Programming as collaborative reference

Oleg Kiselyov Chung-chieh Shan

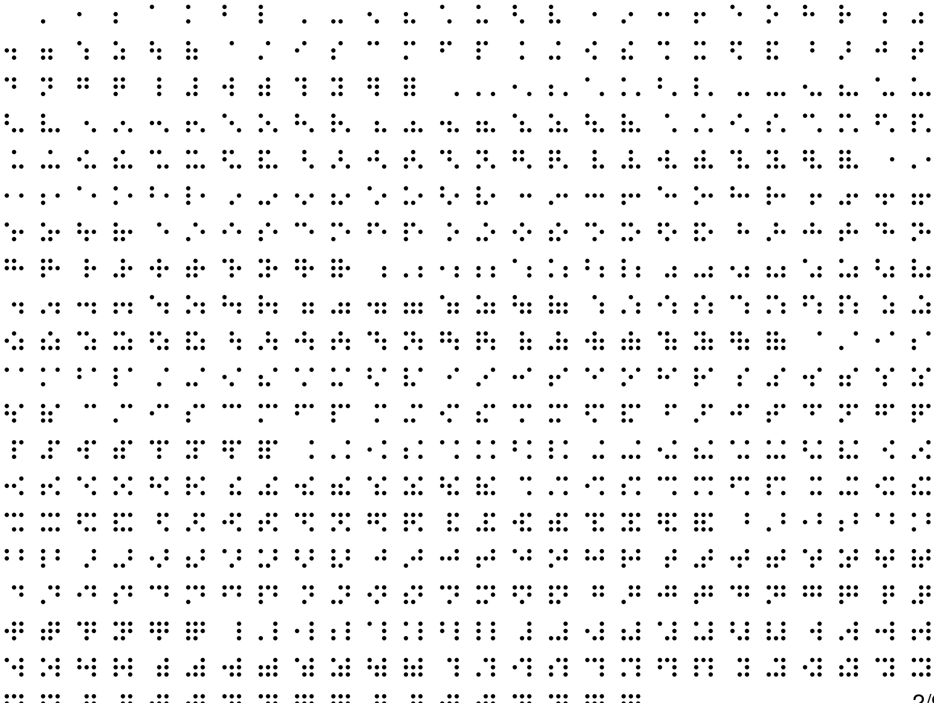
28 January 2012

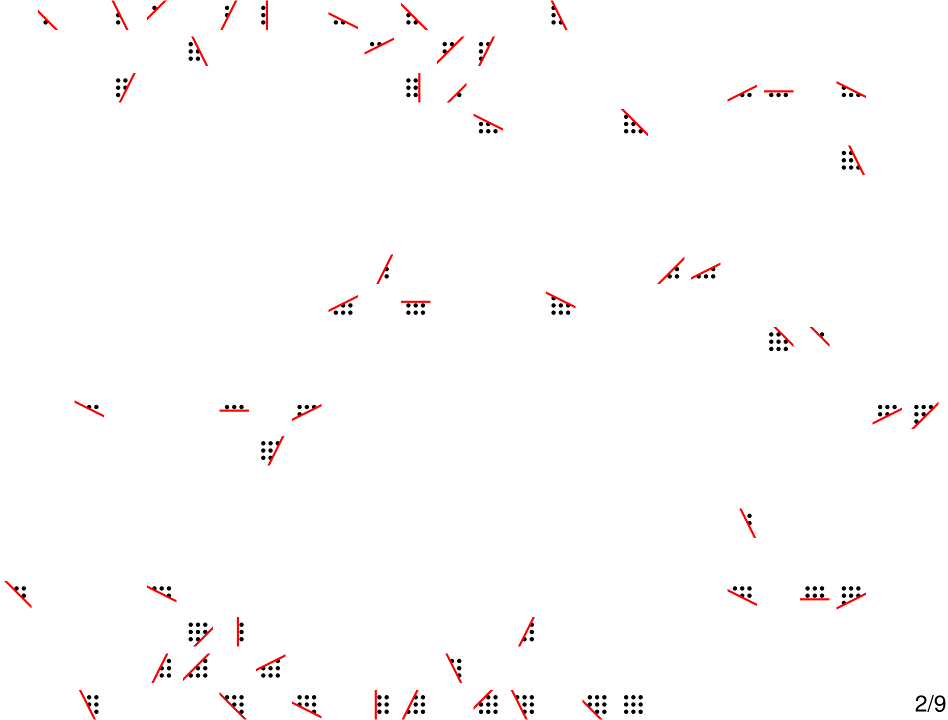
1 Programming 3 as 2 collaborative reference

4

Oleg Kiselyov Chung-chieh Shan

28 January 2012





Pragmatics

Communication bottleneck: So many meanings, so little time.

the president
him
Can everyone hear me?

