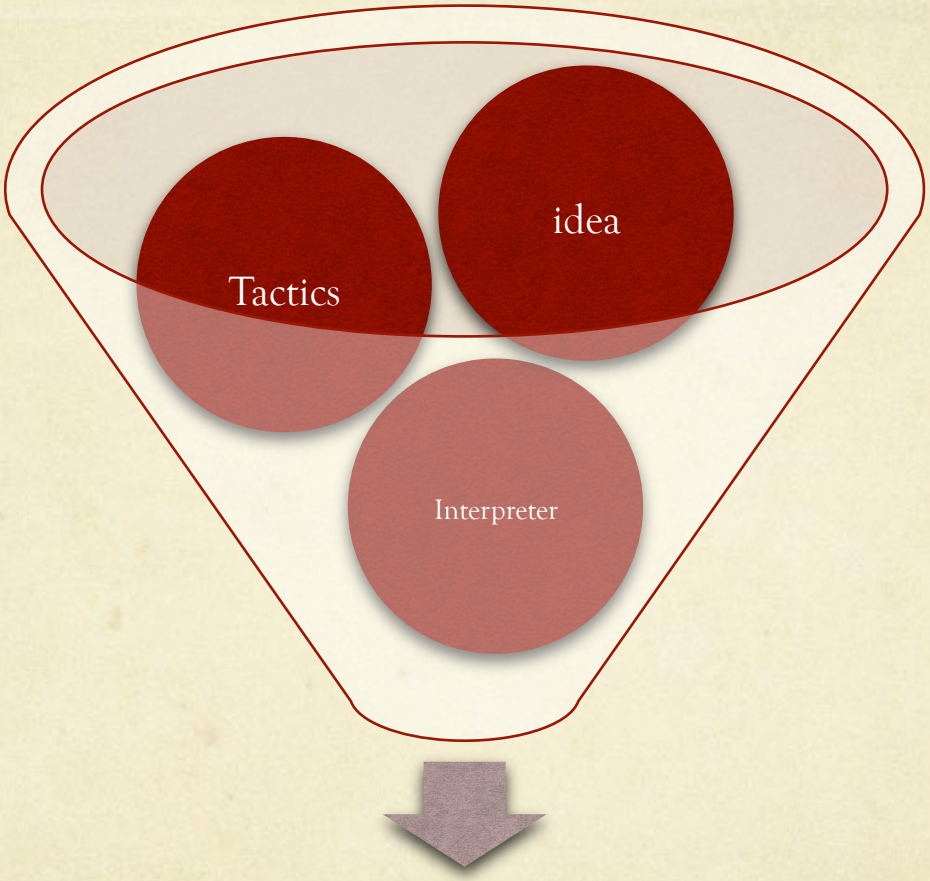

Tactics
Tactics Language
subtitle

Why Tactics



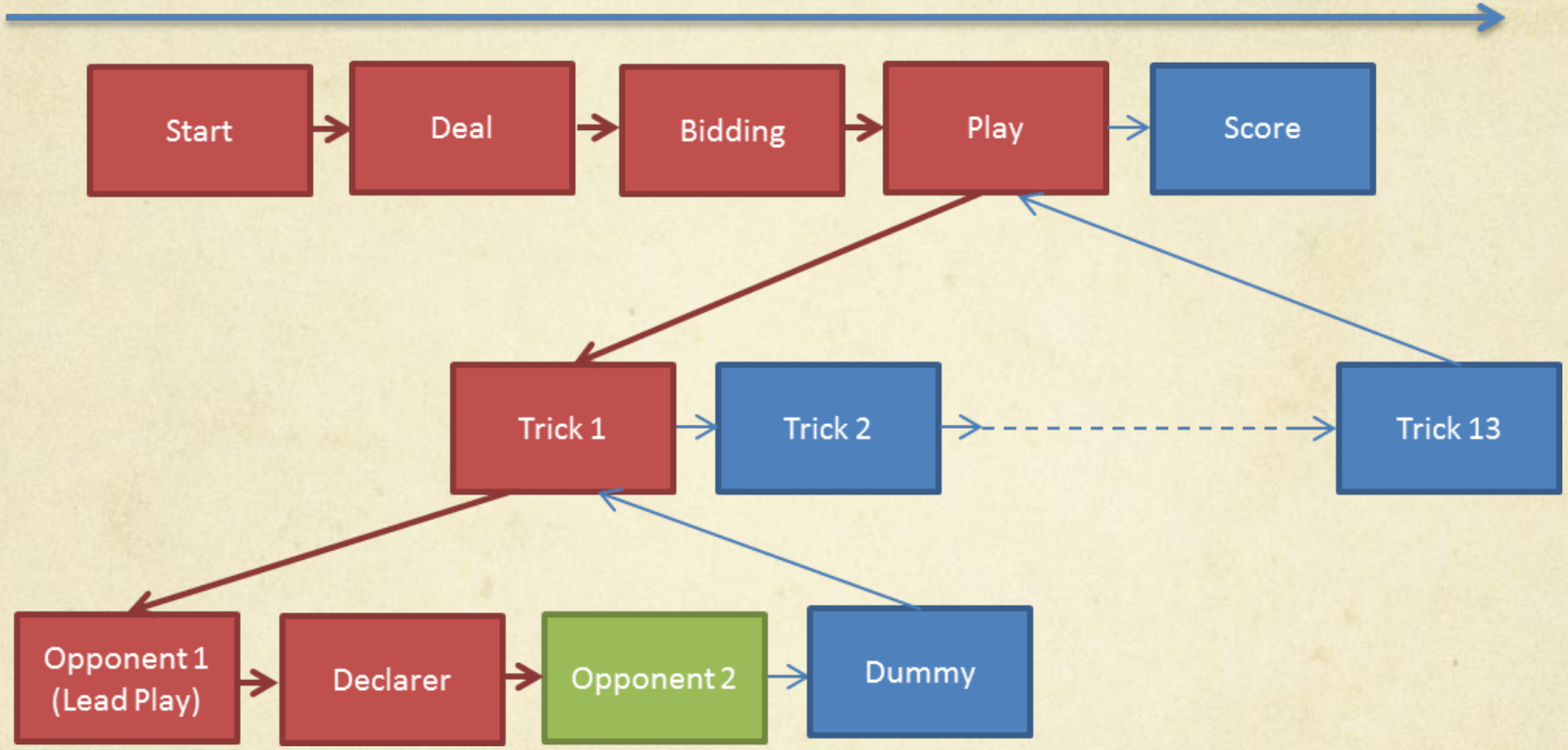
CARD GAME

Why Tactics


Traditional language
Tactics
Specific Data Type
Too huge for card game
Event base


A
K
Q
J
10

Language Feature



Bidding event flows in a Contract Bridge game

Language Feature

Language overview

```
enum suit {SPADES = 1 "Spades", CLUBS "Clubs"};
enum card <num, suit>;// Define enum for cards
enum player {WANG "Minglei", YANG "Shu"};
cont<enum card> deck;// Card deck
cont<enum card>[enum player] hand;

event entry {
    call init;  //call another event
    call newround; //another event
}

int sum(cont<enum card> playerhand)//call by reference
{
    int sum, i;//local variable

    for (i = 0; i < size(playerhand); i++) {
        sum = sum + value(extract(peek(playerhand, i), 1));
    }
} //function in tactics
```

Tactics Tutorial

enum

Event

Container

Data Type

Event

Scoping
Global Variable
Local Variable

Function

```
enum suit {SPADES = 1 "Spades", CLUBS "Clubs"};
enum card <num, suit>;// Define enum for cards
enum player {WANG "Minglei", YANG "Shu"};
cont<enum card> deck;// Card deck
cont<enum card>[enum player] hand;

event entry {
    call init;  //call another event
    call newround; //another event
}

int sum(cont<enum card> playerhand)//call by reference
{
    int sum, i;//local variable

    for (i = 0; i < size(playerhand); i++) {
        sum = sum + value(extract(peek(playerhand, i), 1));
    }
} //function in tactics
```

