Discrete Mathematics with Applications I

COMPSCI&SFWRENG 2DM3

McMaster University, Fall 2019

Wolfram Kahl

2019-12-03

Counting ...

Let A and B be finite sets with $\# A = a$ and $\# B = b$:

- $\# (A \times B) \ = \ ?$ — pairs
- $\# (A \leftrightarrow B) \ = \ \# (\mathbb{P} (A \times B)) \ = \ ?$ — relations
- $\# (A \longrightarrow B) \ = \ ?$ — total functions
- $\# (A \nrightarrow B) \ = \ ?$ — partial functions
- $\# (A \rightarrowtail\!\!\!\twoheadrightarrow A) \ = \ ?$ — homogeneous total bijections
- $\# (A \rightarrowtail\!\!\!\twoheadrightarrow B) \ = \ ?$ — total bijections
- $\# (A \rightarrowtail B) \ = \ ?$ — total injections
- $\# (A \nrightarrow\!\!\!\twoheadrightarrow B) \ = \ ?$ — partial bijections
- $\# (A \nrightarrow B) \ = \ ?$ — partial injections
- $\# (A \twoheadrightarrow B) \ = \ ?$ — total surjections
- $\# \{ S \mid S \subseteq B \land \# S = a \} \ = \ ?$ — a-combinations of B

Plan for Today

- **Kernels — dual to Closures**
- **Combinatorial Analysis — “Counting”** (LADM chapter 16)
  - Permutations, Combinations
- M2

Plan for Tomorrow

- **Topological Sort** (LADM section 14.4)
  - An example for algorithm development based on discrete math

Recall: Closures

Let Ω be a property on relations, i.e.:

$$\Omega \quad : \quad (B \leftrightarrow C) \to \mathbb{B}$$

Relation $Q : B \leftrightarrow C$ is the **Ω-closure** of $R : B \leftrightarrow C$ iff

- Q is the smallest relation
- that contains R
- and has property Ω

or, equivalently, iff

- $R \subseteq Q$
- $\Omega\ Q$
- $(\forall P : B \leftrightarrow C \mid R \subseteq P \wedge \Omega\ P \bullet Q \subseteq P)$

(For some properties, closures are not defined, or not always defined.)

Kernels

Let Ω be a property on relations, i.e.:

$$\Omega \quad : \quad (B \leftrightarrow C) \to \mathbb{B}$$

Relation $Q : B \leftrightarrow C$ is the **Ω-kernel** of $R : B \leftrightarrow C$ iff

- Q is the largest relation
- contained in R
- that has property Ω

or, equivalently, iff

- $Q \subseteq R$
- $\Omega\ Q$
- $(\forall P : B \leftrightarrow C \mid P \subseteq R \wedge \Omega\ P \bullet P \subseteq Q)$

(For some properties, kernels are not defined, or not always defined.)

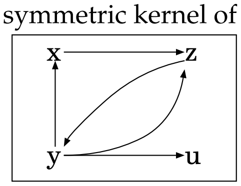
Symmetric Kernel

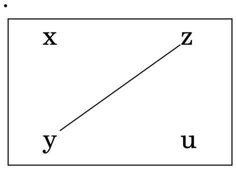
Relation $Q : B \leftrightarrow B$ is the **symmetric kernel** of $R : B \leftrightarrow B$
iff Q is the largest symmetric relation contained in R,

or, equivalently, iff

- $Q \subseteq R$
- $Q = Q^{\smile}$
- $(\forall P : B \leftrightarrow B \mid P \subseteq R \wedge P = P^{\smile} \bullet P \subseteq Q)$

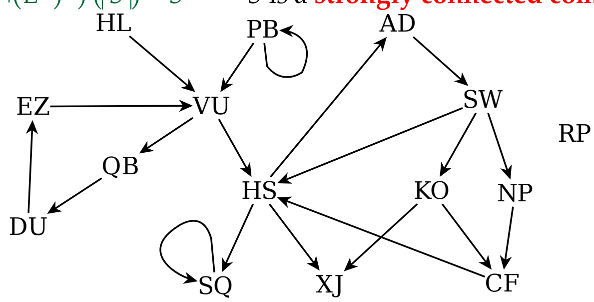
Theorem: The symmetric kernel of $R : B \leftrightarrow B$ is $R \cap R^{\smile}$.





