Compactness in topology and computation

Martín Escardó

School of Computer Science, University of Birmingham, UK

CCA 2005, Satellite Seminar, Kyoto, Japan, 26 Aug 2005

Plan of the talk

1. We develop a computational notion of compactness that arises in

   - Paul Taylor's work on Abstract Stone Duality, and
   - my work on synthetic topology of computational spaces.

Plan of the talk

1. We develop a computational notion of compactness that arises in

   - Paul Taylor's work on Abstract Stone Duality, and
   - my work on synthetic topology of computational spaces.

The notion is based on universal quantification.

Plan of the talk

1. We develop a computational notion of compactness that arises in

   - Paul Taylor's work on Abstract Stone Duality, and
   - my work on synthetic topology of computational spaces.

   The notion is based on universal quantification.

2. On the way, we sketch applications to

   - computational analysis (known to you), and
   - non-deterministic and probabilistic computation (new to me).

Disclaimer

There is much to say about constructive mathematics . . .

. . . as opposed to computable mathematics.

In this talk I study computation using classical logic.

But questions about constructive aspects are most welcome.

As some of you know, I am rather interested in this dimension.

But also I don't take classical or constructive mathematics as a matter of faith.

Please click here if you agree with the above conditions and wish to continue attending the talk.

Reading the small print

There is much to say about constructive mathematics . . .

. . . as opposed to computable mathematics.

In this talk I study computation using classical logic.

But questions about constructive aspects are most welcome.

As some of you know, I am rather interested in this dimension.

But also I don't take classical or constructive mathematics as a matter of faith.

Universal quantification in computation

Consider the functional $F\colon \mathbb{N}^{\mathbb{N}} \to \mathrm{Bool}$ defined by

$$F(s) = \mathrm{true} \iff s_n = 0 \text{ for all } n \in \mathbb{N}.$$

As you all know, this is not computable:

- If the input s is given as a *blackbox*,

  this violates *continuity*.

- If the input s is given as a *whitebox*,

  this violates the *Halting Problem*.

But, perhaps surprisingly:

Some infinite sets *do allow* definition of algorithms by universal quantification over them.

An example was implicit in

- the work of Gandy (1970's, unpublished), and
- independently Berger (1990).

It was made explicit by Simpson (1998).

The Gandy–Berger program (written in Haskell)

Preliminary notation:

```
type    Nat = ...               -- natural numbers
type    Two = ...               -- 2 = {0,1}
type Cantor = Nat -> Two        -- 2^N

  hd :: Cantor -> Two
  tl :: Cantor -> Cantor
cons :: (Two, Cantor) -> Cantor

    hd(s) = s(0)
    tl(s) = \i -> s(i+1)
cons(n,s) = \i -> if i == 0 then n else s(i-1)
```

The Gandy–Berger program

```
epsilon :: (Cantor -> Bool) -> Cantor
 exists :: (Cantor -> Bool) -> Bool
 forall :: (Cantor -> Bool) -> Bool

epsilon(p) =
  let l = cons(0,epsilon(\s -> p(cons(0,s))))
      r = cons(1,epsilon(\s -> p(cons(1,s))))
  in if p(l) then l else r

exists(p) = p(epsilon(p))

forall(p) = not(exists(\s -> not(p(s))))
```

Loading the program

```
$ hugs kyoto.hs
```

Loading the program

```
$ hugs kyoto.hs

__   __ __  __  ____   ___
||   || ||  || ||  || ||__
||___|| ||__|| ||__||  __||
||---||         ___||
||   ||
||   || Version: November 2002

Reading file "/usr/lib/hugs/lib/Prelude.hs":
Reading file "kyoto.hs":
Hugs session for:
/usr/lib/hugs/lib/Prelude.hs
kyoto.hs
```

Running the program

>

Running the program

```
> forall(\s->exists(\t->s(t(0)+t(1))==t(s(1)+s(2))))
```

Running the program

```
> forall(\s->exists(\t->s(t(0)+t(1))==t(s(1)+s(2))))
True
>
```