Tactics Tutorial

Container

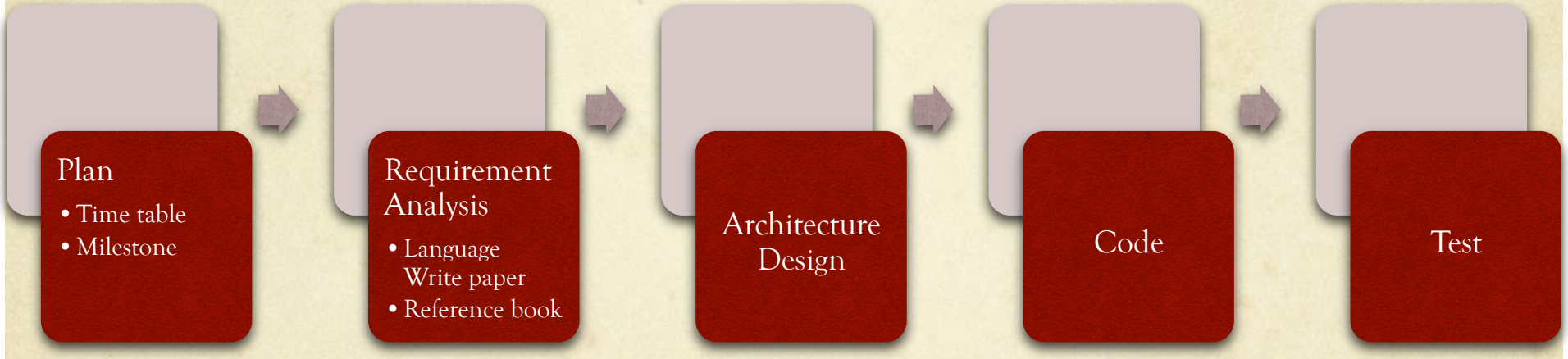
fill(cont name)
shuffle (cont name)
peek (cont name, pos)
remove (cont name, pos)
insert (cont name, pos, elem)

Enum

random(enum enum_name)
combine(elem1,elem2)
extract(elem,n)
value(elem)

Project Plan

Development Process



Project Plan

Manage team

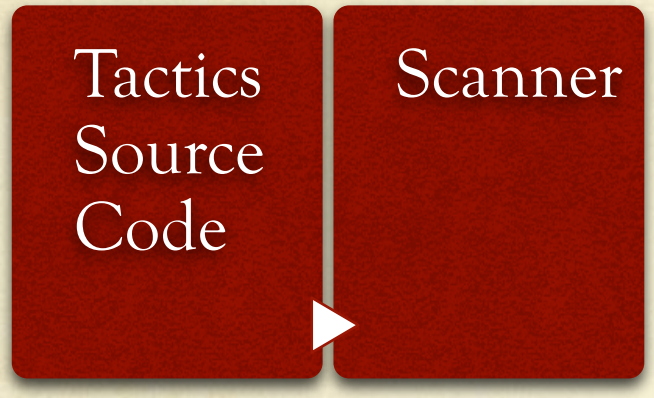
Train

- Implementation Style

Language design

- Key Feature of Tactics
- During Implementation
- Change Management

Architecture



Tactics Source Code

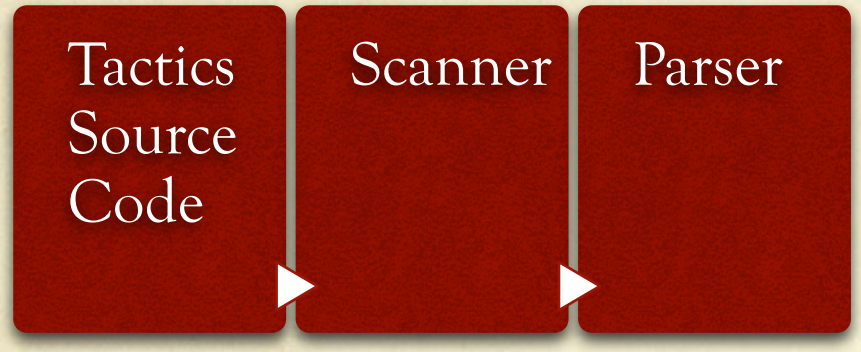
```
if (size(deck) > 0) {
do {
temp = remove(deck, 0);
insert(hand[tempplayer], top, temp);
templayer++;
} while (size(deck) > 0 &&
tempplayer != WANG);
}
```

lex

A lot of token

```
If (function(identifier) greater
number ) { do { identifier assign
remove(identifier,0)
```

