Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL

Jasmin C. Blanchette
Mathias Fleury
Dmitriy Traytel

max planck institut informatik
SAARLAND UNIVERSITY
SAARBRÜCKEN GRADUATE SCHOOL OF COMPUTER SCIENCE
Saarland Informatics Campus
VRIJE UNIVERSITEIT AMSTERDAM
ETH zürich

Motivation

- Jasmin needs ordinals for the transfinite Knuth-Bendix ordering
- Dmitriy wants nested multiset ordering

Multisets

Nested Multisets

Hereditary Multisets

Syntactic Ordinals

Signed Hereditary Multisets

Multisets

A non-empty set

```
typedef 'a multiset = {f :: 'a ⇒ nat. finite {x. f x > 0}}
```

Values are constructed by `{#}` **and** `add_mset`

A non-empty set

```
typedef 'a multiset = {f :: 'a ⇒ nat. finite {x. f x > 0}}
```

Values are constructed by `{#}` **and** `add_mset`

@Isabelle User: please use and extend
`$AFP/Nested_Multisets_Ordinals/Multiset_More`
(we slowly move the theorems to the distribution)

Cancellation Simprocs

- Simplify `add_mset a A + F = F + add_mset b (add_mset a B)`

  into `A = add_mset b B`

- Based on the simproc for natural numbers to handle

  `replicate_mset n a = {a} + {a} + ... + {a}`

  (n times)

Multisets

Syntactic Ordinals

Nested Multisets

Signed Hereditary Multisets

Hereditary Multisets

VU VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

ETH zürich

Nested Multisets

type variable

type constructor

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

keyword to define datatypes and derive theorems

constructors

VU VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

ETH zürich

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

Recursion allowed through bounded natural functor

VU VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

ETH zürich

Bounded Natural Functors

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

- BNF: `('a nmultiset, map_nmultiset, set_nmultiset,` $\aleph_0$`)`

type — **functorial action** — **natural transformation** — **upper bound on cardinality**

Bounded Natural Functors

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

- BNF: `('a nmultiset, map_nmultiset, set_nmultiset,` $\aleph_0$`)`

  - `'a nmultiset` ← **type**
  - `map_nmultiset` ← **functorial action**
  - `set_nmultiset` ← **natural transformation**
  - $\aleph_0$ ← **upper bound on cardinality**

- BNFs are closed under composition

Bounded Natural Functors

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

- BNF: `('a nmultiset, map_nmultiset, set_nmultiset,` $\aleph_0$`)`

  type — **functorial action** — **natural transformation** — **upper bound on cardinality**

- BNFs are closed under composition
- Datatypes and codatatypes are BNFs

VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

ETH zürich

Bounded Natural Functors

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

- BNF: `('a nmultiset, map_nmultiset, set_nmultiset,` $\aleph_0$`)`

  type — **functorial action** — **natural transformation** — **upper bound on cardinality**

- BNFs are closed under composition
- Datatypes and codatatypes are BNFs
- Some non-datatypes are also be BNFs, e.g., multisets

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

Induction principle:

```
⋀x. P (Elem x)
⋀NM. (⋀N. N ∈ set_multiset NM ⟹ P N) ⟹ P (MSet NM)
P N
```

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

Induction principle:

$$\frac{\bigwedge x.\ P\ (\mathsf{Elem}\ x) \qquad \bigwedge NM.\ (\bigwedge N.\ N \in \mathsf{set\_multiset}\ NM \Longrightarrow P\ N) \Longrightarrow P\ (\mathsf{MSet}\ NM)}{P\ N}$$

VRIJE UNIVERSITEIT AMSTERDAM max planck institut informatik ETH zürich

```
datatype 'a nmultiset =
    Elem 'a | MSet (('a nmultiset) multiset)
```

Induction principle:

$$\frac{\bigwedge x.\ P\ (\text{Elem}\ x) \qquad \bigwedge NM.\ (\bigwedge N.\ N \in \text{set\_multiset}\ NM \Longrightarrow P\ N) \Longrightarrow P\ (\text{MSet}\ NM)}{P\ N}$$

Allows to define recursive functions:

```
primrec depth where
  depth (Elem x) = 0
| depth (MSet M) =
    (let X = set (map_mset depth M) in
     if X = {} then 0 else Max X + 1)
```

Multisets

Nested Multisets

Hereditary Multisets

Syntactic Ordinals

Signed Hereditary Multisets

Hereditary Multisets

```
datatype 'a nmultiset =
  Elem 'a | MSet (('a nmultiset) multiset)
```

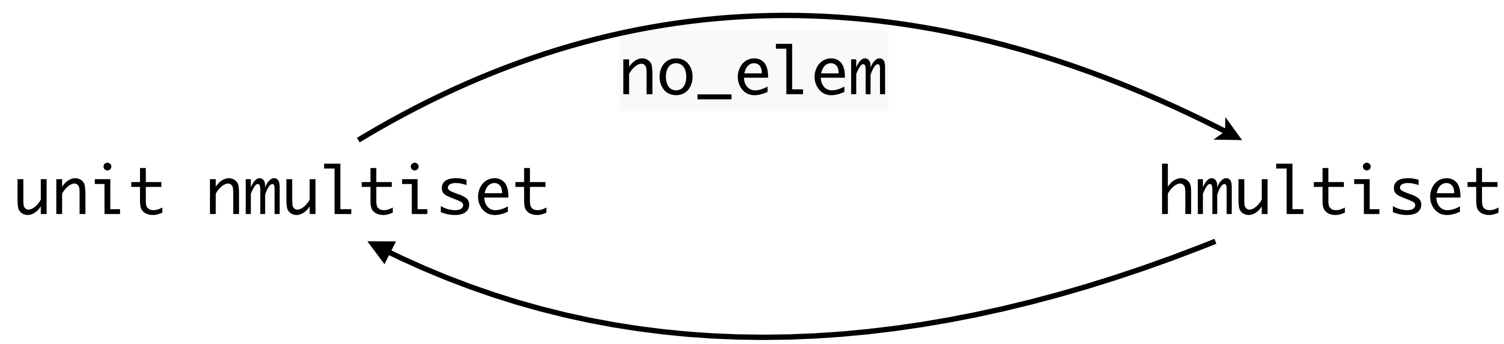
We often don't want `Elem`. Three options:

1. `'a = ∅`
   - `'a` cannot be empty in Isabelle

2. ```
   datatype hmultiset =
     HMSet (hmultiset multiset)
   ```
   - generates selector, induction principle, recursion scheme

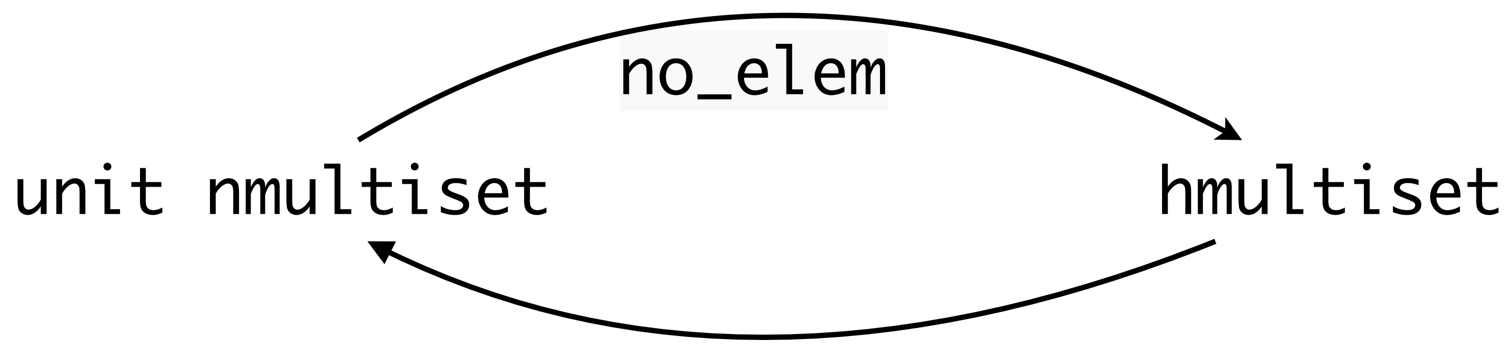
3. `typedef` **and** `no_elem` **predicate**
   - allows to lift definition via the Lifting tool
   ```

```
Abs_hmultiset (MSet M) = HMSet (map_mset Abs_hmultiset M)
```




```
Rep_hmultiset (HMSet M) = MSet (map_mset Rep_hmultiset M)
```

```
Abs_hmultiset (MSet M) = HMSet (map_mset Abs_hmultiset M)
```




```
Rep_hmultiset (HMSet M) = MSet (map_mset Rep_hmultiset M)
```

Lift definition via the Lifting and Transfer tool, e.g.:

```
A < B ⟷ Rep_hmultiset A < Rep_hmultiset B
```

Multisets

Nested Multisets

Hereditary Multisets

Syntactic Ordinals

Signed Hereditary Multisets

Syntactic Ordinals

Cantor normal form for the ordinals below ϵ_0

$$\alpha ::= \omega^{\alpha_1} \cdot c_1 + \cdots + \omega^{\alpha_n} \cdot c_n$$

$$\text{where } c_i \in \mathbb{N}^{>0} \text{ and } \alpha_1 > \cdots > \alpha_n$$

$$\alpha ::= \{ \underbrace{\alpha_1, \ldots, \alpha_1}_{c_1 \text{ occurrences}}, \ldots, \underbrace{\alpha_n, \ldots, \alpha_n}_{c_n \text{ occurrences}} \}$$

E.g.:

$$\{\} = 0 \qquad \{0\} = \{\{\}\} = \omega^0 = 1$$

$$\{1\} = \{\{\{\}\}\} = \omega^{\omega^0} = \omega \qquad \{\omega\} = \omega^\omega$$

Hessenberg addition [Ludwig and Waldmann]:

Definition 7 (Hessenberg Addition). Let $\oplus\colon \mathbf{O} \times \mathbf{O} \to \mathbf{O}$ be the following function:

– For $\alpha \in \mathbf{O} \setminus \{0\}$ we define:

$$0 \oplus 0 = 0$$
$$0 \oplus \alpha = \alpha$$
$$\alpha \oplus 0 = \alpha$$

– Let for natural numbers $m, m' \in \mathbb{N}^{>0}$, $n_1, \ldots, n_m$, $n'_1, \ldots, n'_{m'} \in \mathbb{N}^{>0}$, ordinals $b_1, \ldots, b_m$, $b'_1, \ldots, b'_{m'} \in \mathbf{O}$ such that $b_1 > b_2 > \cdots > b_m$ and $b'_1 > b'_2 > \cdots > b'_{m'}$,

$$\alpha = \sum_{i=1}^{m} (\omega^{b_i} \cdot n_i),\ \beta = \sum_{i=1}^{m'} (\omega^{b'_i} \cdot n'_i) \in \mathbf{O}$$

Isabelle:

```
A + B = HMSet (hmsetmset A + hmsetmset B)
```

Hessenberg multiplication [Ludwig and Waldmann]:

Definition 8 (Hessenberg Multiplication). Let $\odot \colon \mathbf{O} \times \mathbf{O} \to \mathbf{O}$ be the following function:

- For $\alpha \in \mathbf{O} \setminus \{0\}$ we define:

$$0 \odot 0 = 0$$
$$0 \odot \alpha = 0$$
$$\alpha \odot 0 = 0$$

- Let for $m, m' \in \mathbb{N}^{>0}, n_1, \dots, n_m, n'_1, \dots, n'_{m'} \in \mathbb{N}^{>0}, b_1, \dots, b_m, b'_1, \dots, b'_{m'} \in \mathbf{O}$ such that $b_1 > b_2 > \dots > b_m$ and $b'_1 > b'_2 > \dots > b'_{m'}$,

$$\alpha = \sum_{i=1}^{m} (\omega^{b_i} \cdot n_i),\ \beta = \sum_{j=1}^{m'} (\omega^{b'_j} \cdot n'_j)$$

We define then

$$\alpha \odot \beta = \bigoplus_{i=1}^{m} \bigoplus_{j=1}^{m'} \left(\omega^{b_i \oplus b'_j} \cdot (\mathrm{coeff}(\alpha,\, b_i) \cdot \mathrm{coeff}(\beta,\, b'_j))\right)$$

Isabelle:

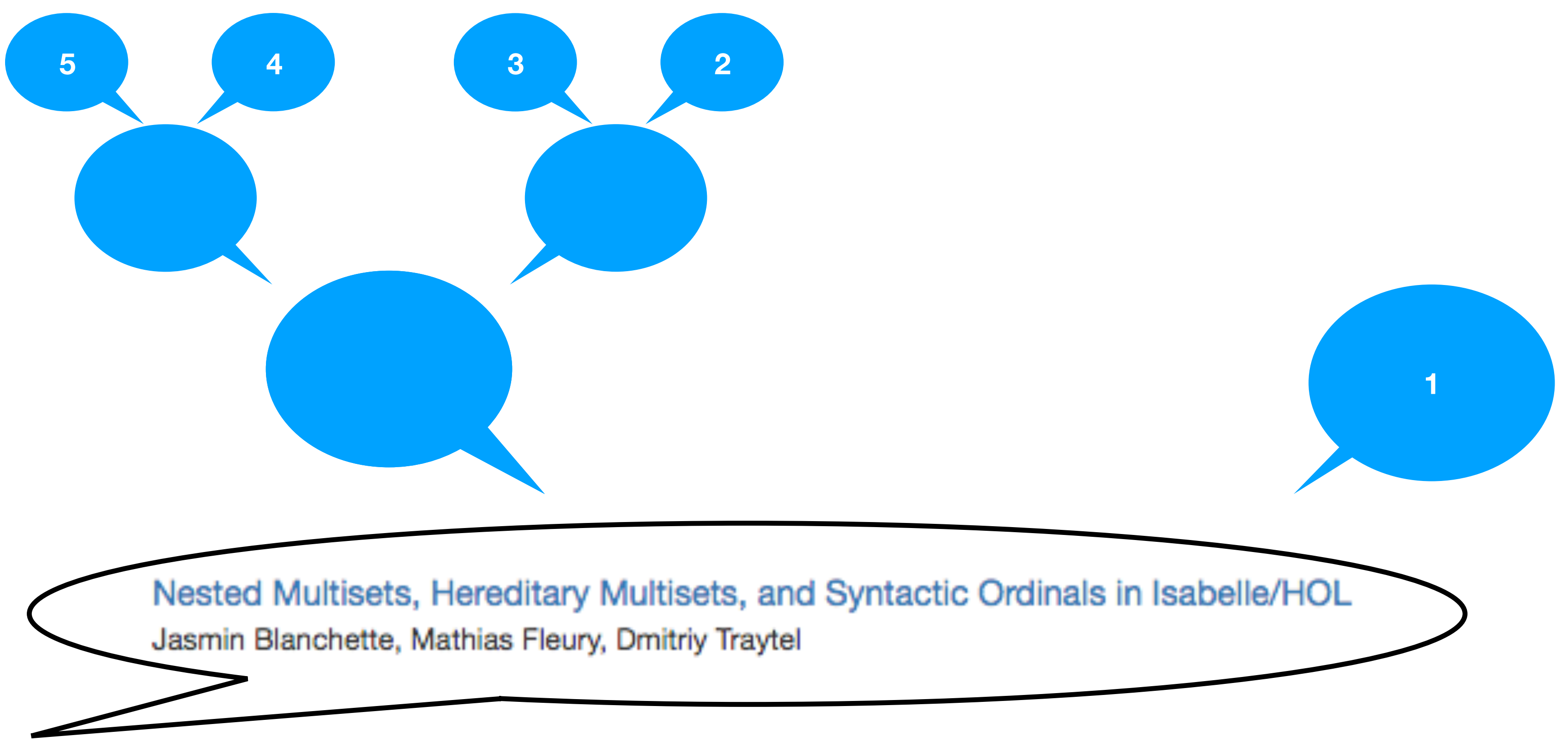
```
A * B = HMSet (map_mset (case_prod (op +))
                   (hmsetmset A ×# hmsetmset B))
```

