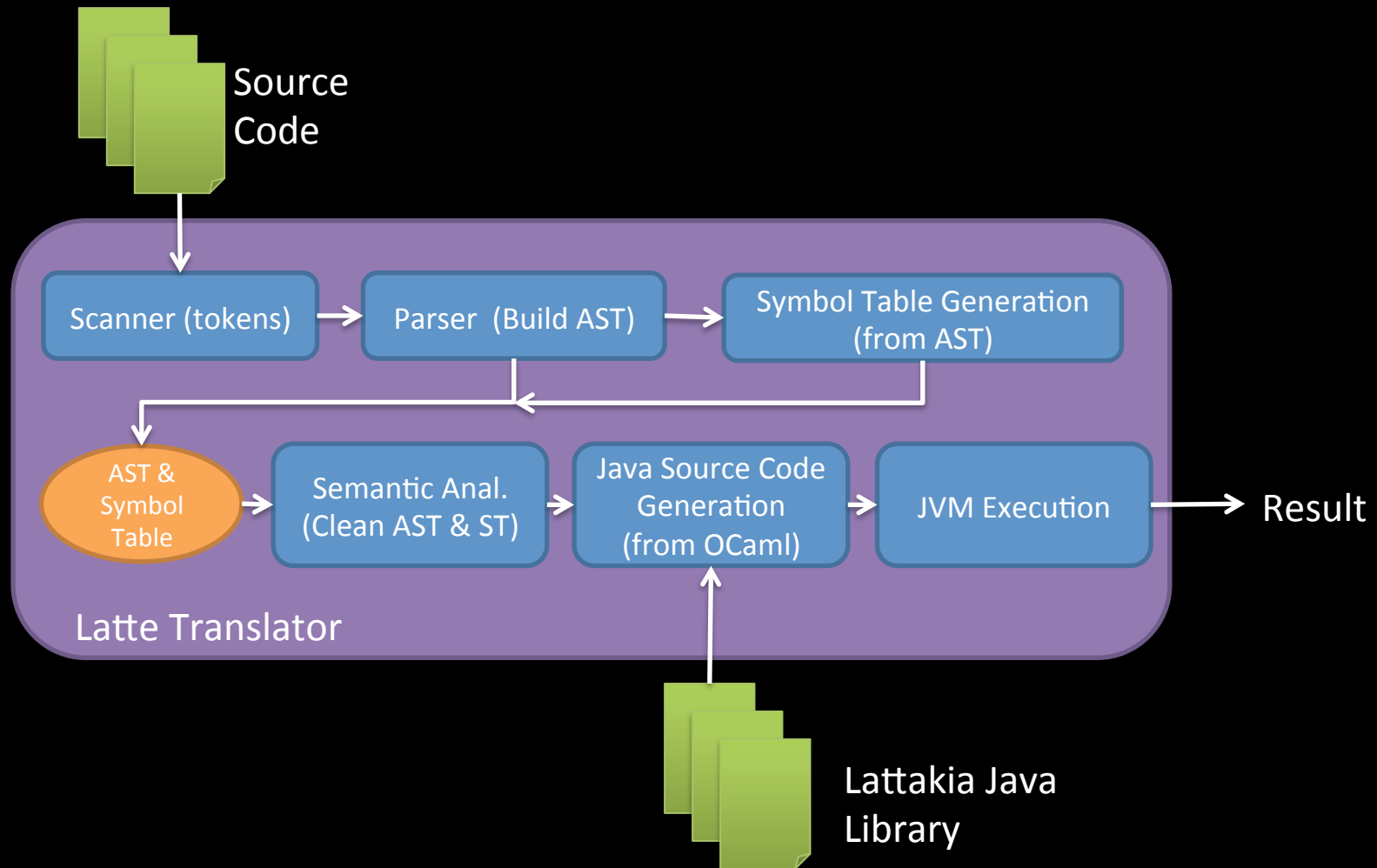
values = (42;7;18;6;1;-3;15;30);
let sorted = quicksort(values);
```