Architecture

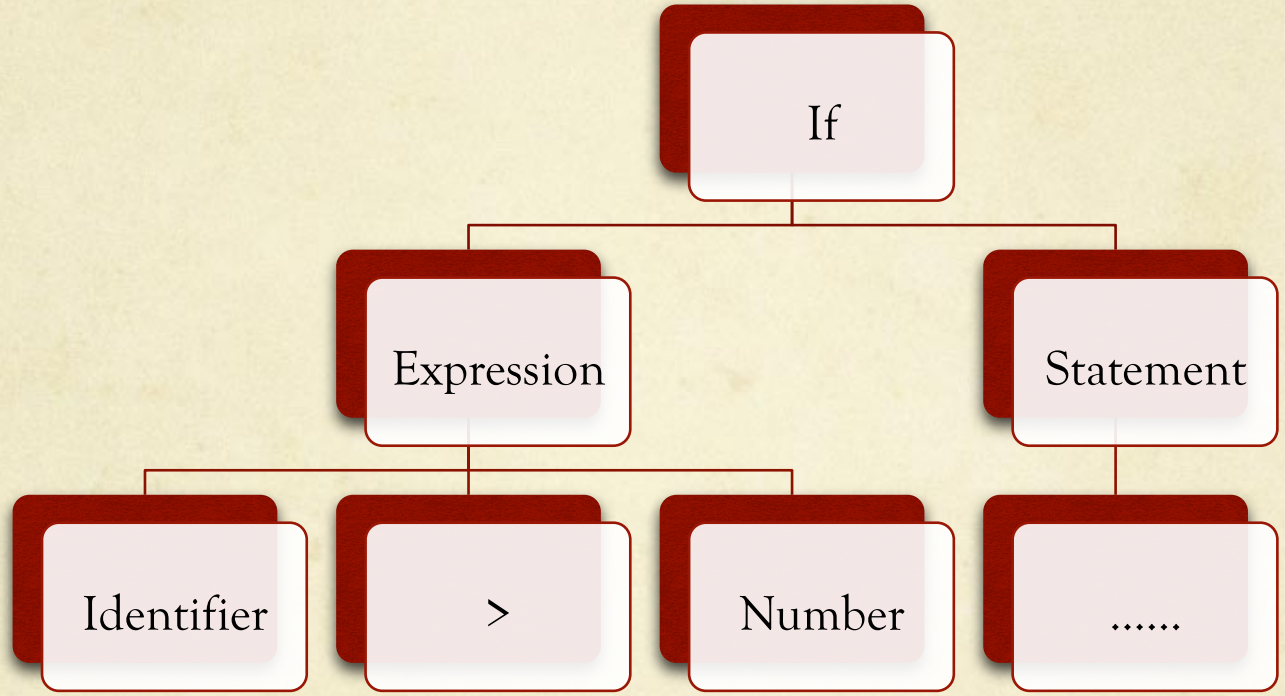


If (function(identifier) greater number) { do { identifier assign remove(identifier,0)

