Programming as collaborative reference

Oleg Kiselyov    Chung-chieh Shan

28 January 2012
Programming as collaborative reference

Oleg Kiselyov    Chung-chieh Shan

28 January 2012
Communication bottleneck: So many meanings, so little time.

*the president*

*him*

*Can everyone hear me?*
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.”

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.”
**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, . . .

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.
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.
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!
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

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

1. かわず
2. 鳥
3. 吹わず
4. 鉄らず
5. カリス

古池や蛙飛び込む水の音

1. 飛び込む
2. 飛びこむ
3. 吹込まれ
4. 飛び込む
5. 跳び込む
6. とびたる
7. とび込む
8. ですとこ

Goal: \text{[G]}C + \text{[S]}T + \text{[T]}T
How do I base64 encode (decode) in C?

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

```c
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
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.