Pragmatics

Communication bottleneck: So many meanings, so little time.

the president
him
Can everyone hear me?

We convey precise meanings flexibly:

- **use context** (Kaplan, Grice, . . .)
- **exchange feedback** (Clark & Wilkes-Gibbs, . . .)

"There are two aspects pertaining to referencing:
what to refer to and how to refer to it."
—Lopes, Dourish, Lorenz, Lieberherr (2003)

Pragmatics

Communication bottleneck: So many meanings, so little time.

the president
him
Can everyone hear me?

We convey precise meanings flexibly:

- **use context** (Kaplan, Grice, . . .)
  scope, type inference, overloading resolution, . . .
- **exchange feedback** (Clark & Wilkes-Gibbs, . . .)
  identifier completion, continuous compilation, . . .

“There are two aspects pertaining to referencing:
what to refer to and how to refer to it.”
—Lopes, Dourish, Lorenz, Lieberherr (2003)

Pragmatics

Communication bottleneck: So many meanings, so little time.

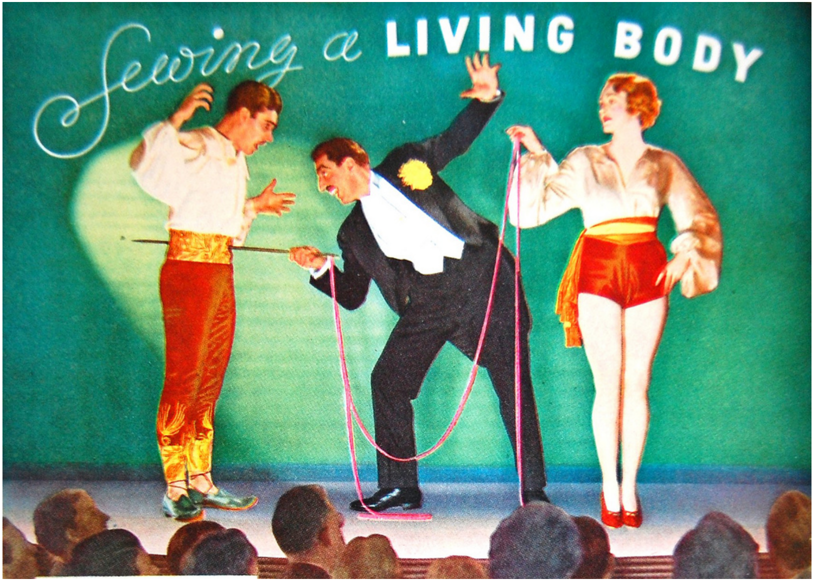
the president
him
Can everyone hear me?

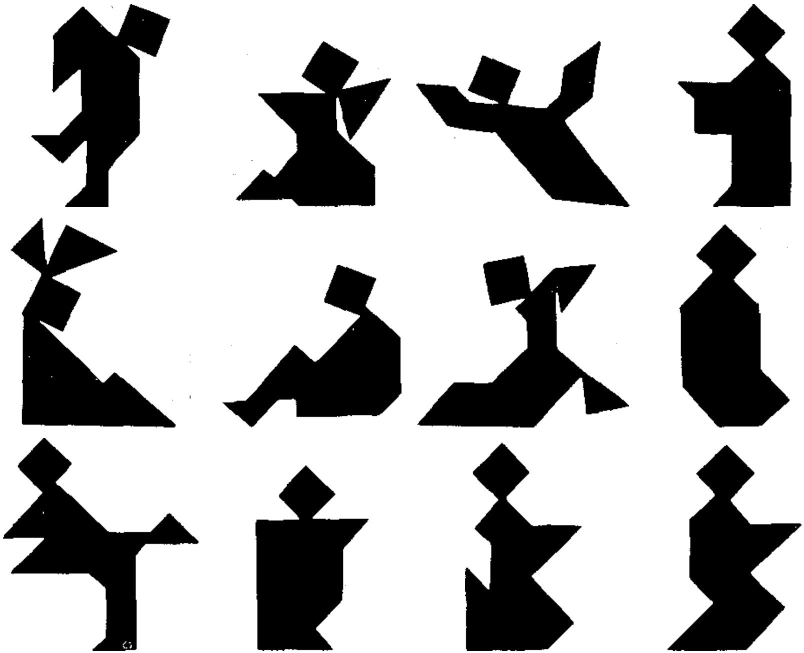
We convey precise meanings flexibly:

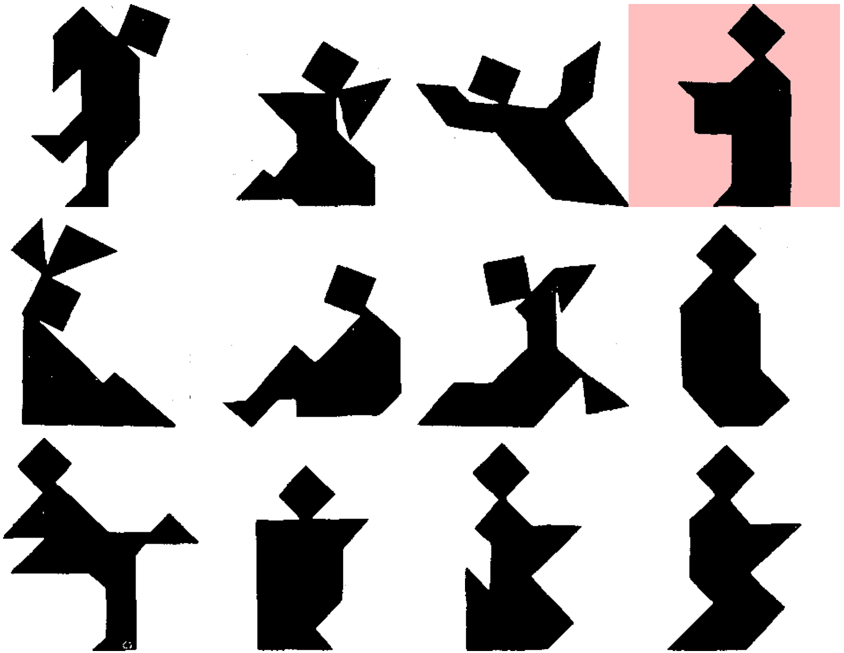
- **use context** (Kaplan, Grice, . . .)
  scope, type inference, overloading resolution, . . .
- **exchange feedback** (Clark & Wilkes-Gibbs, . . .)
  identifier completion, continuous compilation, . . .

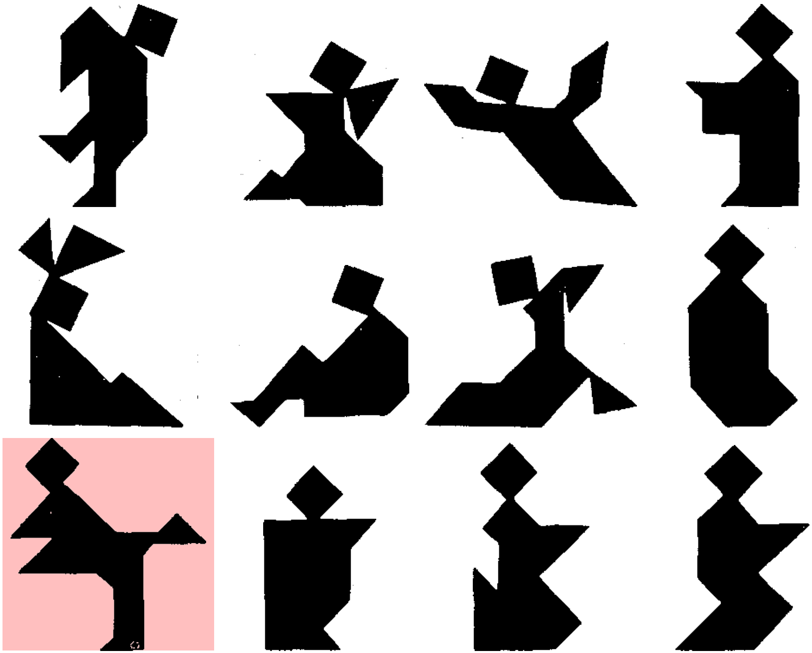
Herbert H. Clark and Deanna Wilkes-Gibbs. 1986.
"Referring as a collaborative process." *Cognition* 22(1):1–39.