Yacc

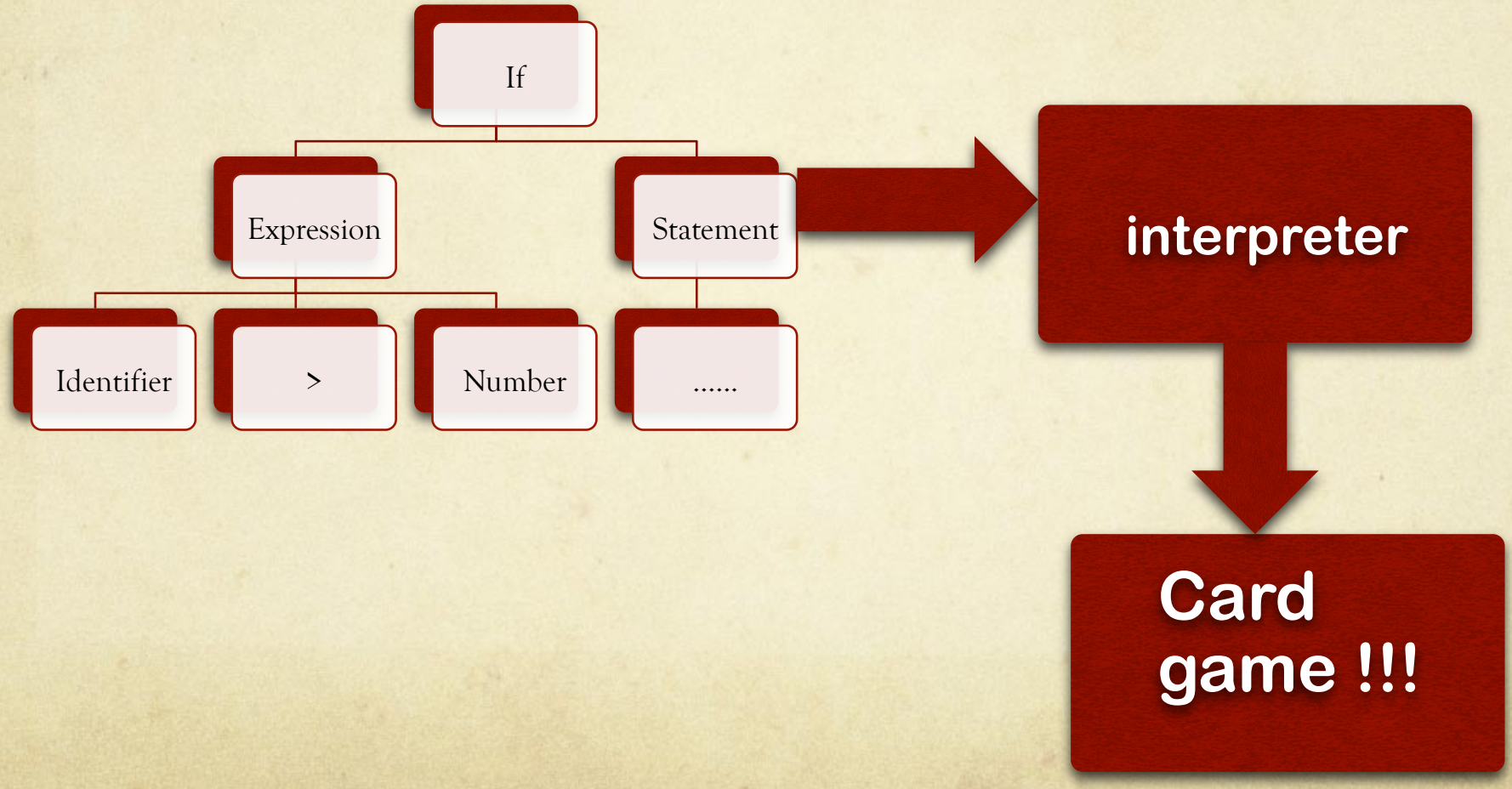
```
LocalStatement
        IF (Expression) { LocalStatementList }
        {$$ = con_s_if($3, $6)}
ExpressionREL
        ExpressionREL > ExpressionADD
        con_x_bin(X_GT, $1, $3);
```