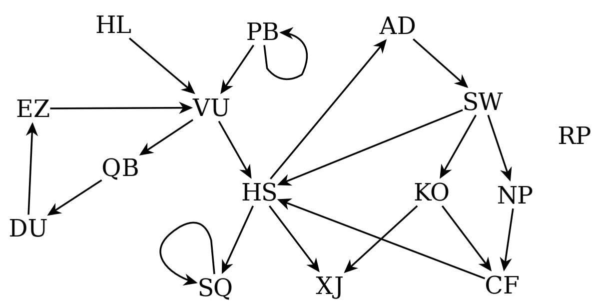
Recall: Reachability in Simple Graph $G = (V, E)$ — 2

- From every node, each node is reachable
  $V \times V \subseteq E^*$ or $\sim \mathrm{Id} \subseteq E^+$ — G is **strongly connected**
- From every node, each node is reachable by traversing edges in either direction
  $V \times V \subseteq (E \cup E^\smile)^*$ or $\sim \mathrm{Id} \subseteq (E \cup E^\smile)^+$ — G is **connected**
- Nodes n_1 and n_2 reachable from each other both ways
  $n_1 \,(E^* \cap (E^*)^\smile)\, n_2$ — n_1 and n_2 are **strongly connected**
- S is an equivalence class of strong connectedness between nodes
  $S \times S \subseteq E^* \;\wedge\; (E^* \cap (E^*)^\smile)\,(\!| S |\!) = S$ — S is a **strongly connected component (SCC)** of G



Strong Connectedness in Simple Graph $G = (V, E)$

- Nodes n_1 and n_2 reachable from each other both ways
  $n_1 \,(E^* \cap (E^*)^\smile)\, n_2$ — n_1 and n_2 are **strongly connected**
- The strong-connectedness relation $E^* \cap (E^*)^\smile$ is the symmetric kernel of E^*
- Due to the properties of reflexive-transitive closure this is an equivalence relation.
- Its equivalence classes are called **strongly connected components (SCCs)** of G
- The SCCs form a **partition** of the vertex set of G.



Graphs

Definition: A **graph** is a tuple $(V, E, \mathsf{src}, \mathsf{trg})$ consisting of

- a set V of ***vertices*** or ***nodes***
- a set E of ***edges*** or ***arrows***
- a mapping $\mathsf{src} : E \mathrel{-\!\!\circ\!\!\rightarrow} V$ that assigns each edge its ***source*** node
- a mapping $\mathsf{trg} : E \mathrel{-\!\!\circ\!\!\rightarrow} V$ that assigns each edge its ***target*** node

Example graph:

$$(\ \{x, y, z\},\ \{a, b, c, d\},\ \{\langle a, x\rangle, \langle b, z\rangle, \langle c, z\rangle, \langle d, x\rangle\},\ \{\langle a, y\rangle, \langle b, y\rangle, \langle c, z\rangle, \langle d, y\rangle\}\ )$$

Graphs, Subgraphs

Definition: A graph is a tuple $(V, E, \mathsf{src}, \mathsf{trg})$ consisting of

- a set V of ***vertices*** or ***nodes***
- a set E of ***edges*** or ***arrows***
- a mapping $\mathsf{src} : E \longrightarrow V$ that assigns each edge its ***source*** node
- a mapping $\mathsf{trg} : E \longrightarrow V$ that assigns each edge its ***target*** node

Definition: Let two graphs $G_1 = (V_1, E_1, \mathsf{src}_1, \mathsf{trg}_1)$ and $G_2 = (V_2, E_2, \mathsf{src}_2, \mathsf{trg}_2)$ be given.

- G_1 is called a ***subgraph*** of G_2 iff $V_1 \subseteq V_2$ and $E_1 \subseteq E_2$ and $\mathsf{src}_1 \subseteq \mathsf{src}_2$ and $\mathsf{trg}_1 \subseteq \mathsf{trg}_2$.
- We write $\mathsf{Subgraph}_G$ for the set of all subgraphs of G.
- For a given graph G, we write $G_1 \sqsubseteq_G G_2$ if both G_1 and G_2 are subgraphs of G, and G_1 is a subgraph of G_2.

