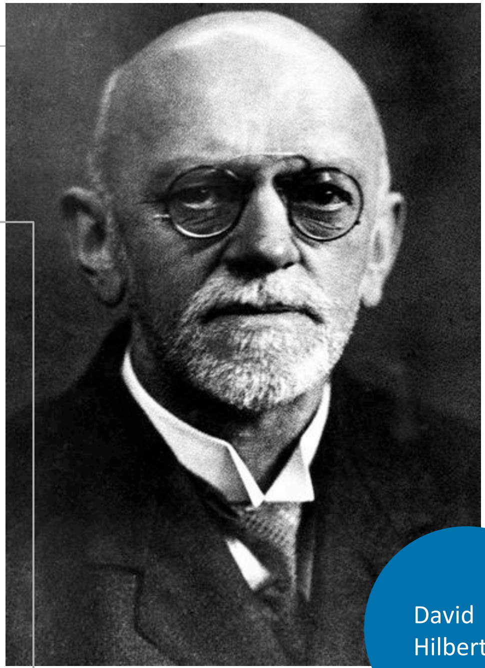
Type Inference in Multirate Faust Is Undecidable

Pierre Jouvelot

MINES ParisTech, PSL Research University

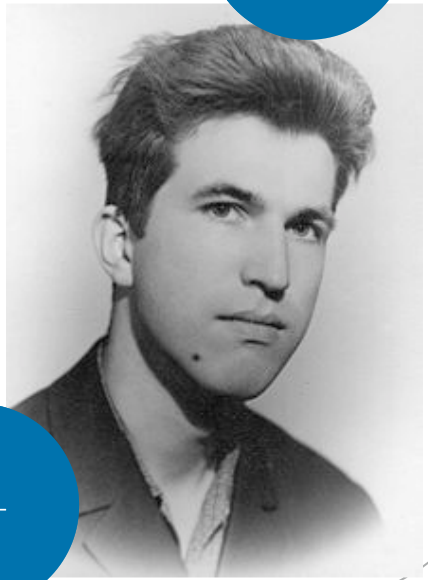
Hilbert's Problems

- 23 key problems for the 20th century (Paris, 1900)
- Problem examples:
  - 1st – Prove the Continuum Hypothesis. (?)
  - 13th – Solve 7-th degree equation using continuous functions of two parameters(~)
  - 17th – Express a nonnegative rational function as quotientof sums of squares (P)
- 10th problem:
  Find an algorithm to determine whether a given polynomial Diophantine equation with integer coefficients has an integer.
- The 10th problem is undecidable (Matiyasevich, 1970)



(1)

David Hilbert



Yuri Matiya-sevich

(2)

(1) http://www.larousse.fr/encyclopedie/personnage/Hilbert/123908 (2) http://en.wikipedia.org/wiki/Yuri_Matiyasevich#mediaviewer/File:Yuri_Matiyasevich._Portrait_1969.jpg

Type Inference as Hilbert's 10^{th} Problem

- The Multirate Faust Type Inference Problem:

  Given a Faust expression E, decide whether there exists a type for E consistent with Orlarey-Jouvelot's Faust multirate static semantics.

- Theorem (UFTI): *Multirate Faust type inference is undecidable.*

- Idea: Encode the type inference problem as a Hilbert's 10^{th} problem.

- Diophantine equations (DE):
  - $ax+by = 1$
  - $x^n+y^n = z^n$ (Théorème de Fermat)
  - $2x^2y^3+4x^5+7 = 0$

- Finding an algorithm for solving DEs is impossible (undecidable)

Faust Encoding (No Subtyping)

- Faust Vector API :
  - New vectors, via `vectorize` : $(t^f, int[n,n]^{f'}) \rightarrow (vector_n(t)^f)$
  - Vector access, via `[]` : $(vector_n(t)^f, int[0,n-1]^f) \rightarrow (t^f)$

- How to create unknowns (x, y)?

```
constantize = 0,_ : vectorize : !;

new_unknown = _ <: constantize,_ ;
```

- Remarks:
  - No output for `constantize`, but enforces inputs of type *b[n,n]*
  - Connector `<:` typing rule enforces equality between intermediate signals

Faust Encoding (No Subtyping) (contd.)

- How to create monomials (x^n) ?

```
power(x,0) = 1;
power(x,1) = x;
power(x,n) = x <: _,power(_,n-1) : *;
```

- Remarks:
  - No duplication of `x` authorized (we need to use _ as arguments)

- How to encode polynomials (x^3y)? Case of $p(x,y,z) = 2x^3y+yz^4+xyz+5$:

```
p(x,y,z) =
    (x,y,z) <:
        (power(_,3),_ ,! : *),
        (!,_,power(_,4) : *),
        ((_,_ : *),_ : *),
        5
    :> _;
```

Faust Encoding (No Subtyping) (contd.)