Running the program

```
> forall(\s->exists(\t->s(t(0)+t(1))==t(s(1)+s(2))))
True

> exists(\s->forall(\t->s(t(0)+t(1))==t(s(1)+s(2))))
```

Running the program

```
> forall(\s->exists(\t->s(t(0)+t(1))==t(s(1)+s(2))))
True

> exists(\s->forall(\t->s(t(0)+t(1))==t(s(1)+s(2))))
False
>
```

Running the program

```
> forall(\s->exists(\t->s(t(0)+t(1))==t(s(1)+s(2))))
True

> exists(\s->forall(\t->s(t(0)+t(1))==t(s(1)+s(2))))
False
>
```

The above two queries amount to

$$\forall s \in 2^{\mathbb{N}} \quad \exists t \in 2^{\mathbb{N}} \quad s_{t_0+t_1} = t_{s_1+s_2},$$
$$\exists s \in 2^{\mathbb{N}} \quad \forall t \in 2^{\mathbb{N}} \quad s_{t_0+t_1} = t_{s_1+s_2}.$$

Sample application (Alex Simpson, 1998, LNCS 1450)

Compute the supremum of the values of a function $2^{\mathbb{N}} \to 2^{\mathbb{N}}$.

```
max :: (Cantor, Cantor) -> Cantor
-- easy definition of lexicographic max omitted

sup :: (Cantor -> Cantor) -> Cantor

sup(f) = let h = hd(f(\n -> 0))
         in if forall(\s -> hd(f(s)) == h)
            then cons(h, sup(tl.f))
            else max( sup(\s -> f(cons(0,s))),
                      sup(\s -> f(cons(1,s))))
```

More sample applications (Alex Simpson, *loc. cit.*)

Compute the functionals $\max, \int : ([0,1] \to \mathbb{R}) \to \mathbb{R}$.

Use signed-digit binary representation.

```
avg :: (I, I) -> I -- definition of average omitted
riemann :: (I -> I) -> I
riemann(f) =
   let h = hd(f(zero))
   in if forall(\s -> hd(f(s)) == h)
      then cons(h, riemann(tl.f))
      else avg( riemann(\s -> f(cons(0,s))),
                riemann(\s -> f(cons(1,s))))
```

(This idea was previously used by Edalat and Escardó 1996, for both $\max$ and $\int$.)

Summary so far

- Universal quantification over some infinite data spaces is computable.

- This has non-trivial applications to computational analysis.

  (The ones given are long known results, but this is not the point.)

Topological analysis of this phenomenon

A topological space is called compact if every open cover has a finite subcover.

I'll now relate this to universal quantification.

Lemma

The following are equivalent for spaces X and Y:

1. The projection $Y \times X \to Y$ is a closed map.

2. $F \subseteq Y \times X$ closed $\implies \{y \in Y \mid \exists x \in X.(y,x) \in F\}$ closed.

3. $W \subseteq Y \times X$ open $\implies \{y \in Y \mid \forall x \in X.(y,x) \in W\}$ open.

Proof of the lemma

The following are equivalent for spaces X and Y:

1. The projection $\pi\colon Y \times X \to Y$ is a closed map.

2. $F \subseteq Y \times X$ closed $\implies \underbrace{\{y \in Y \mid \exists x \in X.(y,x) \in F\}}_{\pi(F)}$ closed.

3. $W \subseteq Y \times X$ open $\implies \underbrace{\{y \in Y \mid \forall x \in X.(y,x) \in W\}}_{(\pi(W^c))^c}$ open.

Theorem

TFAE for a topological space X:

1. X is compact.

2. $\forall Y$, the projection $Y \times X \to Y$ is a closed map.

3. $\forall Y$, $F \subseteq Y \times X$ closed $\implies \{y \in Y \mid \exists x \in X.(y, x) \in F\}$ closed.

4. $\forall Y$, $W \subseteq Y \times X$ open $\implies \{y \in Y \mid \forall x \in X.(y, x) \in W\}$ open.

