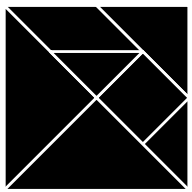
Clafer type system and attributes



Jimmy Liang and Kacper Bak
Generative Software Development Lab
University of Waterloo

GSD Lab Workshop. February 7, 2012

Recent Progress

Many people (Bo - Android)

Defaults (Leo)

Clafer wiki (Kacper)

Type system (Jimmy)

Translator (Jimmy and Kacper)

Intermediate representation - XML (Jimmy)

Test suite (Michal and Kacper)

Visualization (Seyi)

Usability Evaluation (Dina)

Multiobjective optimization (Bo, Derek)

Lightweight methodology (Krzysztof and Michal)

Applications (Financial domain - Marko, Security - M)

Type System

Motivation

Why does Clafer need a type system?

Type check

Free and automatic sanity check.

```
abstract Y
    x : string
    y : int
    [x + y = 3]
```

Clafer reports that “+” cannot be applied on *x* and *y*.

Semantics

The type of an expression affects the semantics.

abstract CentenarianClub
    memberAge : integer *

What does the expression *memberAge*?

Semantics case 1

abstract CentenarianClub
    memberAge : integer *
        [memberAge $\geqslant$ 100]

The expression *memberAge* refers to its integer value since we are doing an integer inequality.

Semantics case 2

```
abstract CentenarianClub
    memberAge : integer *
    [#memberAge = 20]
```

The expression *memberAge* refers to the *Clafer*, not the integer.

Semantics OO-analogy

```
class MemberAge {
    int value;
}
class CentenarianClub {
    MemberAge[] memberAge;
}
```

The expression *memberAge* can refer to

- the *MemberAge* object(s)
- or its *value* field

The inferred type of the expression implicitly determines which of the two case applies.

Type system

A Clafer model consists of two parts.

- Clafer definitions
- Constraints

The type system performs two tasks in parallel.

- Type check the expressions in the constraints.
- Infer the types of expressions in the constraints.

Notation & Definition 1

Definition

"::" is shorthand for "is type".

example: "$x :: \mathit{integer}$" is read as "x is type $\mathit{integer}$".

Definition

"$\vdash$" is shorthand for "entails".

example: "$\Gamma \vdash x :: \mathit{integer}$" is read as "$\Gamma$ entails $x :: \mathit{integer}$".
Sometimes it is clearer to read it as "$x :: \mathit{integer}$ given Γ".

Notation & Definition 2

Definition

The letter "*x*" is a Clafer reference.

In the expression below, "*speed*" is a Clafer reference.

```
[speed > 80]
```

Notation & Definition 3

Definition

The letters "E, F, G" are expressions.

2 leaf expressions: "*speed*" and "80".
1 super expression: "*speed* $>$ 80".

[speed $>$ 80]

Less frequent notations will be explained as they come.

Notation & Definition 4

Definition

A type environment (Γ) is a mapping from Clafer definition to the type of its value.

```
abstract Y : string
    a : integer
    b

X : Y
```

$$\Gamma = \{Y :: string, \quad a :: integer, \quad b :: clafer, \quad X :: string\}$$

Type rule

$$\text{name of rule}\ \frac{\textit{statementA} \qquad \textit{statementB}}{\textit{statementC}}$$

If A and B holds then C follows.

The type system is specified through a series of type rules.

Clafer type rule 1

$$\text{intconst}\ \frac{}{\Gamma \vdash \mathbb{INTEGER} :: \textit{integer}}$$

Clafer type rule 2

$$\text{eq}\ \frac{\Gamma \vdash E :: \tau \qquad \Gamma \vdash F :: \tau}{\Gamma \vdash E = F :: \mathit{boolean}}$$

Clafer type rule 3

Can we prove that the following model passes type checking?
What is the type of each expression?

```
abstract Y
    [0 = 1]
```

$$\Gamma = \{Y :: \mathit{clafer}\}$$

Clafer type rule 4

$$\Gamma = \{Y :: \mathit{clafer}\}$$

Proof.

$$\text{eq}\ \frac{\text{intconst}\ \dfrac{}{\Gamma \vdash\ 0 :: \mathit{integer}} \qquad \text{intconst}\ \dfrac{}{\Gamma \vdash\ 1 :: \mathit{integer}}}{\Gamma \vdash 0 = 1 :: \mathit{boolean}}$$

Clafer type rule 5

$$\text{value}\ \frac{(x :: \tau) \in \Gamma}{\Gamma \vdash x :: \tau}$$

Clafer type rule 6

Prove that the following model is type correct.

```
abstract Y
    a : integer
    [a = 1]
```

$$\Gamma = \{Y :: clafer, \quad a :: integer\}$$

Clafer type rule 7

$$\Gamma = \{Y :: \mathit{clafer}, \quad a :: \mathit{integer}\}$$

Proof.

$$\text{eq}\ \frac{\text{value}\ \dfrac{(a :: \mathit{integer}) \in \Gamma}{\Gamma \vdash\ a :: \mathit{integer}} \qquad \text{intconst}\ \dfrac{}{\Gamma \vdash\ 1 :: \mathit{integer}}}{\Gamma \vdash a = 1 :: \mathit{boolean}}$$

Clafer type rule 8

$$\text{clafer}\ \frac{}{\Gamma \vdash x :: \mathit{clafer}}$$

Clafer type rule 9

Prove that the following model is type correct.

```
abstract Y
    a : integer
    b
    [a = b]
```

$$\Gamma = \{Y :: clafer, \quad a :: integer, \quad b :: clafer\}$$

Clafer type rule 10

$$\Gamma = \{Y :: \mathit{clafer}, \quad a :: \mathit{integer}, \quad b :: \mathit{clafer}\}$$

Proof.

$$\text{eq}\ \frac{\text{clafer}\ \dfrac{}{\Gamma \vdash\ a :: \mathit{clafer}} \qquad \text{clafer}\ \dfrac{}{\Gamma \vdash\ b :: \mathit{clafer}}}{\Gamma \vdash a = b :: \mathit{boolean}}$$

Clafer type rule precedence

```
abstract Y
    a : integer
    b : integer
    [a = b]
```

Integer equality or clafer equality?

Clafer type rule casting 1

```
abstract Y
    a : real
    b : integer
    [a = b]
```

Clafer type rule casting 2

$$\text{eqcast1}\ \frac{\Gamma \vdash E :: \mathit{real} \qquad \Gamma \vdash F :: \mathit{integer}}{\Gamma \vdash E = F :: \mathit{boolean}}$$

$$\text{eqcast2}\ \frac{\Gamma \vdash E :: \mathit{integer} \qquad \Gamma \vdash F :: \mathit{real}}{\Gamma \vdash E = F :: \mathit{boolean}}$$

Attributes

Conclusion

Conclusion

blahbla

blahbla

blahbla

blahbla

Thanks for listening!

Questions?

clafer.org