Architecture



After pervious step, we can get AST

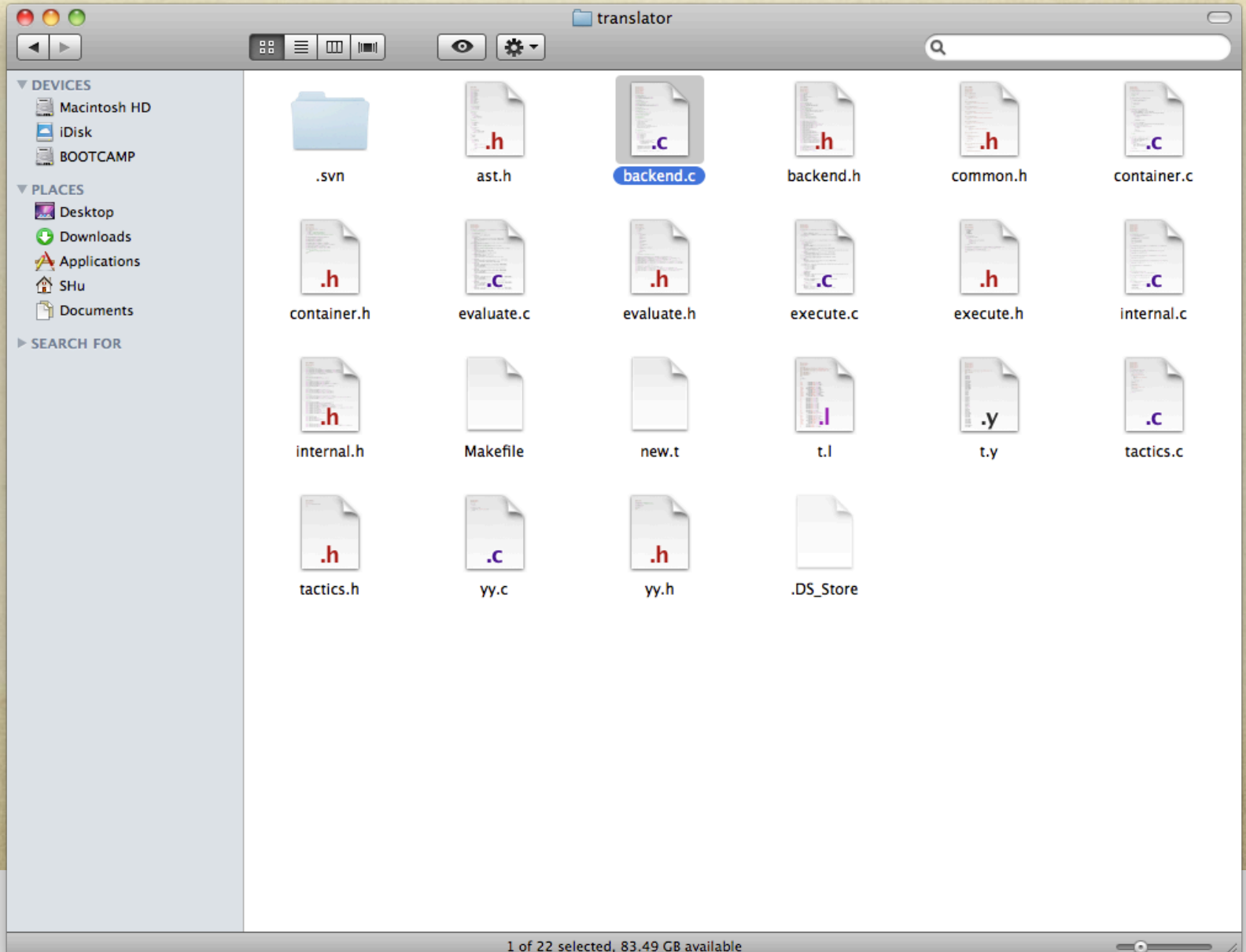
Architecture



System Integration

- Our Tactic language uses flex 2.5.35 to compile our lex file, and bison (GNU Bison) 2.3 to compile our yacc file.
- We use C in the back end to implement our interpreter.
- We use a SVN version control system provided by Google to develop our system.
- We use a makefile to compile our system to make all the components work together

System Integration



Test

- Unit Testing
- Regression Testing
- Integrated Testing

Demo time

Summary

The Tactics language making the design process simple

By providing data type, such as enum, container. Useful for programmer