We are interested in the equivalence of (1) and (4):

a topological space X is compact $\iff$ for any Y,

openness of $\quad W \subseteq Y \times X$

implies that of $\quad \{y \in Y \mid \forall x \in X.(y, x) \in W\}$,

because, replacing “open” by “semi-decidable”, it gives rise to . . .

Computational compactness

We say that a computational space X is compact $\iff$ for any Y,

semidecidability of $W \subseteq Y \times X$

implies that of $\{y \in Y \mid \forall x \in X.(y, x) \in W\}$.

$\mathbb{N}$ is of course a counter-example

Recall the functional $F\colon \mathbb{N}^{\mathbb{N}} \to \mathrm{Bool}$ defined by

$F(s) = \mathrm{true} \iff s_n = 0$ for all $n \in \mathbb{N}$.

This is not computable, as we have discussed.

This has to do with the fact that $\mathbb{N}$ is not (computationally) compact:

1. Consider $Y = \mathbb{N}^{\mathbb{N}}$ and $X = \mathbb{N}$ in the previous definition.
2. Consider $W \subseteq \mathbb{N}^{\mathbb{N}} \times \mathbb{N}$ defined by $W = \{(s, n) \mid s_n = 0\}$.
3. This is decidable and hence semidecidable.
4. However, the singleton set $\{s \in \mathbb{N}^{\mathbb{N}} \mid \forall n \in \mathbb{N}.(s, n) \in W\}$ is not semidecidable and hence not decidable.

Look at the definition again:

X is compact $\iff$ for any Y,

semidecidability of $\quad W \subseteq Y \times X$

implies that of $\quad \{y \in Y \mid \forall x \in X.(y, x) \in W\}$.

Shouldn't we require uniformity in the definition?

A computational space X is compact $\iff$ for any Y,

semidecidability of $\quad W \subseteq Y \times X$

implies that of $\quad \{y \in Y \mid \forall x \in X.(y,x) \in W\}$,

uniformly in W.

We instead consider an alternative formulation for which uniformity is automatic, using the notion of *exponentiation*.

Exponentiation of topological spaces

Let S be any space.

We write S^X to denote the *set of continuous maps* from X to S with a topology such that

1. the evaluation map $e\colon S^X \times X \to S$ defined by $e(f, x) = f(x)$ is continuous, and

2. for any space Y, if $f\colon Y \times X \to S$ is continuous then so is its transpose $\bar{f}\colon Y \to S^X$ defined by $\bar{f}(y) = (x \mapsto f(y, x))$.

Such an exponential topology doesn't always exist, but when it does it is easily seen to be unique.

Exponentiation of topological spaces

Let S be any space.

We write S^X to denote the set of continuous maps from X to S with a topology such that

1. the evaluation map $e\colon S^X \times X \to S$ defined by $e(f,x) = f(x)$ is continuous, and

2. for any space Y, if $f\colon Y \times X \to S$ is continuous then so is its transpose $\bar{f}\colon Y \to S^X$ defined by $\bar{f}(y) = (x \mapsto f(y,x))$.

Such an exponential topology doesn't always exist, but when it does it is easily seen to be unique.

Exponentiation of topological spaces

Let S be any space.

We write S^X to denote the set of continuous maps from X to S with a topology such that

1. the evaluation map $e\colon S^X \times X \to S$ defined by $e(f,x) = f(x)$ is continuous, and

2. for any space Y, if $f\colon Y \times X \to S$ is continuous then so is its transpose $\bar{f}\colon Y \to S^X$ defined by $\bar{f}(y) = (x \mapsto f(y,x))$.

Such an exponential topology doesn't always exist, but when it does it is easily seen to be unique.

Exponentiation of topological spaces

Let S be any space.

We write S^X to denote the set of continuous maps from X to S with a topology such that

1. the evaluation map $e\colon S^X \times X \to S$ defined by $e(f, x) = f(x)$ is continuous, and

2. for any space Y, if $f\colon Y \times X \to S$ is continuous then so is its transpose $\bar{f}\colon Y \to S^X$ defined by $\bar{f}(y) = (x \mapsto f(y, x))$.

