

Inferencia de facetas de declasificación

Propuesta de solución

Matías Meneses C.

1. Descripción y motivación del problema
2. Consideraciones importantes
3. Propuesta: Proceso iterativo de inferencia asistida
4. Titleformats
5. Elements
6. Conclusion

Descripción y motivación del problema

- Se quiere realizar inferencia de facetas públicas para facilitar el trabajo al programador.

- Se quiere realizar inferencia de facetas públicas para facilitar el trabajo al programador.
- Dado que Dart provee la inferencia de facetas privadas, el problema se denomina "Inferencia de facetas públicas (o de declasificación) dadas las facetas privadas".

Consideraciones importantes

Consideraciones importantes

La inferencia global es un problema no decidible, y la presencia de subtyping no mejora la situación.

```
class Person {  
    Person foo(Person a, Person b) {  
        return a.bar(b);  
    }  
  
    Person bar(Person a) {  
        return a.foo(a, this);  
    }  
}
```

Consideraciones importantes

La inferencia global es un problema no decidible, y la presencia de subtyping no mejora la situación.

```
class Person {  
    @??? Person foo(@PersonBar Person a, @PersonFoo Person b) {  
        return a.bar(b);  
    }  
  
    @??? Person bar(@PersonFoo Person a) {  
        return a.foo(a, this);  
    }  
}
```


Entonces, mejor le pedimos al usuario que especifique las facetas de declasificación de la firma de los métodos...

Propuesta: Proceso iterativo de
inferencia asistida

Propuesta: Proceso iterativo de inferencia asistida

- Paso 1: Inferir considerando solo las facetas de la firma de un metodo que están definidas.

Propuesta: Proceso iterativo de inferencia asistida

- Paso 1: Inferir considerando solo las facetas de la firma de un metodo que están definidas.
- Pasos sucesivos: Utilizando las facetas inferidas en el paso 1 y las modificaciones del usuario, refinar las facetas cuando sea pertinente.

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    void foo(Person a, Person b) {  
        a.bar(b);  
    }  
  
    void bar(Person p) {  
        p.baz(p);  
    }  
  
    void baz(Person p) {  
        print(p);  
    }  
}
```

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    void foo(Person a, Person b) {  
        a.bar(b);  
    }
```

```
    void bar(Person p) {  
        p.baz(p);  
    }
```

```
    void baz(Person p) {  
        print(p);  
    }  
}
```



Inferir facetas no definidas.

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    @void void foo(@PersonBar Person a,  
        @Person Person b) {  
        a.bar(b);  
    }  
  
    @void void bar(@PersonBaz Person p) {  
        p.baz(p);  
    }  
  
    @void void baz(@Person Person p) {  
        print(p);  
    }  
}
```

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    @void void foo(@PersonBar Person a,  
        @Person Person b) {  
        a.bar(b);  
    }  
  
    @void void bar(@PersonBaz Person p) {  
        p.baz(p);  
    }  
  
    @void void baz(@Person Person p) {  
        print(p);  
    }  
}
```



Flujo no permitido. Refinar facetas.

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    @void void foo(@PersonBar Person a,  
        @PersonBaz Person b) {  
        a.bar(b);  
    }  
  
    @void void bar(@Person Person p) {  
        p.baz(p);  
    }  
  
    @void void baz(@Person Person p) {  
        print(p);  
    }  
}
```

Propuesta: Proceso iterativo de inferencia asistida

```
class Person {  
    @void void foo(@PersonBar Person a,  
        @PersonBaz Person b) {  
        a.bar(b);  
    }  
}
```

```
@void void bar(@Person Person p) {  
    p.baz(p);  
}
```

```
@void void baz(@Person Person p) {  
    print(p);  
}  
}
```



Flujo no permitido. Refinar facetas.

Propuesta: Proceso iterativo de inferencia asistida

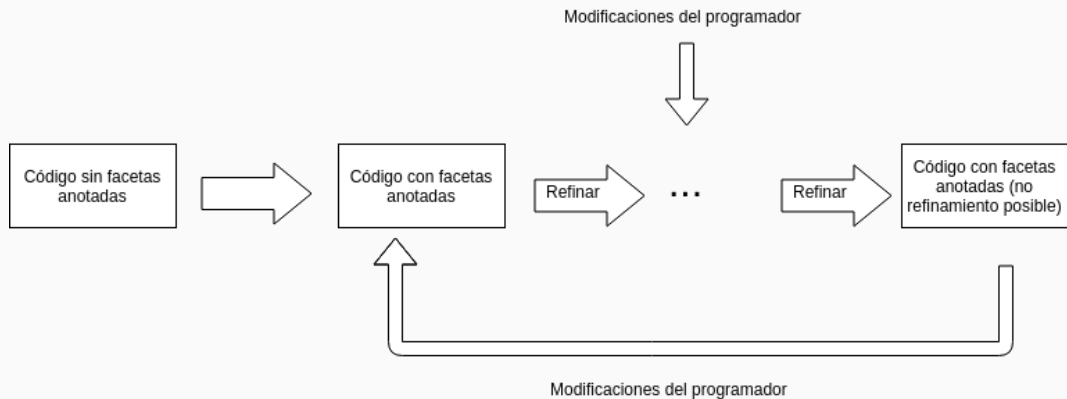
```
class Person {  
    @void void foo(@PersonBar Person a,  
        @Person Person b) {  
        a.bar(b);  
    }  
}
```