VU VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

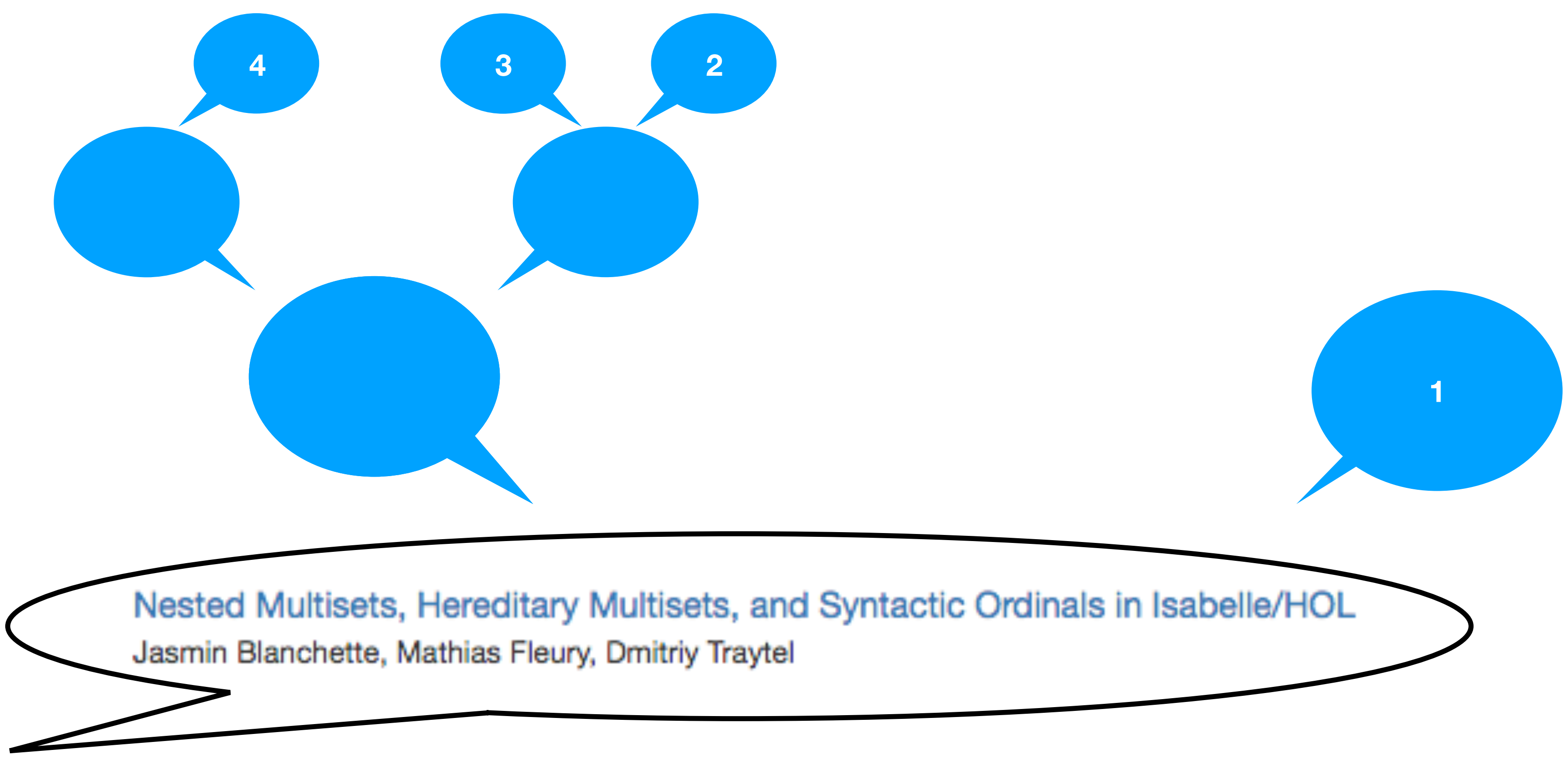
ETH zürich

The “Question” Hydra

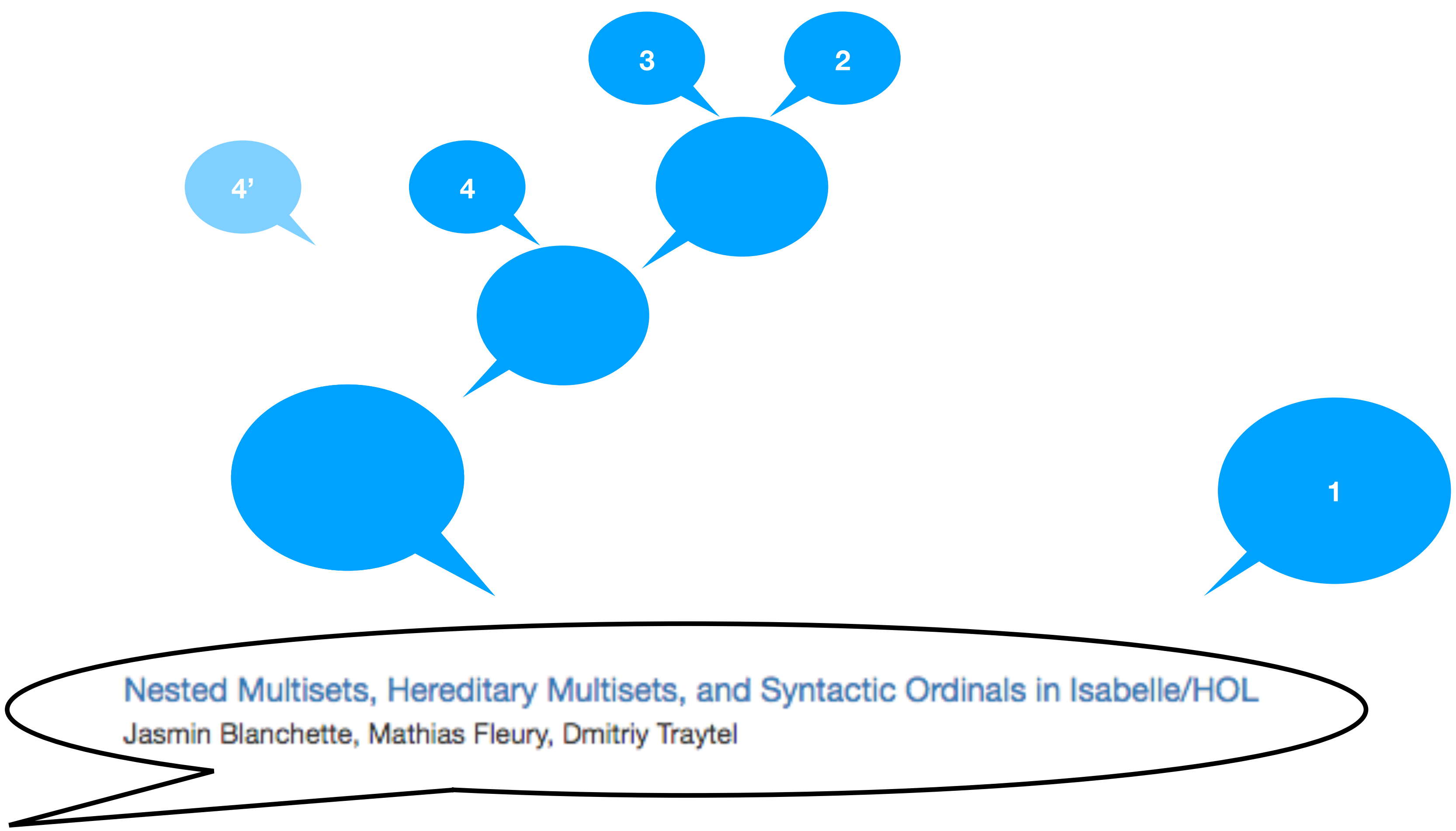


VU VRIJE UNIVERSITEIT AMSTERDAM max planck institut informatik ETH zürich