Such an exponential topology doesn't always exist, but when it does it is easily seen to be unique.

Sierpinski space

Let $\mathbb{S}$ be the Sierpinski space with

an isolated point $\top$ (true) and

a limit point $\bot$ (false).

The open sets are $\emptyset$, $\{\top\}$ and $\{\bot, \top\}$, but not $\{\bot\}$.

A map $p\colon X \to \mathbb{S}$ is continuous $\iff$ $p^{-1}(\top)$ is open.

$U \subseteq X$ is open $\iff$ its characteristic map $\chi_U\colon X \to \mathbb{S}$ is continuous.

Theorem

TFAE if the exponential $\mathbb{S}^X$ exists:

1. X is compact.

2. The universal-quantification functional $A\colon \mathbb{S}^X \to \mathbb{S}$ defined by

$$A(p) = \top \iff \forall x \in X . p(x) = \top$$

   is continuous.

Proof

We routinely use the previously proved:

Lemma. TFAE for a topological space X:

1. X is compact.
2. For every space Y and every open $W \subseteq Y \times X$, the set $\{y \in Y \mid \forall x \in X.(y,x) \in W\}$ is open.

. . . and nothing else.

In particular, we don't need to know what the exponential topology is.

The universal property suffices.

Proof

X compact $\implies$ $A\colon \mathbb{S}^X \to \mathbb{S}$ continuous:

Because the evaluation map $e\colon \mathbb{S}^X \times X \to \mathbb{S}$ is continuous, the set

$$W \stackrel{\text{def}}{=} e^{-1}(\top) = \{(p,x) \in \mathbb{S}^X \times X \mid p(x) = \top\}$$

is open.

Considering $Y = \mathbb{S}^X$ in the lemma, the compactness of X gives the open set

$$\{p \in \mathbb{S}^X \mid \forall x \in X.(p,x) \in W\} = \{p \in \mathbb{S}^X \mid \forall x \in X.p(x) = \top\} = A^{-1}(\top).$$

Therefore A is continuous.

Proof

X *compact* $\implies A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous:*

Because the evaluation map $e\colon \mathbb{S}^X \times X \to \mathbb{S}$ is continuous, the set

$$W \stackrel{\text{def}}{=} e^{-1}(\top) = \{(p,x) \in \mathbb{S}^X \times X \mid p(x) = \top\}$$

is open.

Considering $Y = \mathbb{S}^X$ in the lemma, the compactness of X gives the open set

$$\{p \in \mathbb{S}^X \mid \forall x \in X.(p,x) \in W\} = \{p \in \mathbb{S}^X \mid \forall x \in X.p(x) = \top\} = A^{-1}(\top).$$

Therefore A is continuous.

Proof

X *compact* $\implies A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous:*

Because the evaluation map $e\colon \mathbb{S}^X \times X \to \mathbb{S}$ is continuous, the set

$$W \stackrel{\text{def}}{=} e^{-1}(\top) = \{(p,x) \in \mathbb{S}^X \times X \mid p(x) = \top\}$$

is open.

Considering $Y = \mathbb{S}^X$ in the lemma, the compactness of X gives the open set

$$\{p \in \mathbb{S}^X \mid \forall x \in X.(p,x) \in W\} = \{p \in \mathbb{S}^X \mid \forall x \in X.p(x) = \top\} = A^{-1}(\top).$$

Therefore A is continuous.

Proof

X *compact* $\implies A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous:*

Because the evaluation map $e\colon \mathbb{S}^X \times X \to \mathbb{S}$ is continuous, the set

$$W \stackrel{\text{def}}{=} e^{-1}(\top) = \{(p,x) \in \mathbb{S}^X \times X \mid p(x) = \top\}$$

is open.

Considering $Y = \mathbb{S}^X$ in the lemma, the compactness of X gives the open set

$$\{p \in \mathbb{S}^X \mid \forall x \in X.(p,x) \in W\} = \{p \in \mathbb{S}^X \mid \forall x \in X.p(x) = \top\} = A^{-1}(\top).$$

Therefore A is continuous.