Def. and Theorem: Given a subset $V_0 \subseteq V$ of the vertex set of graph $G = (V, E, \mathsf{src}, \mathsf{trg})$, the edges incident with only nodes in V_0 are $E_0 := E \cap \mathsf{src}^{\smile} (\!| V_0 |\!) \cap \mathsf{trg}^{\smile} (\!| V_0 |\!)$, and then $G_0 := (V_0,\ E_0,\ E_0 \lhd \mathsf{src},\ E_0 \lhd \mathsf{trg})$ is called the ***subgraph of*** G ***induced by*** V_0.
It is a graph, and a subgraph of G. — **Induced subgraphs are well-defined**

Theorem: $\sqsubseteq_G$ is an ordering on $\mathsf{Subgraph}_G$.

Theorem: $\sqsubseteq_G$ has binary meets defined by intersection.

...

Rules of Sum, Product, and Difference — LADM p. 337

(16.1) **Rule of sum:** The size of the union of n finite pairwise-disjoint sets is the sum of their sizes:

$$(\forall i, j : \mathbb{N} \mid i < j < n \bullet S_i \cap S_j = \{\}) \quad \Rightarrow \quad \#(\bigcup i : \mathbb{N} \mid i < n \bullet S_i) = (\sum i \mid i < n \bullet \# S_i)$$

(16.2) **Rule of product:** The size of the Cartesian product of n finite sets is the product of their sizes:

$$\#(S_0 \times \cdots \times S_{n-1}) \quad = \quad (\prod i \mid i < n \bullet \# S_i)$$

(16.3) **Rule of difference:** The size of a set with a subset of it removed is the size of the set minus the size of the subset:

$$T \subseteq S \quad \Rightarrow \quad \#(S - T) = \# S - \# T$$

(11.73) **Size of power set:** If S is a finite set, then: $\#(\mathbb{P}\, S) = 2^{\# S}$

Size of isomorphic sets: $(\exists f \bullet f \in A \rightarrowtail\!\!\!\!\twoheadrightarrow B) \quad \equiv \quad \# A = \# B$

Counting ...

Let A and B be finite sets with $\# A = a$ and $\# B = b$:

- $\#(A \times B) \ = \ ?$ — pairs
- $\#(A \leftrightarrow B) \ = \ \#(\mathbb{P}(A \times B)) \ = \ ?$ — relations
- $\#(A \longrightarrow B) \ = \ ?$ — total functions
- $\#(A \nrightarrow B) \ = \ ?$ — partial functions

Permutations

LADM, p. 338: A **permutation** of a set of elements (or of a sequence of elements) is a linear ordering of the elements.
For example, two permutations of the set $\{5, 4, 1\}$ are $[1, 4, 5]$ and $[1, 5, 4]$.

Theorem: If $\# S = n$, then there are $n!$ permutations of S.

Observation: Permutations on A can be seen as bijective mappings on A.

- $\# (A \rightarrowtail\!\!\!\!\twoheadrightarrow A) \ = \ ?$ — homogeneous total bijections
- $\# (A \rightarrowtail\!\!\!\!\twoheadrightarrow B) \ = \ ?$ — total bijections

r-Permutations

For $r : \mathbb{N}$, an **r-permutation of** S is a permutation of an r-sized subset of S.
(16.4) $P(n, r) = n!/(n - r)!$
(16.5) **Theorem:** The number of r-permutations of a set of size n equals $P(n, r)$.

Observation: r-permutations on S "are" injective mappings from $\{0, \ldots, r - 1\}$ to S.

Definition: $\mathsf{Fin}\ n \ = \ \{i : \mathbb{N} \mid i < n\}$

Corollary: $\# (\mathsf{Fin}\ n) \ = \ n$

Corollary: The set of r-permutations on S can be modelled as the set $\mathsf{Fin}\ r \rightarrowtail S$

Equivalent presentations: $P(n, r) = n!/(n - r)! \quad \equiv \quad (n - r)! \cdot P(n, r) = n! \quad \equiv \quad P(n, r) = (\prod i \mid i < r \bullet n - i) \quad \equiv \quad P(n, r) = (\prod j \mid n - r < j \le n \bullet j)$

- $\# (A \rightarrowtail B) \ = \ ?$ — total injections
- $\# (A \rightarrowtail\!\!\!\!\twoheadrightarrow\!\!\!\!\!\!\!+ B) \ = \ ?$ — partial bijections

Permutations of a Bag

All the permutations of the set { S, O, N }
SON, SNO, OSN, ONS, NSO, NOS

All permuations of the bag ≀ M, O, M ∫:
MOM, MMO, OMM

(16.7) **Theorem:** The number of permutations of a bag of size n with k distinct elements occurring $n_1, n_2, \ldots, n_k$ times is:

$$\frac{n!}{n_1! \cdot n_2! \cdot \cdots \cdot n_k!}$$

Proof: By induction on k ...

r-Permutations with Repetition

LADM, p. 339: Consider forming an r-permutation of a set but allowing each element to be used more than once. Such a permutation is called an r-permutation with repetition. For example, here are all the 2-permutations with repetition of the letters in SON:

SS, SO, SN, OS, 00, ON, NS, NO, NN.