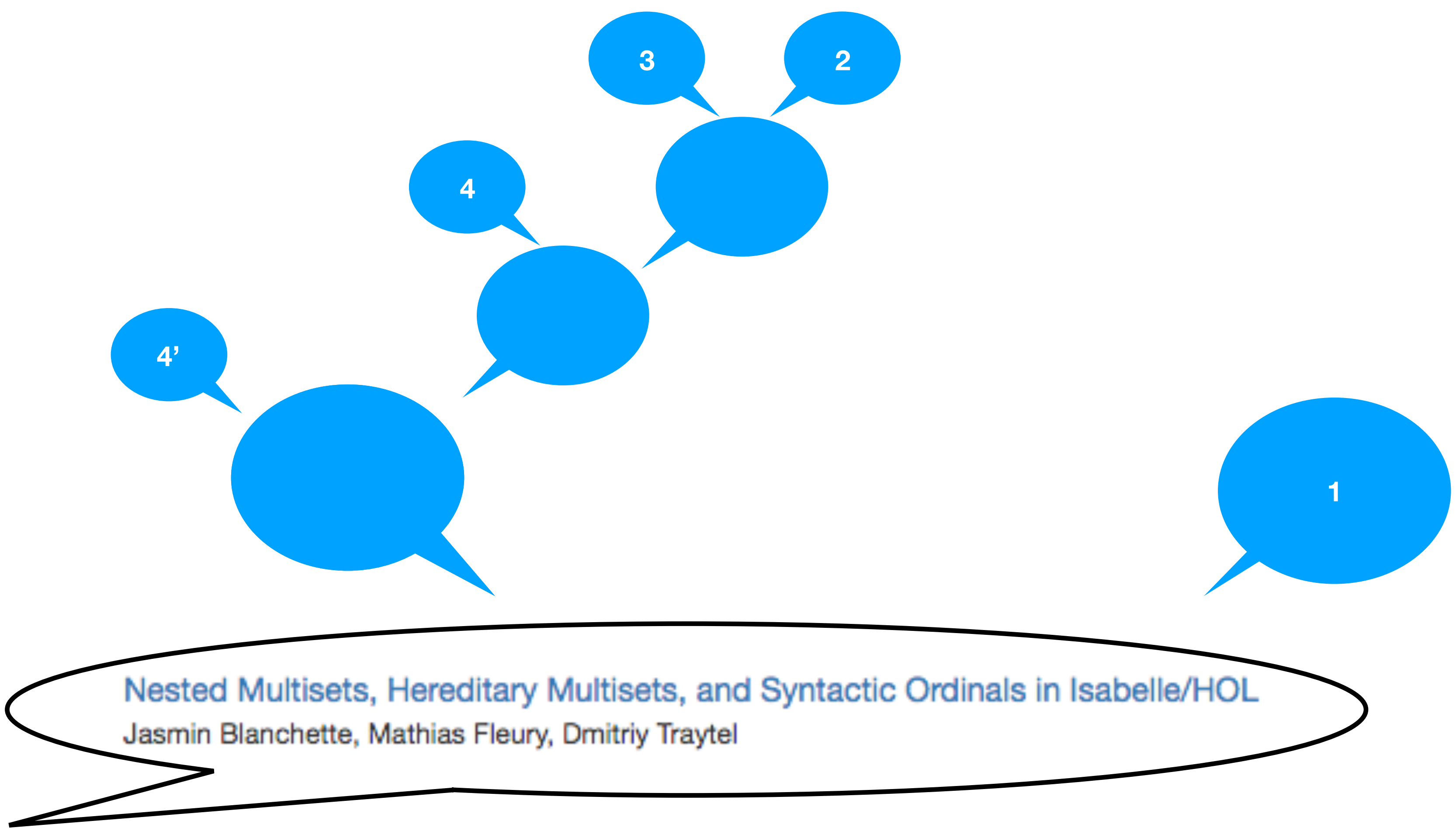
The “Question” Hydra



The “Question” Hydra

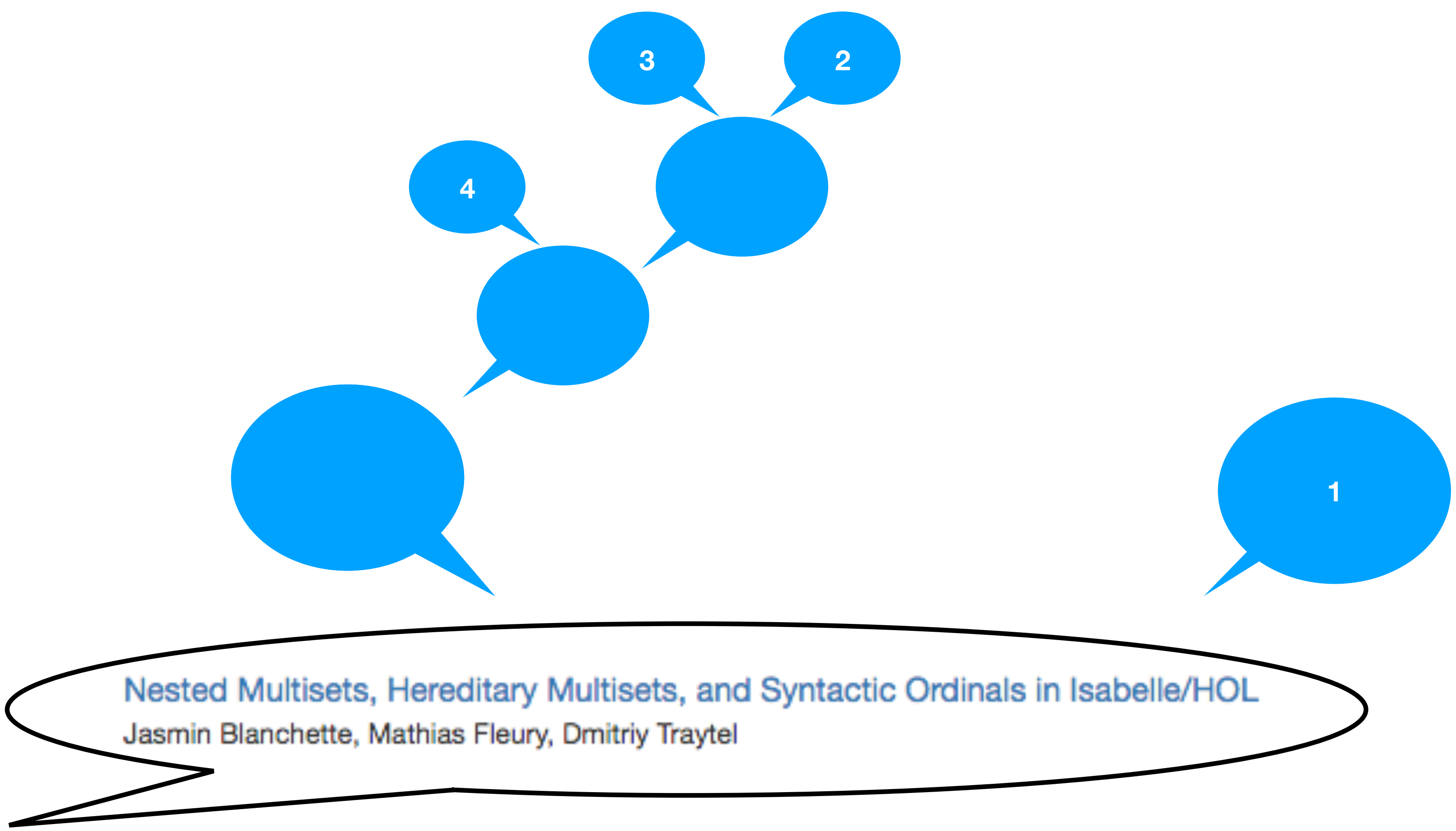


The "Question" Hydra



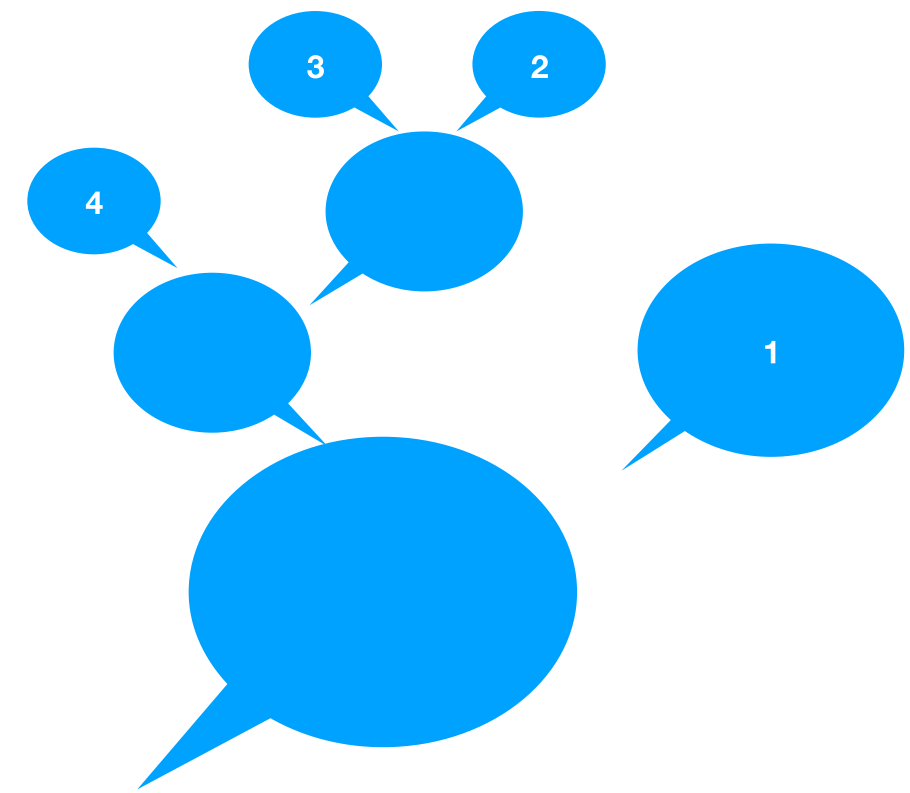
VU VRIJE UNIVERSITEIT AMSTERDAM | max planck institut informatik | ETH zürich