Proof

$A\colon \mathbb{S}^X \to \mathbb{S}$ continuous $\implies$ X compact:

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W\colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w\colon Y \to \mathbb{S}^X$ and the composite $A \circ w\colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Proof

$A \colon \mathbb{S}^X \to \mathbb{S}$ *continuous* $\implies$ X *compact:*

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W \colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w \colon Y \to \mathbb{S}^X$ and the composite $A \circ w \colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Proof

$A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous* $\implies$ X *compact:*

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W\colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w\colon Y \to \mathbb{S}^X$ and the composite $A \circ w\colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Proof

$A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous* $\implies$ X *compact:*

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W\colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w\colon Y \to \mathbb{S}^X$ and the composite $A \circ w\colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Proof

$A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous* $\implies$ X *compact:*

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W\colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w\colon Y \to \mathbb{S}^X$ and the composite $A \circ w\colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Proof

$A\colon \mathbb{S}^X \to \mathbb{S}$ *continuous* $\implies$ X *compact:*

To apply the lemma, let Y be any space and $W \subseteq Y \times X$ be open.

Because $\chi_W\colon Y \times X \to \mathbb{S}$ is continuous, so are its transpose $w\colon Y \to \mathbb{S}^X$ and the composite $A \circ w\colon Y \to \mathbb{S}$.

Hence $V \stackrel{\text{def}}{=} (A \circ w)^{-1}(\top)$ is open.

Now $\forall x \in X.(y, x) \in W$ iff $\forall x \in X.w(y)(x) = \top$ iff $A(w(y)) = \top$ iff $y \in V$.

This shows that $\{y \in Y \mid \forall x \in X.(y, x) \in W\} = V$ and hence it is open, as required.

Uniformity for free (for exponential experts)

If X is compact, then we have a continuous map $\mathbb{S}^X \to \mathbb{S}$.

Elevating this map to the power Y, we get $(\mathbb{S}^X)^Y \to \mathbb{S}^Y$.

But $(\mathbb{S}^X)^Y \cong \mathbb{S}^{Y \times X}$.

Hence we get a continuous map $\mathbb{S}^{Y \times X} \to \mathbb{S}^Y$.

This sends the characteristic map of $W \subseteq Y \times X$ to that of $\{y \in Y \mid \forall x \in X.(y,x) \in W\}$.

In this sense, the passage from W to $\{y \in Y \mid \forall x \in X.(y,x) \in W\}$ is continuous.

Computational interpretation of the Sierpinski space

Space of results of semi-decisions:

$\top$ = termination = observable true.

$\bot$ = non-termination = unobservable false.

The asymmetry of the topology of the Sierpinski space reflects the asymmetry of the notion of semi-decision.

The negation map $\mathbb{S} \to \mathbb{S}$ is not continuous.

Semi-decidable subset of $X \cong$ computable map $X \to \mathbb{S}$.

Official formulation of computational compactness

We say that X is computationally compact if the universal quantification functional $\mathbb{S}^X \to \mathbb{S}$ is computable.

In other words:

X is computationally compact $\iff$ universal quantification over semi-decidable predicates is semi-decidable.

Generality and precision of the definition

One has to work in computational settings that incorporate a Sierpinski domain $\mathbb{S}$ and exponentiation $\mathbb{S}^X$ for computational spaces X.

One has to say precisely what a Sierpinski domain is.

Possible settings:

- Taylor's ASD.
- Higher-type programming languages (e.g. Haskell, ML, PCF).
- Realizability toposes with a dominance.
- Cartesian closed categories of represented spaces (Schröder).

Haskell compactness of the Cantor space

We previously discussed a boolean-valued universal quantification.

```
data S = T

ifs :: (S,a) -> a
ifs(T,x) = x

forall :: (Cantor -> S) -> S

forall(p) =
  p(ifs(forall(\s -> p(cons(0,s)))
    /\  forall(\s -> p(cons(1,s)))), \n -> 0))
```

The Tychonoff program

The product of an r.e. sequence of computationally compact spaces is itself computationally compact.