Given a set of size n, in constructing an r-permutation with repetition, for each element we have n choices. The following theorem follows trivially from this observation and the rule of product.

(16.6) **Theorem:** The number of r-permutations with repetition of a set of size n is n^r.

Observation: An r-permutation of S "is" a total function in $\mathrm{Fin}\ r \multimap\!\!\!\rightarrow S$.

- $\#\,(A \multimap\!\!\!\rightarrow B) \;=\; ?$ — total functions

r-Combinations — LADM p. 340

- An r-combination of a set is a subset of size r.
- A permutation is a sequence; a combination is a set.
- For example, the 2-permutations of the set consisting of the letters in SOHN are

  SO,SH,SN,OH,ON,OS,HN,HS,HO,NS,NO,NH

  while the 2-combinations are

  $\{S, O\}, \{S, H\}, \{S, N\}, \{O, H\}, \{O, N\}, \{H, N\}$

(16.9) **Definition:** The binomial coefficient $\binom{n}{r}$, which is read as "n choose r", is defined by:

$$\binom{n}{r} \quad = \quad \frac{n!}{r! \cdot (n-r)!} \qquad (\text{for } 0 \le r \le n)$$

(16.10) **Theorem:** The number of r-combinations of n elements is $\binom{n}{r}$.

- $\#\,\{\, S \mid S \subseteq B \wedge \# S = a \,\} \;=\; ?$ — a-combinations of B

Counting Challenges

Let A and B be finite sets with $\# A = a$ and $\# B = b$:

- $\#\,(A \times B) \;=\; ?$ — pairs
- $\#\,(A \leftrightarrow B) \;=\; \#\,(\mathbb{P}\,(A \times B)) \;=\; ?$ — relations
- $\#\,(A \multimap\!\!\!\rightarrow B) \;=\; ?$ — total functions
- $\#\,(A \nrightarrow B) \;=\; ?$ — partial functions
- $\#\,(A \rightarrowtail\!\!\!\!\twoheadrightarrow A) \;=\; ?$ — homogeneous total bijections
- $\#\,(A \rightarrowtail\!\!\!\!\twoheadrightarrow B) \;=\; ?$ — total bijections
- $\#\,(A \rightarrowtail B) \;=\; ?$ — total injections
- $\#\,(A \nrightarrow\!\!\!\!\twoheadrightarrow B) \;=\; ?$ — partial bijections
- $\#\,\{\, S \mid S \subseteq B \wedge \# S = a \,\} \;=\; ?$ — a-combinations of B

- $\#\,(A \nrightarrow\!\!\!\!\rightarrowtail B) \;=\; ?$ — partial injections

- $\#\,(A \twoheadrightarrow B) \;=\; ?$ — total surjections

M2.1

```
Lemma “Relationship via ×”: x ⦇ S × T ⦈ y  ≡  x ∈ S ∧ y ∈ T
Proof:
    x ⦇ S × T ⦈ y
  ≡⟨ “Definition of `_⦇_⦈_`” ⟩
    ⟨ x, y ⟩ ∈ S × T
  ≡⟨ “Membership in ×” ⟩
    x ∈ S ∧ y ∈ T
```

M2.1A

```
Theorem (14.8) “Distributivity of × over ∪”:
    S × (T ∪ U) = (S × T) ∪ (S × U)
Proof:
  Using “Relation extensionality”:
    For any `x, y`:
        x ⦇ S × (T ∪ U) ⦈ y
      ≡⟨ “Relationship via ×” ⟩
        x ∈ S ∧ y ∈ T ∪ U
      ≡⟨ “Union” ⟩
        x ∈ S ∧ (y ∈ T ∨ y ∈ U)
      ≡⟨ “Distributivity of ∧ over ∨” ⟩
        (x ∈ S ∧ y ∈ T) ∨ (x ∈ S ∧ y ∈ U)
      ≡⟨ “Relationship via ×” ⟩
        x ⦇ S × T ⦈ y ∨ x ⦇ S × U ⦈ y
      ≡⟨ “Relation union” ⟩
        x ⦇ (S × T) ∪ (S × U) ⦈ y
```

