Searchable types

A type $X : U$ is **searchable** if, given any decidable predicate

$$(p, d) : \Sigma\, p : X \to U, \left(\Pi\, x : X\,, p(x) + \neg p(x)\right),$$

we can find some $x : X$ such that,

$$\left(\Sigma\, \mathrm{x}_0 : \mathrm{X}, \mathrm{p}(\mathrm{x}_0)\right) \Rightarrow p(x).$$

```
predicate : (X : 𝓤 ̇) → (𝓤₀ ⁺) ⊔ 𝓤 ̇
predicate X = X → 𝓤₀ ̇

decidable : 𝓤 ̇→ 𝓤 ̇
decidable X = X + ¬ X

everywhere-decidable : {X : 𝓤 } → predicate X → 𝓤 ̇
everywhere-decidable {𝓤} {X} p = Π x : X , decidable (p x)

d-predicate : 𝓤 ̇→ (𝓤₀ ⁺) ⊔ 𝓤 ̇
d-predicate X = Σ p : (X → 𝓤₀ ̇) , everywhere-decidable p
```

```
searchable : 𝓤 ̇→ (𝓤₀ ⁺) ⊔ 𝓤 ̇
searchable X = Π (p , _) : d-predicate X , Σ x : X , (Σ x₀ : X , p x₀ → p x)
```

What types are(n't) searchable?

- Any finite type, e.g. $2 = \{0,1\}$, is trivially searchable.
- The natural numbers are **not** constructively searchable.

```
ℕ-searchable-implies-LPO : searchable ℕ → LPO
```

- Martín Escardó's "seemingly impossible searchable program" (defined in Agda and formalised in Haskell) tells us that $N \to 2$ is searchable.
- In fact, searchable types are intuitively closed under countable products (**Tychonoff theorem**).
  - This says if we have types $\mathrm{T} : N \to U$, all of which are searchable, then $\Pi\ \mathrm{T} : \mathrm{U}$ is also searchable.

Classical logic in current Tychonoff proof

- Haskell's meta-theory relies on a Brouwerian continuity principal supposing that all defined functions are continuous.
- Agda's termination checker cannot use such a principal: we either require explicit continuity, or we turn the termination checker off.
- Martín's Agda proof that $N \to 2$ is searchable uses an *explicit modulus of uniform continuity* for the searched predicate – this is **safe**.
- However, his proof that searchable types are closed under countable products requires the Agda termination checker to be turned off – this is **unsafe**.

Explicit modulus of continuity

- Martín's Agda proof that $N \to 2$ is searchable uses an *explicit modulus of uniform continuity* for the searched predicate – this is **safe**.
- This modulus of continuity is defined as the $\delta : N$ for $p : (N \to 2) \to U$ such that,

$$\Pi\, \alpha, \beta : (N \to 2),\ \left(x =^{\delta} y\right) \Rightarrow \left(p(\alpha) \Rightarrow p(\beta)\right).$$

```
_≡⟦_⟧_ : {X : 𝓤 ̇} → (ℕ → X) → ℕ → (ℕ → X) → 𝓤 ̇
α ≡⟦ m ⟧ β = Π k ꞉ ℕ , (k ≤ℕ m → α k ≡ β k)

is-u-continuous-𝟚ᴺ : ((ℕ → 𝟚) → 𝓤₀ ̇) → 𝓤₀ ̇
is-u-continuous-𝟚ᴺ p = Σ m ꞉ ℕ , ((α β : ℕ → 𝟚) → α ≡⟦ m ⟧ β → p α → p β)
```

Uniform notion of uniform continuity

- To give a constructive formulation of Tychonoff, we need to give an *explicit modulus of uniform continuity*, but for a wide variety of types; including for infinite collections (Π-types) of those such types.

- For this purpose, we define **closeness functions**

$$c : X \times X \to N\infty,$$

using the type of extended naturals N_∞.

Extended naturals

- N_∞ is the type of decreasing binary sequences – this encodes the natural numbers extended with a point at infinity.

```agda
_≥₂_ : 𝟚 → 𝟚 → 𝓤₀ ̇
a ≥₂ b = b ≡ ₁ → a ≡ ₁

decreasing-binary-seq : (ℕ → 𝟚) → 𝓤₀ ̇
decreasing-binary-seq α = Π n ꞉ ℕ , α n ≥₂ α (succ n)

ℕ∞ : 𝓤₀ ̇
ℕ∞ = Σ decreasing-binary-seq
```

```agda
_≼_ : ℕ∞ → ℕ∞ → 𝓤₀ ̇
(α , _) ≼ (β , _) = Π n ꞉ ℕ , (α n ≡ ₁ → β n ≡ ₁)

0-minimal : (α : ℕ∞) → (0 ↑) ≼ α
0-minimal α k ()

∞-maximal : (α : ℕ∞) → α ≼ ∞
∞-maximal α k αₖ≡₁ = refl
```

- For any $n : N$ we have $n \uparrow : N_\infty$, e.g. $5 \uparrow = 11111000000000 \ldots$
- We also have $\infty : N_\infty$, i.e. $\infty = 11111111111111 \ldots$
- Finally, there is an order relation $\preccurlyeq : N_\infty \to N_\infty \to U$.