In Haskell, we need witnesses for the inhabitation of each space.

```
type Seq a = Nat -> a
type Quant = (a -> S) -> S

t :: (Seq a, Seq (Quant a)) -> (Quant (Seq a))
t(w,a) = \p ->
hd(a)(\x->p(ifs(t(tl(w),tl(a))(\s->p(cons(x,s))),w)))
```

The proof that this program works is non-trivial. It uses denotational semantics, domain theory and topology (Tychonoff!).

There are plenty of computationally compact sets

Starting with the finite ones, one gets non-trivial ones using the Tychonoff program.

Then a further supply is obtained by taking computable images of computationally compact sets.

One can apply Tychonoff again, and so on.

Question: Is every computationally compact set a computable image of the Cantor space?

Other computational versions of topological notions

Open, closed, Hausdorff, discrete. (Cf. Taylor's ASD.)

They interact with computational compactness as expected.

In fact, with more transparent proofs (using λ-calculus).

If there is time, I'll use the blackboard to give you some examples.

Application to non-deterministic computation

Must testing:

Given

- a non-deterministic program P with values on X,
- a semi-decidable set $U \subseteq X$,

semi-decide whether the output of P must land in U.

Obstable: Such a program has infinitely many outputs in general.

But: The outputs form a computationally compact set.

Hence: Must-testing is semi-decidable.

This is a rather simplified story, suitable for a tired audience.

Application to probabilistic computation

Given

- a probabilistic program P with values on X, and
- a semi-decidable set $U \subseteq X$,

(lower semi-)compute the probability that the output of P lands in U.

This is again possible, again using a compactness argument, and a more general version of integration.

To conclude (well, almost: there is another slide)

1. Perhaps surprisingly, there are infinite sets over which one can universally quantify in a computable fashion.

To conclude (well, almost: there is another slide)

1. Perhaps surprisingly, there are infinite sets over which one can universally quantify in a computable fashion.

2. Not only they include usual topologically compact sets,

$2^{\mathbb{N}}$, $[0, 1]$, their countable products, images etc.,

To conclude (well, almost: there is another slide)

1. Perhaps surprisingly, there are infinite sets over which one can universally quantify in a computable fashion.

2. Not only they include usual topologically compact sets,

$$2^{\mathbb{N}},\ [0,1], \text{ their countable products, images etc.},$$

3. but also they must be topologically compact, and hence they exclude all non-compact sets of analysis and topology.

To conclude (well, almost: there is another slide)

1. Perhaps surprisingly, there are infinite sets over which one can universally quantify in a computable fashion.

2. Not only they include usual topologically compact sets,

$$2^{\mathbb{N}},\ [0,1], \text{ their countable products, images etc.},$$

3. but also they must be topologically compact, and hence they exclude all non-compact sets of analysis and topology.

4. A definition of computational compactness is based on a reformulation of topological compactness that avoids open coverings.

To conclude (well, almost: there is another slide)

1. Perhaps surprisingly, there are infinite sets over which one can universally quantify in a computable fashion.

2. Not only they include usual topologically compact sets,

$2^{\mathbb{N}}$, $[0, 1]$, their countable products, images etc.,

3. but also they must be topologically compact, and hence they exclude all non-compact sets of analysis and topology.

4. A definition of computational compactness is based on a reformulation of topological compactness that avoids open coverings.

5. For experts: notice how we have avoided the Scott topology in the discussion, using projection maps $Y \times X \to Y$ instead. But.

Self-references

1. Synthetic topology of data types and classical spaces
ENTCS, Elsevier, volume 87, pages 21-156, 2004.

2. Operational domain theory and topology of a sequential programming language. With W.K. Ho. In Proceedings of the 20th Annual IEEE Symposium on Logic in Computer Science, June 2005, pages 427-436.

3. Notes on compactness. Published on the web in 2005.

4. `http://www.cs.bham.ac.uk/~mhe/`