M2.1A

```
Theorem (14.8) “Distributivity of × over ∪”:
    S × (T ∪ U) = (S × T) ∪ (S × U)
Proof:
  Using “Set extensionality”:
    For any `p`:
        p ∈ S × (T ∪ U)
      ≡⟨ “Membership in ×” ⟩
        fst p ∈ S ∧ snd p ∈ T ∪ U
      ≡⟨ “Union” ⟩
        fst p ∈ S ∧ (snd p ∈ T ∨ snd p ∈ U)
      ≡⟨ “Distributivity of ∧ over ∨” ⟩
        (fst p ∈ S ∧ snd p ∈ T) ∨ (fst p ∈ S ∧ snd p ∈ U)
      ≡⟨ “Membership in ×” ⟩
        p ∈ S × T  ∨  p ∈ S × U
      ≡⟨ “Union” ⟩
        p ∈ (S × T) ∪ (S × U)
```

M2.1B

```
Theorem (14.9) “Distributivity of × over ∩”:
    S × (T ∩ U) = (S × T) ∩ (S × U)
Proof:
  Using “Relation extensionality”:
    For any `x, y`:
        x ⦇ S × (T ∩ U) ⦈ y
      ≡⟨ “Relationship via ×” ⟩
        x ∈ S ∧ y ∈ T ∩ U
      ≡⟨ “Intersection” ⟩
        x ∈ S ∧ y ∈ T ∧ y ∈ U
      ≡⟨ “Distributivity of ∧ over ∧” ⟩
        x ∈ S ∧ y ∈ T ∧ x ∈ S ∧ y ∈ U
      ≡⟨ “Relationship via ×” ⟩
        x ⦇ S × T ⦈ y ∧ x ⦇ S × U ⦈ y
      ≡⟨ “Relation intersection” ⟩
        x ⦇ (S × T) ∩ (S × U) ⦈ y
```

M2.1

```
Theorem “Non-empty sets”: S ≠ {} ≡ (∃ x • x ∈ S)
Proof:
    S ≠ {}
  ≡⟨ “Definition of ≠” ⟩
    ¬ (S = {})
  ≡⟨ “Set extensionality” ⟩
    ¬ (∀ x • x ∈ S ≡ x ∈ {})
  ≡⟨ “Empty set” ⟩
    ¬ (∀ x • x ∈ S ≡ false)
  ≡⟨ “Definition of ¬ from ≡” ⟩
    ¬ (∀ x • ¬ (x ∈ S))
  ≡⟨ “Generalised De Morgan” ⟩
    (∃ x • x ∈ S)
```

M2.1A

```
Theorem (1MA): T ≠ {} ⇒ S × T = T × S ⇒ S ⊆ T
Proof:
  Assuming `T ≠ {}` and using with “Non-empty sets”,
           `S × T = T × S` and using with “Set extensionality”:
    Assuming witness `y` satisfying `y ∈ T` by assumption `T ≠ {}`:
      Using “Set inclusion”:
        For any `z`:
            z ∈ S
          ≡⟨ Assumption `y ∈ T`, “Identity of ∧” ⟩
            z ∈ S ∧ y ∈ T
          ≡⟨ “Membership in ×” ⟩
            ⟨ z , y ⟩ ∈ S × T
          ≡⟨ Assumption `S × T = T × S` ⟩
            ⟨ z , y ⟩ ∈ T × S
          ≡⟨ “Membership in ×” ⟩
            z ∈ T ∧ y ∈ S
          ⇒⟨ “Weakening” ⟩
            z ∈ T
```

M2.1B

```
Theorem (14.13): S ≠ {}  ⇒  S × T ⊆ S × U  ⇒  T ⊆ U
Proof:
  Assuming `S ≠ {}` and using with “Non-empty sets”:
    Assuming `S × T ⊆ S × U` and using with “Relation inclusion”:
      Assuming witness `x` satisfying `x ∈ S` by assumption `S ≠ {}`:
        Using “Set inclusion”:
          For any `y`:
              y ∈ T
            ≡⟨ Assumption `x ∈ S`, “Identity of ∧” ⟩
              x ∈ S ∧ y ∈ T
            ≡⟨ “Relationship via ×” ⟩
              x ( S × T ) y
            ⇒⟨ Assumption `S × T ⊆ S × U` ⟩
              x ( S × U ) y
            ⇒⟨ “Relationship via ×”, “Weakening” ⟩
              y ∈ U
```