The “Question” Hydra



VU VRIJE UNIVERSITEIT AMSTERDAM max planck institut informatik ETH zürich

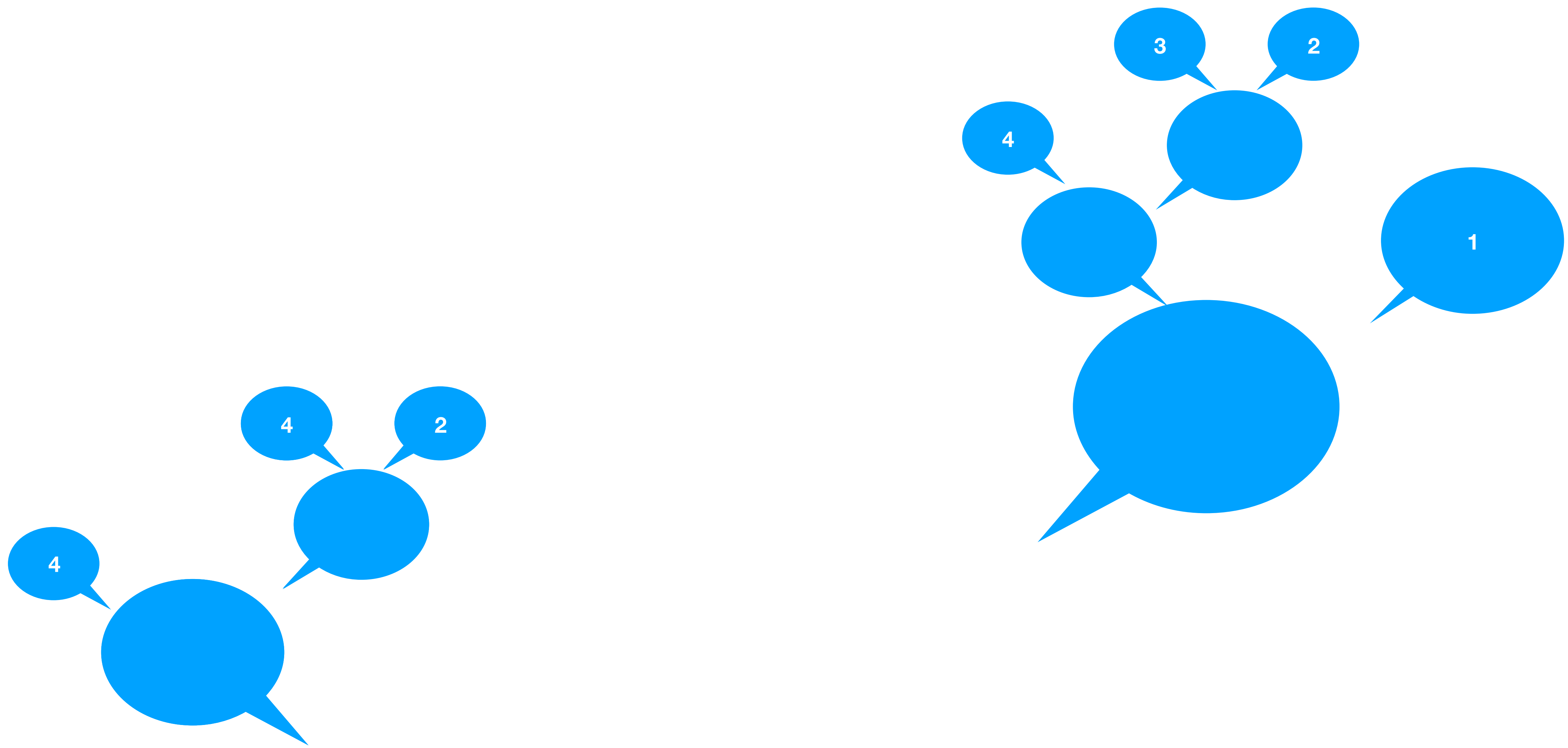
The “Question” Hydra



Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL
Jasmin Blanchette, Mathias Fleury, Dmitriy Traytel