- How to encode tests (*p(x) = 0*)?

```
is_zero(p) = (0,1 : vectorize),p : [] : !;
```

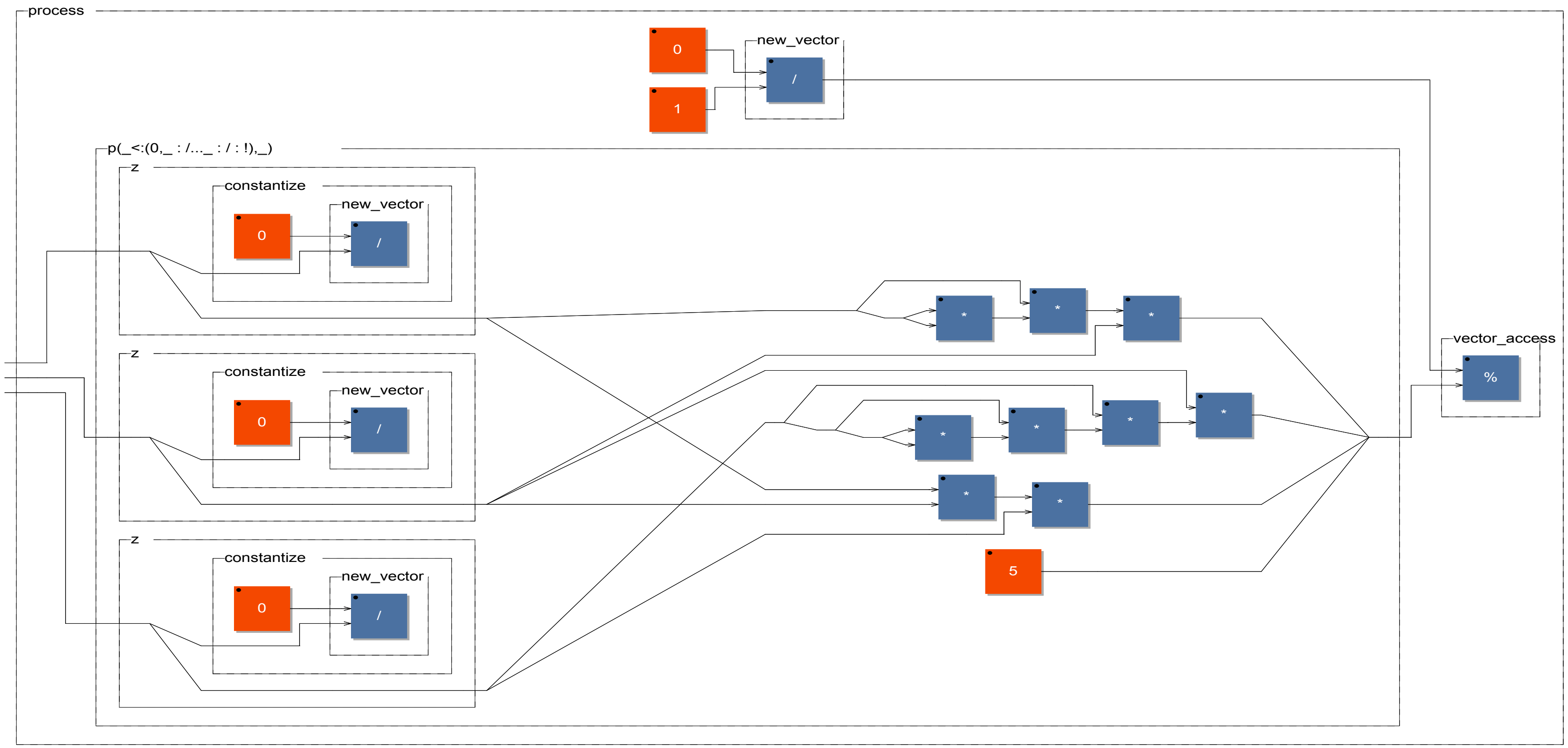
- How to encode Hilbert's 10th problem?

```
x = new_unknown;
y = new_unknown;
z = new_unknown;
```

-

```
process = is_zero(p(x,y,z));
```

Faust Encoding (No Subtyping) (end)



Handling Subtyping

- Flexibility induced by the subtyping rule (interval widening)
- Loss of constant property for signals
- Subtyping can remove constancy:
  - at top level… but no output!
  - when connecting constant signals to the inputs of polymorphic (either on *t* or on [*n*,*n*']) processors
- Hilbert encoding of signal processors without subtyping:
  - constant (e.g., `5`)
  - constant-preserving (e.g., `*`)
  - fully polymorphic (e.g., `vectorize` or `_`)
- `constant(E)` ensures that all the intermediate signals within `E` are constant:

  constant(E : E') = constant(E) <: constantize,constant(E')
  constant(E op E') = constant(E) op constant(E')
  constant(I) = I

- Remarks:
  - Hypothesis: connection between E and E' has arity one
  - `constant` can only be applied to scalar signals
  - Lemma: *If* E_i *is constant-preserving,* $constant(E_1 : E_2)$ *is too.*

Discussion

- Theorem UFTI grounds an intuition on the power of Faust type system
- Rates not necessary to prove undecidability
- Opportunities:
  - Language type extensions to handle user assertions
  - Other « undecidable constructs » are ok (null vectors?)
- Future work:
  - Use Faust/CoQ to formally prove the UFTI theorem
  - Do explicit input types kill the proof?

FEEVER Meeting – October 13,2014

Type Inference of Multirate Faust Is Undecidable

Pierre Jouvelot

MINES ParisTech, PSL Research University

Hilbert's 10th Problem

- Le 21ème siècle : vers une société numérisée
- Du réel analogique au virtuel numérique
- Textes, sons, photos, films… voire même des objets !
- Comprendre la musique numérique…

Des octets pour la musique

(1)

(1) flickr / APM Alex