M2.1B

```
Theorem (14.13): S ≠ {}  ⇒  S × T ⊆ S × U  ⇒  T ⊆ U
Proof:
  Assuming `S ≠ {}` and using with “Non-empty sets”:
    Assuming `S × T ⊆ S × U` and using with “Set inclusion”:
      Assuming witness `x` satisfying `x ∈ S` by assumption `S ≠ {}`:
        Using “Set inclusion”:
          For any `y`:
              y ∈ T
            ≡⟨ Assumption `x ∈ S`, “Identity of ∧” ⟩
              x ∈ S ∧ y ∈ T
            ≡⟨ “Membership in ×” ⟩
              ⟨ x, y ⟩ ∈ S × T
            ⇒⟨ Assumption `S × T ⊆ S × U` ⟩
              ⟨ x, y ⟩ ∈ S × U
            ⇒⟨ “Membership in ×”, “Weakening” ⟩
              y ∈ U
```

M2.2

the following definition of the predicate `is-sorted` on sequences of **integers**, where “`is-sorted xs`” means that the sequence `xs` is sorted in ascending order.

For example, the sequence “`1 ◃ 2 ◃ 2 ◃ 3 ◃ ϵ`” is sorted in this sense.

```
Declaration: is-sorted : Seq ℤ → 𝔹
Axiom “is-sorted ϵ”:           is-sorted ϵ
Axiom “is-sorted singleton”:  is-sorted (x ◃ ϵ)
Axiom “is-sorted ◃◃”:          is-sorted (x ◃ y ◃ ys)  ≡  x ≤ y ∧ is-sorted (y ◃ ys)
```

Throughout this question, all sequence elements are of type ℤ!

---

a definition of `insert` intended **for sorted lists of integers**:

```
Declaration: insert : ℤ → Seq ℤ → Seq ℤ

Axiom “insert ϵ”: insert x ϵ = x ◃ ϵ
Axiom “insert before ◃”: x ≤ y ⇒ insert x (y ◃ ys) = x ◃ (y ◃ ys)
Axiom “insert after ◃”: x > y ⇒ insert x (y ◃ ys) = y ◃ insert x ys
```

For example:

```
insert 3 (1 ◃ 4 ◃ ϵ) = (1 ◃ 3 ◃ 4 ◃ ϵ)
```

Throughout this question, all sequence elements are of type ℤ!

M2.2A

```
Theorem (2A1): ∃ x : ℤ • ∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x)
Proof:
    ∃ x : ℤ • ∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x)
  ⇐⟨ “∃-Introduction” ⟩
    (∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x))[x ≔ 1]
  ≡⟨ Substitution ⟩
    ∃ xs : Seq ℤ • is-sorted (1 ◃ xs ▹ 1)
  ⇐⟨ “∃-Introduction” ⟩
    (is-sorted (1 ◃ xs ▹ 1))[xs ≔ ϵ]
  ≡⟨ Substitution ⟩
    is-sorted (1 ◃ ϵ ▹ 1)
  ≡⟨ “Definition of ▹ for ◃” ⟩
    is-sorted (1 ◃ (ϵ ▹ 1))
  ≡⟨ “Definition of ▹ for ϵ” ⟩
    is-sorted (1 ◃ (1 ◃ ϵ))
  ≡⟨ “is-sorted ◃◃” ⟩
    1 ≤ 1 ∧ is-sorted (1 ◃ ϵ)
  ≡⟨ “is-sorted singleton” ⟩
    1 ≤ 1 ∧ true
  ≡⟨ Evaluation ⟩
    true
```

M2.2A

```
Theorem (2A1): ∃ x : ℤ • ∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x)
Proof:
    ∃ x : ℤ • ∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x)
  ⇐⟨ “∃-Introduction” ⟩
    (∃ xs : Seq ℤ • is-sorted (x ◃ xs ▹ x))[x ≔ 0]
  ≡⟨ Substitution ⟩
    ∃ xs : Seq ℤ • is-sorted (0 ◃ xs ▹ 0)
  ⇐⟨ “∃-Introduction” ⟩
    (is-sorted (0 ◃ xs ▹ 0))[xs ≔ ϵ]
  ≡⟨ Substitution ⟩
    is-sorted ((0 ◃ ϵ ▹ 0))
  ≡⟨ “Definition of ▹ for ϵ” ⟩
    is-sorted (0 ◃ 0 ◃ ϵ)
  ≡⟨ “is-sorted ◃◃”, Fact `0 ≤ 0`, “Identity of ∧” ⟩
    is-sorted (0 ◃ ϵ)
  ≡⟨ “is-sorted singleton” ⟩
    true
```