```
@void void bar(@Person Person p) {  
    p.baz(p);  
}
```

No hay errores :)

```
@void void baz(@Person Person p) {  
    print(p);  
}  
}
```

Propuesta: Proceso iterativo de inferencia asistida



Primero, se extiende el lenguaje base de Type-based relaxed noninterference para incluir referencias, instrucciones condicionales y secuencias de instrucciones.

```
e ::= v | e;e | e = e | if e then e else e | while e do e |  
      e.m(e) | m(x)e  
v ::= DV | x | [z: U => list(m(x)e)]  
U ::= 0 | TVar | void  
0 ::= Obj(TVar). [list(m: U -> U)]
```

x variable

m method label

U public facet

DV Dart primitive value

Luego, se escriben las reglas de inferencia, que generan un set de constraints para cada tipo de expresión.

Luego, se escriben las reglas de inferencia, que generan un set de constraints para cada tipo de expresión.

$$\begin{array}{c} \text{(seq)} \\ \hline \frac{\Gamma, M, pc, pt \vdash e1 : t1 \mid C1 \quad \Gamma, M, pc, pt \vdash e1 : t1 \mid C2}{\Gamma, M, pc, pt \vdash e1; e2 : t2 \mid C2 \cup C1} \end{array}$$

Luego, se escriben las reglas de inferencia, que generan un set de constraints para cada tipo de expresión.

(if)

$$\frac{\begin{array}{l} \Gamma, M, pc, pt \vdash e1 : t1 \mid C1 \\ \Gamma, M, pc1, pt \vdash e2 : t2 \mid C2 \\ \Gamma, M, pc1, pt \vdash e3 : t3 \mid C3 \end{array}}{\Gamma, M, pc, pt \vdash \text{if } e1 \text{ then } e2 \text{ else } e3 : t \mid \{t2 <: t\} \cup \{t3 <: t\} \cup C3 \cup C2 \cup \{t1 <: pc1\} \cup C1}$$

Sections group slides of the same topic