VU VRIJE UNIVERSITEIT AMSTERDAM — max planck institut informatik — ETH zürich

The "Question" Hydra



Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL

Jasmin Blanchette, Mathias Fleury, Dmitriy Traytel

The “Question” Hydra

Nested Multisets, Hereditary Multisets, and Syntactic Ordinals in Isabelle/HOL
Jasmin Blanchette, Mathias Fleury, Dmitriy Traytel

Will there be a coffee break?

encode(   ) = 0

encode() = ω^encode() + encode()

Will there be a coffee break?

```
encode(        )  =  0

encode(        )  = ω^encode(   ) + encode(   )
```

```
encode(        )  = ω^encode(   ) + encode(   )
                  = ω^(ω^encode(   ) + encode(   )
                      + encode(   )
```

Will there be a coffee break?

```
encode(        )  =  0

encode(        )  = ω^encode(   ) + encode(   )
```

```
encode(              )  = ω^encode(   ) + encode(   )

                        = ω^(ω^encode(   ) + encode(   )

                             + encode(   )
```

Will there be a coffee break?

```
encode(        )   =   0

encode(        )   = ω^encode(   ) + encode(   )
```

```
encode(              )   = ω^encode(   ) + encode(   )

                         = ω^(ω^encode(   ) + encode(   )

                               + encode(   )

encode(              )   = ω^encode(   ) + encode(        )

                         = 2 * ω^encode(   ) + encode(   )
```

Will there be a coffee break?

```
encode(        )  =  0

encode(        )  = ω^encode(   ) + encode(   )
```

```
encode(        )  = ω^encode(   ) + encode(   )
                  = ω^(ω^encode(   ) + encode(   )
                      + encode(   )
      V
encode(        )  = ω^encode(   ) + encode(   )
                  = 2 * ω^encode(   ) + encode(   )
```

Multisets

Syntactic Ordinals

Nested Multisets

Signed Hereditary Multisets

Hereditary Multisets

Signed Hereditary Multisets

```
lemma (* [Ludwig and Waldmann] *)
  assumes α2 + β2 * γ < α1 + β1 * γ and β2 ≤ β1 and γ < δ
  shows α2 + β2 * δ < α1 + β1 * δ
```

Sketch of the “ideal” proof

have $\beta 2 * (\delta - \gamma) < \beta 1 * (\delta - \gamma)$

thus the result

VU VRIJE UNIVERSITEIT AMSTERDAM max planck institut informatik ETH zürich

```
lemma (* [Ludwig and Waldmann] *)
  assumes α2 + β2 * γ < α1 + β1 * γ and β2 ≤ β1 and γ < δ
  shows α2 + β2 * δ < α1 + β1 * δ
```

Sketch of the “ideal” proof

have β2 * (δ - γ) < β1 * (δ - γ)

thus the result

But: subtraction is ill-behaved

$$\alpha \cdot (\beta - \gamma) = \omega^2 + \omega \quad \neq \quad \omega = \alpha \cdot \beta - \alpha \cdot \gamma$$

$$\text{where } \alpha = \omega^2 + \omega \text{ and } \qquad \beta = 1 \text{ and } \gamma = \omega$$

Lemma: $\alpha_1 + \beta_1 \gamma > \alpha_2 + \beta_2 \gamma$

$\beta_1 \geq \beta_2$

$\delta > \gamma$

$\Rightarrow \alpha_1 + \beta_1 \delta > \alpha_2 + \beta_2 \delta$

Proof:

$\beta_1 = \beta_0 + \beta_1'$, $\beta_2 = \beta_0 + \beta_2'$, $deg(\beta_1') > deg(\beta_2')$ or $\beta_1' = \beta_2' = 0$

$\gamma = \eta + \gamma'$, $\delta = \eta + \delta'$, $deg(\delta') > deg(\gamma')$

$$\alpha_1 + \beta_0 \gamma + \beta_1' \gamma = \alpha_1 + \beta_1 \gamma > \alpha_2 + \beta_2 \gamma = \alpha_2 + \beta_0 \gamma + \beta_2' \gamma$$

(*) $\Rightarrow \alpha_1 + \beta_1' \gamma > \alpha_2 + \beta_2' \gamma$ (by tot., mon.)