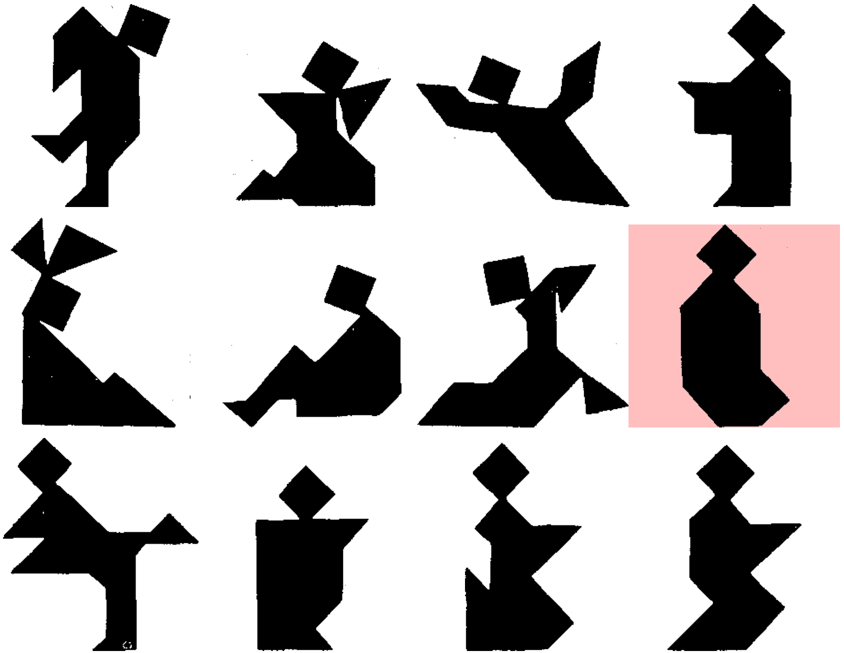
Collaborative reference

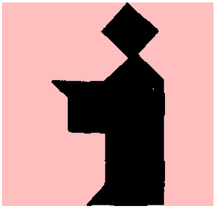








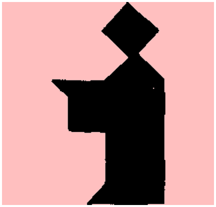




Interactive, not literary.

A: *the guy reading with, holding his book to the left.*
B: *Okay, kind of standing up?*
A: *Yeah.*
B: *Okay.*

Context and feedback!



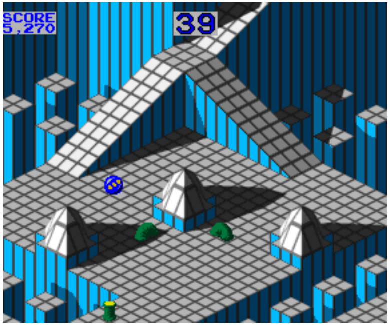
Interactive, not literary.

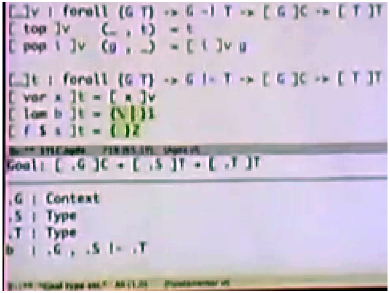
A: *the guy reading with, holding his book to the left.*
B: *Okay, kind of standing up?*
A: *Yeah.*
B: *Okay.*

Context and feedback!

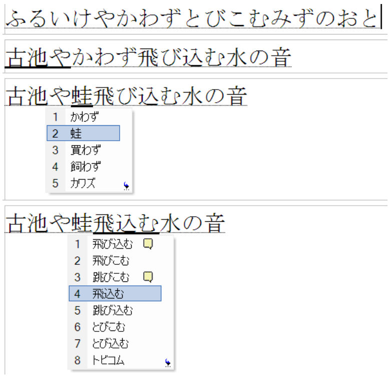
Marble madness

Marble madness





```
[_]v : forall {G T} -> G -| T -> [ G ]C -> [ T ]T
[ top ]v    (_ , t)  = t
[ pop i ]v  (g , _)  = [ i ]v g

[_]t : forall {G T} -> G |- T -> [ G ]C -> [ T ]T
[ var x ]t = [ x ]v
[ lam b ]t = (\ [ ])1
[ f $ s ]t = ( )2
[illegible]
Goal: [ .G ]C * [ .S ]T * [ .T ]T

.G : Context
.S : Type
.T : Type
b  : .G , .S |- .T

[illegible]
```