Compiling & Running GCD

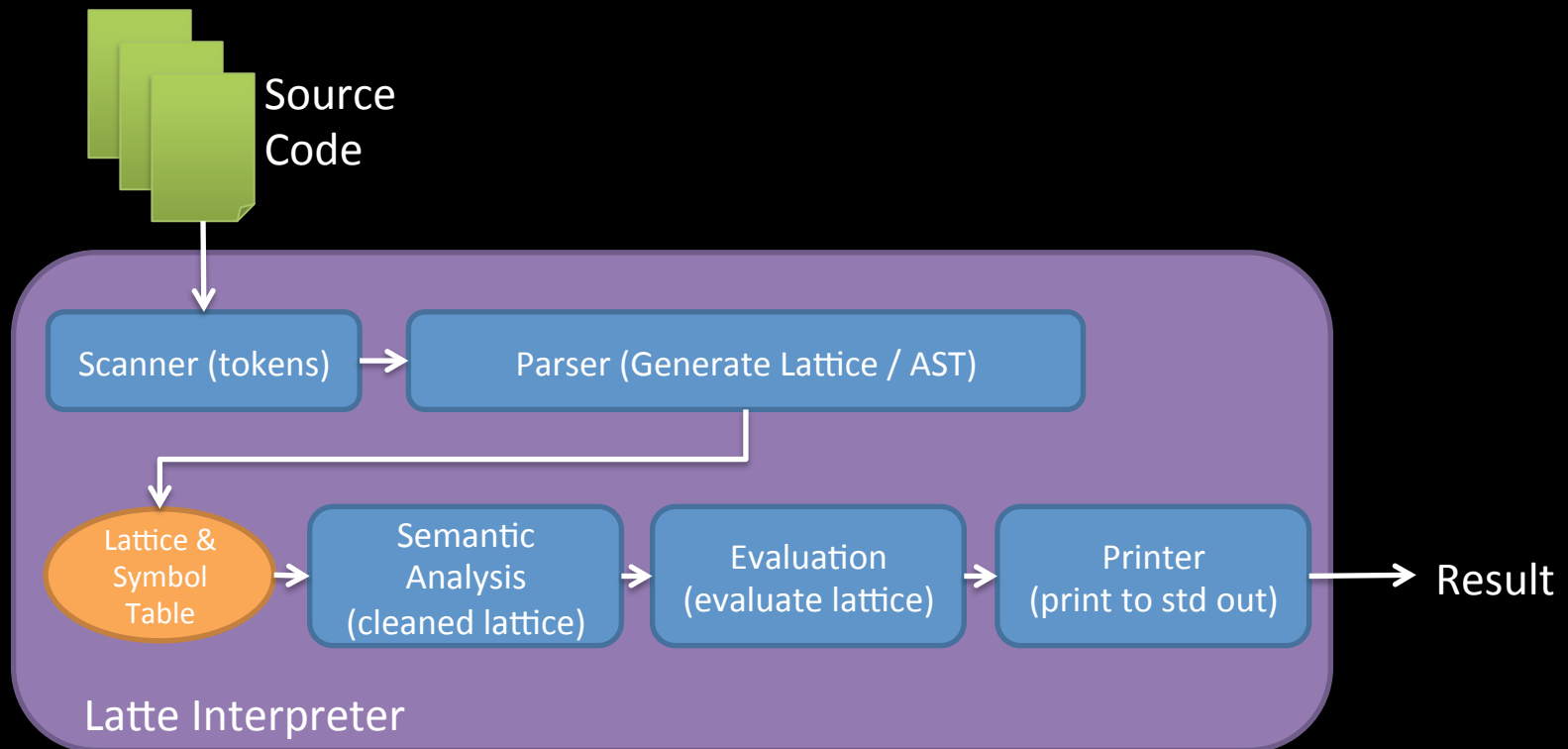
```
**  
GCD - calculate the greatest common denominator.  