M2.2A

```
Theorem (2A2):
  ∃ x : ℤ • ∃ xs : Seq ℤ •
  ¬ is-sorted (x ◃ xs) ⇒ is-sorted (xs ▹ x)
Proof:
    ∃ x : ℤ • ∃ xs : Seq ℤ • ¬ is-sorted (x ◃ xs) ⇒ is-sorted (xs ▹ x)
  ⇐⟨ “∃-Introduction” ⟩
    (∃ xs : Seq ℤ • ¬ is-sorted (x ◃ xs) ⇒ is-sorted (xs ▹ x))[x ≔ 0]
  ≡⟨ Substitution ⟩
    ∃ xs : Seq ℤ • ¬ is-sorted (0 ◃ xs) ⇒ is-sorted (xs ▹ 0)
  ⇐⟨ “∃-Introduction” ⟩
    (¬ is-sorted (0 ◃ xs) ⇒ is-sorted (xs ▹ 0))[xs ≔ ϵ]
  ≡⟨ Substitution ⟩
    ¬ is-sorted (0 ◃ ϵ) ⇒ is-sorted (ϵ ▹ 0)
  ≡⟨ “is-sorted singleton”, “Definition of `false`”, “ex falso quodlibet” ⟩
    true
```

M2.2B

```
Theorem (2B1): ∃ x : ℤ • ∃ xs : Seq  ℤ • xs ⌢ xs = insert x xs
Proof:
    ∃ x : ℤ • ∃ xs : Seq  ℤ • xs ⌢ xs = insert x xs
  ⇐⟨ “∃-Introduction” ⟩
    (∃ xs : Seq  ℤ • xs ⌢ xs = insert x xs)[x ≔ 4]
  ≡⟨ Substitution ⟩
    (∃ xs : Seq  ℤ • xs ⌢ xs = insert 4 xs)
  ⇐⟨ “∃-Introduction” ⟩
    (xs ⌢ xs = insert 4 xs)[xs ≔ 4 ◃ ϵ]
  ≡⟨ Substitution ⟩
    (4 ◃ ϵ) ⌢ (4 ◃ ϵ) = insert 4 (4 ◃ ϵ)
  ≡⟨ “Definition of ⌢ for ϵ”, “Definition of ⌢ for ◃” ⟩
    4 ◃ 4 ◃ ϵ = insert 4 (4 ◃ ϵ)
  ≡⟨ “insert before ◃” with “Reflexivity of ≤” ⟩
    4 ◃ 4 ◃ ϵ = 4 ◃ 4 ◃ ϵ
  ≡⟨ “Reflexivity of =” ⟩
    true
```

M2.2B

```
Theorem (2B2):
  ∃ x : ℤ • ∃ xs : Seq ℤ • xs ≠ x ◃ ϵ ⇒ insert x xs = xs ▹ x
Proof:
    ∃ x : ℤ • ∃ xs : Seq ℤ • xs ≠ x ◃ ϵ ⇒ insert x xs = xs ▹ x
  ⇐⟨ “∃-Introduction” ⟩
    (∃ xs : Seq ℤ • xs ≠ x ◃ ϵ ⇒ insert x xs = xs ▹ x)[x ≔ 3]
  ≡⟨ Substitution ⟩
    (∃ xs : Seq ℤ • xs ≠ 3 ◃ ϵ ⇒ insert 3 xs = xs ▹ 3)
  ⇐⟨ “∃-Introduction” ⟩
    (xs ≠ 3 ◃ ϵ ⇒ insert 3 xs = xs ▹ 3)[xs ≔ 3 ◃ ϵ]
  ≡⟨ Substitution ⟩
    (3 ◃ ϵ ≠ 3 ◃ ϵ ⇒ insert 3 (3 ◃ ϵ) = (3 ◃ ϵ) ▹ 3)
  ≡⟨ “Irreflexivity of ≠” ⟩
    false ⇒ insert 3 (3 ◃ ϵ) = (3 ◃ ϵ) ▹ 3
  — This is “ex falso quodlibet”
```