ふるいけやかわずとびこむみずのおと

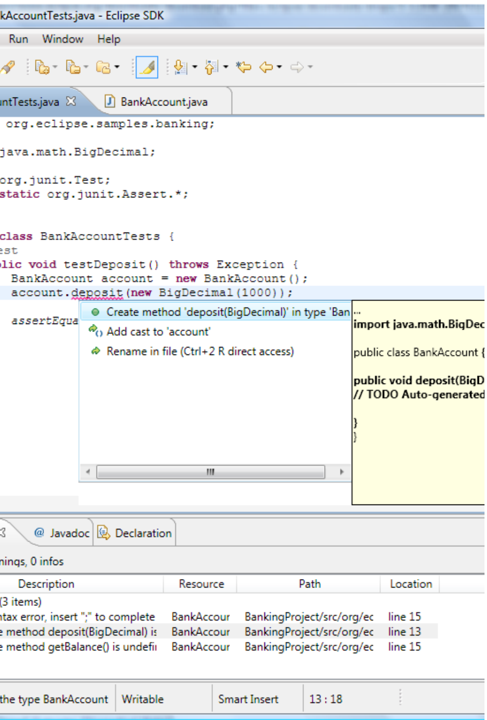
古池やかわず飛び込む水の音

古池や蛙飛び込む水の音

1 かわず
2 蛙
3 買わず
4 飼わず
5 カワズ

古池や蛙飛込む水の音

1 飛び込む
2 飛びこむ
3 跳びこむ
4 飛込む
5 跳び込む
6 とびこむ
7 とび込む
8 トビコム



```
 org.eclipse.samples.banking;

java.math.BigDecimal;

org.junit.Test;
static org.junit.Assert.*;


class BankAccountTests {
est
lic void testDeposit() throws Exception {
  BankAccount account = new BankAccount();
  account.deposit(new BigDecimal(1000));

  assertEqua
```

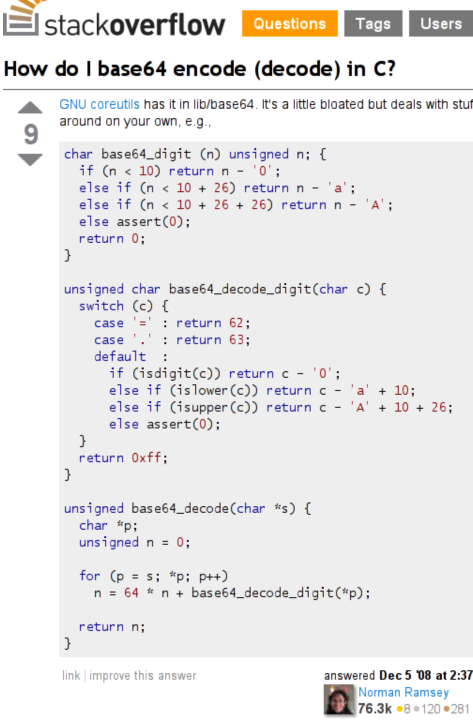
Create method 'deposit(BigDecimal)' in type 'Ban
Add cast to 'account'
Rename in file (Ctrl+2 R direct access)

```
...
import java.math.BigDec

public class BankAccount {

public void deposit(BigD
// TODO Auto-generated

}
}
```

nings, 0 infos

Description	Resource	Path	Location
(3 items)			
ntax error, insert ";" to complete	BankAccour	BankingProject/src/org/ec	line 15
e method deposit(BigDecimal) is	BankAccour	BankingProject/src/org/ec	line 13
e method getBalance() is undefi	BankAccour	BankingProject/src/org/ec	line 15

the type BankAccount | Writable | Smart Insert | 13 : 18



How do I base64 encode (decode) in C?

9

GNU coreutils has it in lib/base64. It's a little bloated but deals with stuf
around on your own, e.g.,

```
char base64_digit (n) unsigned n; {
  if (n < 10) return n - '0';
  else if (n < 10 + 26) return n - 'a';
  else if (n < 10 + 26 + 26) return n - 'A';
  else assert(0);
  return 0;
}

unsigned char base64_decode_digit(char c) {
  switch (c) {
    case '=' : return 62;
    case '.' : return 63;
    default  :
      if (isdigit(c)) return c - '0';
      else if (islower(c)) return c - 'a' + 10;
      else if (isupper(c)) return c - 'A' + 10 + 26;
      else assert(0);
  }
  return 0xff;
}

unsigned base64_decode(char *s) {
  char *p;
  unsigned n = 0;

  for (p = s; *p; p++)
    n = 64 * n + base64_decode_digit(*p);

  return n;
}
```

link | improve this answer

answered Dec 5 '08 at 2:37
Norman Ramsey
76.3k ●8 ●120 ●281

Tools:

- Logic!
- Meta!
- Types!
- Monads!

COREF demo.

Tools:

- Logic!
- Meta!
- Types!
- Monads!

COREF demo.

Targets:

- Overloading resolution: overlapping?
- Type inference: undecidable?

A principled distinction between what's said & what's meant.