  
**  
def gcd(x; y) = ( [y == 0] x  
                 | [y != 0 ] gcd(y, x % y);  
                 );  
a = 35;  
b = 49;  
print("GCD of 35 and 49 is ");  
print(gcd(a; b));  
print("\n");
```

```
> ./latte < GCD.lat  
GCD of 35 and 49 is 7  
>
```

Initial Design



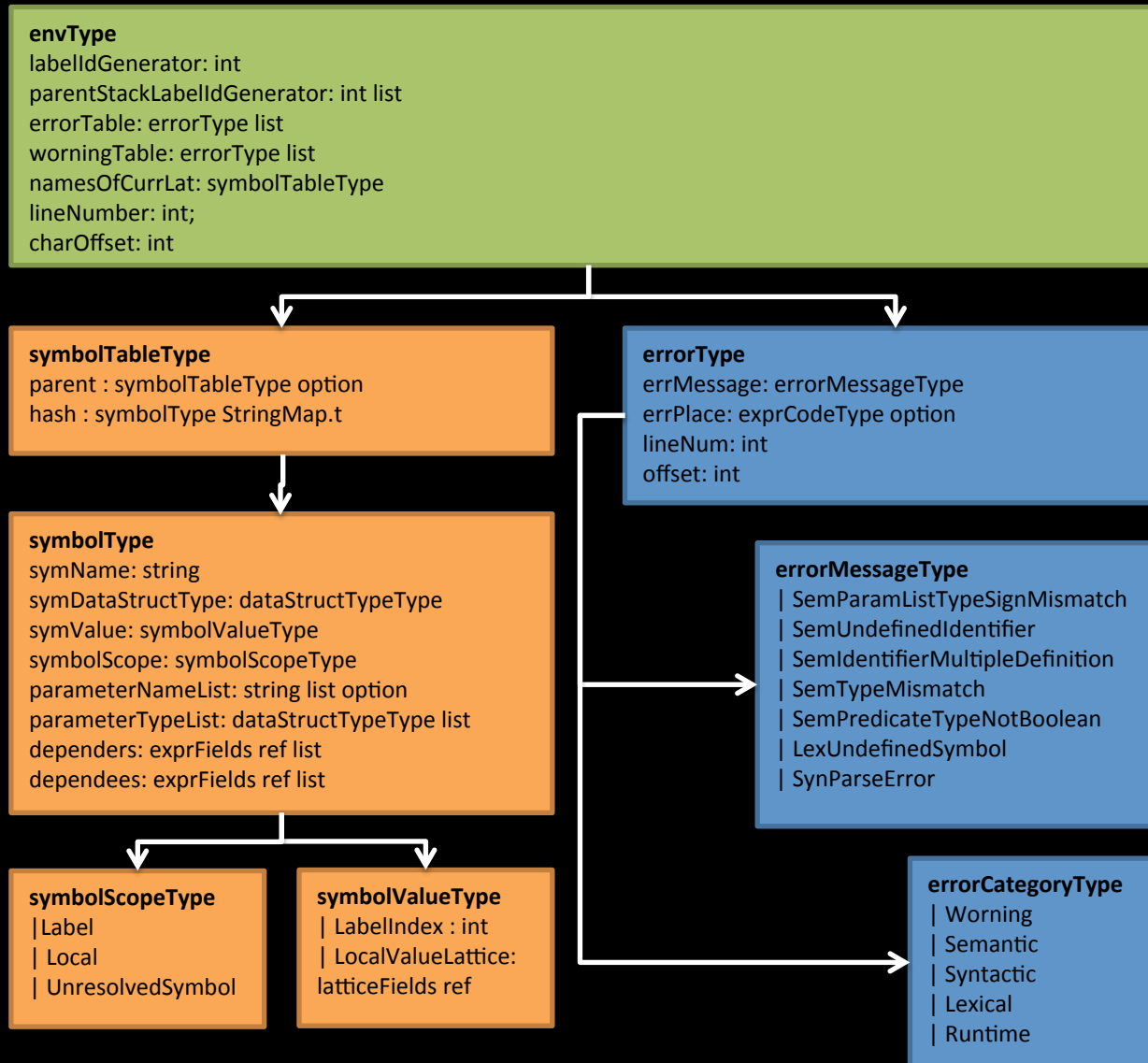
Revised Design



Lessons Learned

- Writing a compiler isn't that bad... writing one for a complex language is.
- Ocaml is actually kinda cool.
- Translation is difficult when you are trying to hit a moving target.
- Things get done when you put everyone in a room.
- Keep it simple stupid

Environment / Symbol Table



Lattice Construction