M2.3A

```
Declaration: sum : Seq ℕ → ℕ
Axiom “Definition of `sum` for ϵ”: sum ϵ = 0
Axiom “Definition of `sum` for ◃”: sum (x ◃ xs) = x + sum xs

Theorem “Initialisation for `Sum`”:
      true
    ⇒[ xs := xs₀ ; s := 0 ]
      s + sum xs = sum xs₀
Proof:
      s + sum xs = sum xs₀
    [ s := 0 ]⇐  ⟨ “Assignment” with substitution ⟩
      0 + sum xs = sum xs₀
    [ xs := xs₀ ]⇐  ⟨ “Assignment” with substitution ⟩
      0 + sum xs₀ = sum xs₀
    ≡⟨ “Left-identity of +” ⟩
      sum xs₀ = sum xs₀
    ≡⟨ “Reflexivity of =” ⟩
      true
```

M2.3A

```
Theorem “Invariant for `Sum`”:
      xs ≠ ϵ ∧ s + sum xs = sum xs₀
    ⇒[ s := s + head xs ; xs := tail xs ]
      s + sum xs = sum xs₀
Proof:
      s + sum xs = sum xs₀
    [ xs := tail xs ]⇐  ⟨ “Assignment” with substitution ⟩
      s + sum (tail xs) = sum xs₀
    [ s := s + head xs ]⇐ ⟨ “Assignment” with substitution ⟩
      s + head xs + sum (tail xs) = sum xs₀
    ≡⟨ “Definition of `sum` for ◃” ⟩
      s + sum (head xs ◃ tail xs) = sum xs₀
    ≡⟨ Substitution ⟩
      (s + sum z = sum xs₀)[z ≔ head xs ◃ tail xs]
    ⇐⟨ “Strengthening” ⟩
      head xs ◃ tail xs = xs  ∧
      (s + sum z = sum xs₀)[z ≔ head xs ◃ tail xs]
    ≡⟨ “Replacement”, substitution ⟩
      xs = head xs ◃ tail xs  ∧  s + sum xs = sum xs₀
    ⇐⟨ “Monotonicity of ∧” with “Non-empty-sequence extensionality” ⟩
      xs ≠ ϵ ∧ s + sum xs = sum xs₀
```

M2.3B

```
Declaration: max : Seq ℕ → ℕ
Axiom “Definition of `max` for ϵ”: max ϵ = 0
Axiom “Definition of `max` for ◃”: max (x ◃ xs) = x ↑ max xs

Theorem “Initialisation for `Maximum`”:
      true
    ⇒[ xs := xs₀ ; m := 0 ]
      m ↑ max xs = max xs₀
Proof:
      m ↑ max xs = max xs₀
    [ m := 0 ]⇐  ⟨ “Assignment” with substitution ⟩
      0 ↑ max xs = max xs₀
    [ xs := xs₀ ]⇐  ⟨ “Assignment” with substitution ⟩
      0 ↑ max xs₀ = max xs₀
    ≡⟨ “Identity of ↑” ⟩
      max xs₀ = max xs₀
    ≡⟨ “Reflexivity of =” ⟩
      true
```

M2.3B

```
Theorem “Invariant for `Maximum`”:
      xs ≠ ϵ ∧ m ↑ max xs = max xs₀
    ⇒[ m := m ↑ head xs ; xs := tail xs ]
      m ↑ max xs = max xs₀
Proof:
      m ↑ max xs = max xs₀
    [ xs := tail xs ]⇐  ⟨ “Assignment” with substitution ⟩
      m ↑ max (tail xs) = max xs₀
    [ m := m ↑ head xs ]⇐ ⟨ “Assignment” with substitution ⟩
      m ↑ head xs ↑ max (tail xs) = max xs₀
    ≡⟨ “Definition of `max` for ◃” ⟩
      m ↑ max (head xs ◃ tail xs) = max xs₀
    ≡⟨ Substitution ⟩
      (m ↑ max z = max xs₀)[z ≔ head xs ◃ tail xs]
    ⇐⟨ “Strengthening” ⟩
      head xs ◃ tail xs = xs  ∧
      (m ↑ max z = max xs₀)[z ≔ head xs ◃ tail xs]
    ≡⟨ “Replacement”, substitution ⟩
      xs = head xs ◃ tail xs  ∧  m ↑ max xs = max xs₀
    ⇐⟨ “Monotonicity of ∧” with “Non-empty-sequence extensionality” ⟩
      xs ≠ ϵ ∧ m ↑ max xs = max xs₀
```