DOMAIN SPECIFIC LANGUAGE – 9 APR 2013

A domain-specific language (DSL) is a type of programming language of a particular problem domain.

- HTML
- SQL

DSL: have abstraction for a specific domain
GPL: have GP abstraction

Two way to build DLS:

- **DLS stand alone** an advantage is that the compiler check if it correct
- **Embed DLS in GPL** a disadvantage is that you don't have debug compiler

Parsing

Is the process that analyses a string in input and give back a representation of data according to the rules of a formal grammar.
Ex of formal Grammar:

String-> List
List-> ws(int ws)*
Int->(0|1|2|3|4|5|6|7|8|9)+
ws-> (' '|'\t|'\n')*

Parser Generator

Take grammar and give you back code for parser.
Some example:

- Yacc
- Bison
- Javacup
- JavaCC
- ANTLR
- Happy
- PLY

Parser Combinators

Basic Idea: "Take small parser and create a big one".
You have primitive parser and you combine one or more of this to create another parser.

Java Version
Define parser :

A parse(String s);
<B> Parser <Parser<Pair<A,B>> then (Parser<A> p)

Then now we want recognize white space and then digit:

Parser <Character> ws = oneof(" \t\n");
Parser <Character> digit = oneof("0123456789");
Parser <String> digits = digit.oneOrMore();
Parser <String> wss = ws.zeroOrMore();
Parser<Integer> n = digits.something();

Parser<List<Integer>> L= wss.then(n.then(wss).zeroOrMore())):

Haskell version

Data Maybe a = Nothing| Just a

Type parser a =
String -> Maybe (String,a)

oneOf set “” = Nothing
oneOf set (x:xs) = If X ‘elem’ set then
Just(xs,x) else Nothing