Closeness functions

- A closeness function $c : X \times X \to N_\infty$ is defined by properties:

```
record is-clofun {X : 𝓤 ̇} (c : X × X → ℕ∞) : 𝓤 ̇ where
  field
    equal→inf-close : (x     : X) → c (x , x) ≡ ∞
    inf-close→equal : (x y   : X) → c (x , y) ≡ ∞ → x ≡ y
    symmetricity : (x y   : X) → c (x , y) ≡ c (y , x)
    ultrametric : (x y z : X) → min (c (x , y)) (c (y , z)) ≼ c (x , z)
```

- Two elements $x, y : X$ of a type with closeness function $c : X \times X \to N_\infty$ are δ-close, for $\delta : N$ if and only if,
$$\delta \uparrow \preccurlyeq c(x, y).$$
- Those elements are instead ∞-close if and only if,
$$\infty \preccurlyeq c(x, y).$$

Uniformly continuous predicates

- Recall Martín's modulus of uniform continuity on $N \to 2$ was defined as the $\delta : N$ for $p : (N \to 2) \to U$ such that,

$$\Pi\ \alpha, \beta : (N \to 2),\ \left(x =^{\delta} y\right) \Rightarrow \left(p(\alpha) \Rightarrow p(\beta)\right).$$

- Using closeness functions $c : X \times X \to N_\infty$, we instead define the modulus of uniform continuity $\delta : N$ for $p : X \to U$ such that,

$$\Pi\ \alpha, \beta : (N \to 2), \left(\delta \uparrow \preccurlyeq c(x, y)\right) \Rightarrow \left(p(\alpha) \Rightarrow p(\beta)\right).$$

```
_is-u-mod-of_on_ : {X : 𝓤 ̇} → ℕ → predicate X → (X × X → ℕ∞) → 𝓤 ̇
_is-u-mod-of_on_ {𝓤} {X} δ p c = Π (x , y) : (X × X) , ((δ ↑) ≼ c (x , y) → p x → p y)

u-continuous : {X : 𝓤 ̇} → (X × X → ℕ∞) → predicate X → 𝓤 ̇
u-continuous {𝓤} {X} c p = Σ δ : ℕ , δ is-u-mod-of p on c
```

Canonical closeness functions

- There is a closeness function $c_d : X \times X \to N_\infty$ for every discrete type X, where:

$$c_d(x, y) = \infty \qquad \text{if } x = y,$$
$$c_d(x, y) = 0 \uparrow \qquad \text{otherwise.}$$

Every predicate on discrete X is uniformly continuous on c_d with modulus 1.

- There is a closeness function $c : (N \to X) \times (N \to X) \to N_\infty$ for every sequence type of a discrete type X, where

$$c_{ds}(x, y)_n = 1 \qquad \text{if } x =^n y,$$
$$c_{ds}(x, y)_n = 0 \qquad \text{otherwise.}$$

Every predicate on discrete-sequence $N \to X$ is uniformly continuous on c_{ds} if and only if it is uniformly continuous in the previous sense.

From searchable to continuously searchable

A type $X : U$ with clofun $c : X \times X \to N_\infty$ is **continuously searchable** if, given any uniformly continuous decidable predicate

$$(p, d, \delta, \phi) : \Sigma\, p : X \to U, \left(\Pi\, x : X\,, p(x) + \neg p(x)\right), (\Sigma \delta : N\,, \delta\ is{-}u{-}mod{-}of\, p\ on\ c)$$

we can find some $x : X$ such that,

$$\left(\Sigma\, \mathrm{x}_0 : \mathrm{X}, \mathrm{p}(\mathrm{x}_0)\right) \Rightarrow p(x).$$

```
uc-d-predicate : (X : 𝓤 ̇) → (X × X → ℕ∞) → (𝓤₀ ⁺) ⊔ 𝓤 ̇
uc-d-predicate X c = Σ (p , d) : d-predicate X , u-continuous c p

c-searchable : (X : 𝓤 ̇) → (X × X → ℕ∞) → (𝓤₀ ⁺) ⊔ 𝓤 ̇
c-searchable X c = Π ((p , _)  , _) : uc-d-predicate X c , Σ x₀ : X , (Σ p → p x₀)
```

Continuously searchable types

- Every searchable type is trivially continuously searchable.
- Every continuously searchable discrete type is searchable.

- We will now prove that every discrete-sequence type $N \to X$ is continuously searchable using the discrete-sequence closeness function c_{ds}.
  - The proof is by induction on the modulus of uniform continuity δ of the predicate p being searched (i.e. using the same method as Escardó, as the two notions of uniform continuity are equivalent).
- Furthermore, we will safely formalise the Tychonoff theorem – that states continuously searchable types are closed under countable products – in Agda.

Splitting an infinite predicate in two

- Given a predicate p on an infinite sequence, we define:
  - $(p_t\ x) \coloneqq \lambda xs.\, p(x :: xs)$, the **tail-predicate for $\boldsymbol{p}$ via $\boldsymbol{x}$**, for any $x : X$,
  - $p_h \coloneqq \lambda x.\ p(x :: \mathcal{E}xs\ x)$, the **head-predicate for $\boldsymbol{p}$**,

    where $\mathcal{E}xs\ x : N \to X$ is the sequence satisfying the tail-predicate for p via x.

- So, we use the searcher for X to find an $x : X$ satisfying the head-predicate for p, which in turn calls the searcher for $(N \to X)$ **inductively** to find some $xs : X$ satisfying the tail-predicate for p via x.