$$
\begin{aligned}
\alpha_2 + \beta_2 \delta &= \alpha_2 + \beta_0 \delta + \beta_2' \delta \\
&= \alpha_2 + \beta_0 \delta + \beta_2' \eta + \beta_2' \delta' \\
&\leq \alpha_2 + \beta_0 \delta + \underline{\beta_2' \eta} + \beta_2' \delta' + \underline{\beta_2' \gamma'} \quad \text{(mon.)} \\
&= \underline{\alpha_2 + \beta_2' \gamma} + \beta_0 \delta + \beta_2' \delta' \\
&< \alpha_1 + \underline{\beta_1' \gamma} + \underline{\beta_0 \delta} + \beta_2' \delta' \quad (*, \text{ mon.}) \\
&= \alpha_1 + \beta_1' \eta + \underline{\beta_1' \gamma'} + \beta_0 \eta + \beta_0 \delta' + \underline{\beta_2' \delta'} \\
&\leq \alpha_1 + \beta_1' \eta + \beta_0 \eta + \beta_0 \delta' + \beta_1' \delta' \\
&\quad (deg(\beta_1' \delta') > deg(\beta_1' \gamma' + \beta_2' \\
&\quad \text{or } \beta_1' \delta' = \beta_1' \gamma' = \beta_2' \delta' = 0 \\
&= \alpha_1 + \beta_1 \delta
\end{aligned}
$$

```
lemma (* [Ludwig and Waldmann] *)
  assumes α2 + β2 * γ < α1 + β1 * γ and β2 ≤ β1 and γ < δ
  shows α2 + β2 * δ < α1 + β1 * δ
proof -
  obtain β0 β2a β1a where β1 = β0 + β1a and β2 = β0 + β2a and
    head_ω β2a < head_ω β1a ∨ β2a = 0 ∧ β1a = 0 by …

  obtain η γa δa where γ = η + γa and δ = η + δa and
    head_ω γa < head_ω δa by …

  have α2 + β0 * γ + β2a * γ = α2 + β2 * γ by …
  also have … < α1 + β1 * γ by …
  also have … = α1 + β0 * γ + β1a * γ by …
  finally have *: α2 + β2a * γ < α1 + β1a * γ by …

  have α2 + β2 * δ = α2 + β0 * δ + β2a * δ by …
  also have … = α2 + β0 * δ + β2a * η + β2a * δa by …
  also have … ≤ α2 + β0 * δ + β2a * η + β2a * δa + β2a * γa by …
  also have … = α2 + β2a * γ + β0 * δ + β2a * δa by …
  also have … < α1 + β1a * γ + β0 * δ + β2a * δa by …
  also have … = α1 + β1a * η + β1a * γa + β0 * η + β0 * δa + β2a * δa by …
  also have … ≤ α1 + β1a * η + β0 * η + β0 * δa + β1a * δa by …
  finally show ?thesis by …
qed
```

```
equiv_zmset (Mp, Mn) (Np, Nn) = Mp + Nn = Np + Mn

quotient_type 'a zmultiset =
  'a multiset × 'a multiset / equiv_zmset
```

```
equiv_zmset (Mp, Mn) (Np, Nn) = Mp + Nn = Np + Mn

quotient_type 'a zmultiset =
  'a multiset × 'a multiset / equiv_zmset

equiv_zmset ({}, {7}) ({3}, {3,7})
```

VU VRIJE UNIVERSITEIT AMSTERDAM   max planck institut informatik   ETH zürich

```
equiv_zmset (Mp, Mn) (Np, Nn) = Mp + Nn = Np + Mn
```

```
quotient_type 'a zmultiset =
  'a multiset × 'a multiset / equiv_zmset
```

```
equiv_zmset ({}, {7}) ({3}, {3,7})
```

Does the operation on pairs respect
the equivalence relation?

```
lift_definition minus_zmultiset
  :: 'a zmultiset ⇒ 'a zmultiset ⇒ 'a zmultiset
  is λ(Mp, Mn) (Np, Nn). (Mp + Nn, Mn + Np)
```

VU VRIJE UNIVERSITEIT AMSTERDAM
max planck institut informatik
ETH zürich

Associativity of multiplication builds down to

```
An * (Bn * Cn + Bp * Cp - (Bn * Cp + Cn * Bp))
   + (Cn * (An * Bp + Bn * Ap - (An * Bn + Ap * Bp))
   + (Ap * (Bn * Cp + Cn * Bp - (Bn * Cn + Bp * Cp))
   + Cp * (An * Bn + Ap * Bp - (An * Bp + Bn * Ap)))) =
 An * (Bn * Cp + Cn * Bp - (Bn * Cn + Bp * Cp))
   + (Cn * (An * Bn + Ap * Bp - (An * Bp + Bn * Ap))
   + (Ap * (Bn * Cn + Bp * Cp - (Bn * Cp + Cn * Bp))
   + Cp * (An * Bp + Bn * Ap - (An * Bn + Ap * Bp))))
```

Associativity of multiplication builds down to

```
An * (Bn * Cn + Bp * Cp - (Bn * Cp + Cn * Bp))
   + (Cn * (An * Bp + Bn * Ap - (An * Bn + Ap * Bp))
   + (Ap * (Bn * Cp + Cn * Bp - (Bn * Cn + Bp * Cp))
   + Cp * (An * Bn + Ap * Bp - (An * Bp + Bn * Ap)))) =
 An * (Bn * Cp + Cn * Bp - (Bn * Cn + Bp * Cp))
   + (Cn * (An * Bn + Ap * Bp - (An * Bp + Bn * Ap))
   + (Ap * (Bn * Cn + Bp * Cp - (Bn * Cp + Cn * Bp))
   + Cp * (An * Bp + Bn * Ap - (An * Bn + Ap * Bp))))
```

Magic truncation lemma:

```
lemma a * (c - b) + a * b = a * (b - c) + a * c
  by (metis distrib_left diff_plus_sym_hmset)
```

VRIJE UNIVERSITEIT AMSTERDAM

max planck institut informatik

ETH zürich

Signed hereditary multisets: `hmultiset zmultiset`

unsigned powers → hmultiset

signed coefficient → zmultiset

e.g. `ω^2 - ω - 1`

```
lemma (* new version of Ludwig and Waldmann’s lemma *)
  assumes α2 + β2 * γ < α1 + β1 * γ and β2 ≤ β1 and γ < δ
  shows α2 + β2 * δ < α1 + β1 * δ
proof -
  let ?z = zhmset_of

  have ?z α2 + ?z β2 * ?z δ <
       ?z α1 + ?z β1 * ?z γ + ?z β2 * (?z δ - ?z γ)
     by …
  also have … ≤ ?z α1 + ?z β1 * ?z γ + ?z β1 * (?z δ - ?z γ)
     by …
  finally show ?thesis by …
qed
```

VU VRIJE UNIVERSITEIT AMSTERDAM · max planck institut informatik · ETH zürich

```
lemma (* new version of Ludwig and Waldmann's lemma *)
  assumes α2 + β2 * γ < α1 + β1 * γ and β2 ≤ β1 and γ < δ
  shows α2 + β2 * δ < α1 + β1 * δ
proof -
  let ?z = zhmset_of

  have ?z α2 + ?z β2 * ?z δ <
       ?z α1 + ?z β1 * ?z γ + ?z β2 * (?z δ - ?z γ)
     by …
  also have … ≤ ?z α1 + ?z β1 * ?z γ + ?z β1 * (?z δ - ?z γ)
     by …
  finally show ?thesis by …
qed
```

Now Waldmann has a proper theoretical foundation for ordinals with signed coefficients

Conclusion

Many nice tools in Isabelle, especially:

- datatypes
- lifting package
- quotients
- Sledgehammer

Formalisation in the Archive of Formal Proofs, also:

- Goodstein sequence
- key lemma towards decidability of Unary PCF