```
\section{Elements}
```

for which metropolis provides a nice progress indicator ...

Titleformats

metropolis supports 4 different titleformats:

- Regular
- Smallcaps
- allsmallcaps
- ALLCAPS

They can either be set at once for every title type or individually.

This frame uses the `smallcaps` titleformat.

Potential Problems

Be aware, that not every font supports small caps. If for example you typeset your presentation with pdfTeX and the Computer Modern Sans Serif font, every text in `smallcaps` will be typeset with the Computer Modern Serif font instead.

This frame uses the `allsmallcaps` titleformat.

Potential problems

As this titleformat also uses smallcaps you face the same problems as with the `smallcaps` titleformat. Additionally this format can cause some other problems. Please refer to the documentation if you consider using it.

As a rule of thumb: Just use it for plaintext-only titles.

This frame uses the `allcaps` titleformat.

Potential Problems

This titleformat is not as problematic as the `allsmallcaps` format, but basically suffers from the same deficiencies. So please have a look at the documentation if you want to use it.

Elements

The theme provides sensible defaults to
`\emph{emphasize}` text, `\alert{accent}` parts
or show `\textbf{bold}` results.

becomes

The theme provides sensible defaults to emphasize text, **accent** parts or show bold results.

- Regular
- Italic
- SmallCaps
- Bold
- Bold Italic
- Bold SmallCaps
- Monospace
- Monospace Italic
- Monospace Bold
- Monospace Bold Italic

Items

- Milk
- Eggs
- Potatos

Enumerations

1. First,
2. Second and
3. Last.

Descriptions

PowerPoint Meeh.
Beamer Yeeeha.

- This is important

- This is important
- Now this

- This is important
- Now this
- And now this

- This is really important
- Now this
- And now this

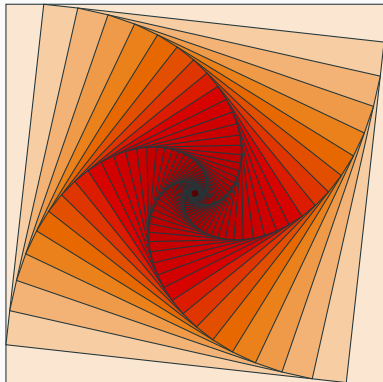


Figure 1: Rotated square from texample.net.

Table 1: Largest cities in the world (source: Wikipedia)

City	Population
Mexico City	20,116,842
Shanghai	19,210,000
Peking	15,796,450
Istanbul	14,160,467

Three different block environments are pre-defined and may be styled with an optional background color.

Default
Block content.

Alert
Block content.

Example
Block content.

Default

Block content.

Alert

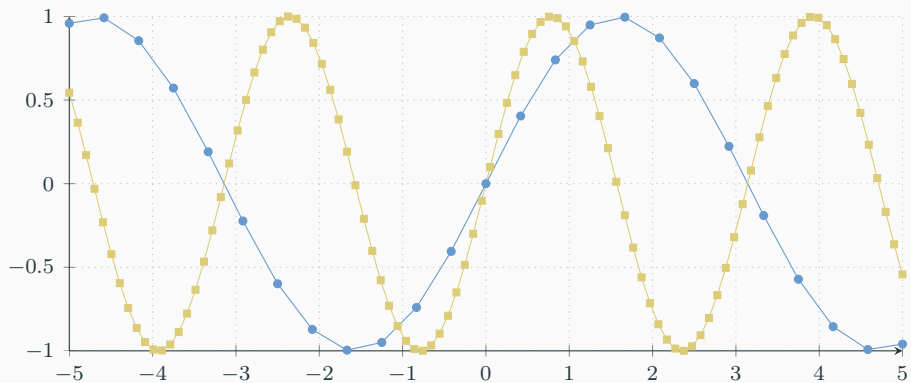
Block content.

Example

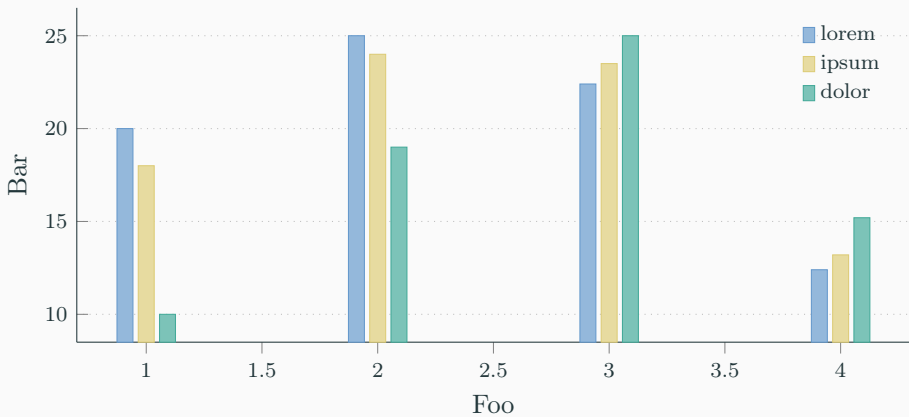
Block content.

$$e = \lim_{n \rightarrow \infty} \left(1 + \frac{1}{n}\right)^n$$

Line plots



Bar charts



Veni, Vidi, Vici

metropolis defines a custom beamer template to add a text to the footer. It can be set via

```
\setbeamertemplate{frame footer}{My custom footer}
```

Some references to showcase `[allowframebreaks]` [4, 2, 5, 1, 3]

Conclusion

Get the source of this theme and the demo presentation from

`github.com/matze/mtheme`

The theme itself is licensed under a Creative Commons Attribution-ShareAlike 4.0 International License.








Questions?

Sometimes, it is useful to add slides at the end of your presentation to refer to during audience questions.

The best way to do this is to include the `appendixnumberbeamer` package in your preamble and call `\appendix` before your backup slides.

metropolis will automatically turn off slide numbering and progress bars for slides in the appendix.

-  P. Erdős.
A selection of problems and results in combinatorics.
In Recent trends in combinatorics (Matrahaza, 1995), pages 1–6. Cambridge Univ. Press, Cambridge, 1995.
-  R. Graham, D. Knuth, and O. Patashnik.
Concrete mathematics.
Addison-Wesley, Reading, MA, 1989.
-  G. D. Greenwade.
The Comprehensive Tex Archive Network (CTAN).
TUGBoat, 14(3):342–351, 1993.

-  D. Knuth.
Two notes on notation.
Amer. Math. Monthly, 99:403–422, 1992.
-  H. Simpson.
Proof of the Riemann Hypothesis.
preprint (2003), available at <http://www.math.drofnats.edu/riemann.ps>,
2003.