Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

Press ⌘⏎ in the code editor to execute code.

New | Save | Load | Run | Verification ON

In[1]

```
J R N : set

A[ J ] :-
 ψ ↦ fix ((θ i j ↦ min ⟨ψ i j,
                        min (k ↦ θ i k + θ k j + w i k j)⟩)
          : (JxJn< → R) → JxJn< → R)
```

Specification

Out[1]

```
• ψ ↦ fix(θ ↦ i ↦ j ↦ (min (⟨A, B⟩)))                    [ (J → J → R) → J → J → R ]
  • A : (ψ i j)                                          [ R ]
  • B : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))        [ R ]
```

Inferred types for sub-terms

In[2]

```
J₀ J₁ ⊆ J
Slice (find (θ ↦ ?)) (`? ⟨`J₀xJ₀, `J₀xJ₁, `J₁xJ₁⟩)
```

Partition

Tactic application

Out[2]

```
• program(ψ ↦ fix(A / (B / C)))                                  [ (((J x J) ∩ <) → R) → ((J₀ x J₀) ∩ <) → R ]
  • A : θ ↦ i ↦ j ↦ (min (⟨D, E⟩))                               [ R ]
    • D : (ψ i j)                                                [ R ]
    • E : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))              [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨F, G⟩))                               [ R ]
    • F : (ψ i j)                                                [ R ]
    • G : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))              [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨H, I⟩))                               [ R ]
    • H : (ψ i j)                                                [ R ]
    • I : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))              [ R ]
```

Verified!

Transformed program

Referring to a sub-term

In[3]

```
Stratify "/" (fixee A) A ψ
```

Out[3]

```
• program(ψ ↦ (let ψ := fix(C / (ψ' / ψ')) in fix(ψ' / (A / B))))   [ (((J x J) ∩ <) → R) → ((J₀ x J₁) ∩ <) → R ]
  • A : θ ↦ i ↦ j ↦ (min (⟨D, E⟩))                                   [ R ]
    • D : (ψ i j)                                                    [ R ]
    • E : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                  [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨F, G⟩))                                   [ R ]
    • F : (ψ i j)                                                    [ R ]
    • G : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                  [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨H, I⟩))                                   [ R ]
    • H : (ψ i j)                                                    [ R ]
    • I : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                  [ R ]
```

In[4]

```
Stratify "/" (fixee B) B ψ
```

Out[4]

```
• program(ψ ↦ (let ψ := fix(C / (ψ' / ψ')) in let ψ := fix(ψ' / (ψ' / B)) in fix(ψ' / (A ψ'
  • A : θ ↦ i ↦ j ↦ (min (⟨D, E⟩))                                   [ (((J x J) ∩ <) → R) → ((J₀ x J₁) ∩ <) → R ]
    • D : (ψ i j)                                                    [ R ]
    • E : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                  [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨F, G⟩))                                   [ (((J x J) ∩ <) → R) → ((J₁ x J₁) ∩ <) → R ]
```

Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

In[4]

```
Stratify "/" (fixee [B]) [B] ψ
```

Out[4]

```
• program(ψ ↦ (let ψ := fix([C] / (ψ' / ψ')) in let ψ := fix(ψ' / (ψ' / [B])) in fix(ψ' / ([A]
  • [A] : θ ↦ i ↦ j ↦ (min (⟨[D], [E]⟩))                    [ (((J × J) ∩ <) → R) → ((J₀ × J₁) ∩ <) → R ]
    • [D] : (ψ i j)                                                                  [ R ]
    • [E] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • [B] : θ ↦ i ↦ j ↦ (min (⟨[F], [G]⟩))                    [ (((J × J) ∩ <) → R) → ((J₁ × J₁) ∩ <) → R ]
    • [F] : (ψ i j)                                                                  [ R ]
    • [G] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • [C] : θ ↦ i ↦ j ↦ (min (⟨[H], [I]⟩))                    [ (((J × J) ∩ <) → R) → ((J₀ × J₀) ∩ <) → R ]
    • [H] : (ψ i j)                                                                  [ R ]
    • [I] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```

(apply same tactic to both terms)

Invoke synthesis

In[5]

```
[B] [C] ↦ SynthAuto . ... ψ
```

Out[5]

```
• program(ψ ↦ (let ψ := ([C] / (ψ' / ψ')) ψ in let ψ := (ψ' / (ψ' / [B])) ψ in fix(ψ' / ([A]
  • [A] : θ ↦ i ↦ j ↦ (min (⟨[D], [E]⟩))                    [ (((J × J) ∩ <) → R) → ((J₀ × J₁) ∩ <) → R ]
    • [D] : (ψ i j)                                                                  [ R ]
    • [E] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • [B] : [A[J₁]] : ψ ↦ fix(θ ↦ i ↦ j ↦ (min (⟨[F], [G]⟩)))
    • [F] : (ψ i j)                                          [ R ] ((((J × J) ∩ <) → R) → ((J₁ × J₁) ∩ <) → R ]
    • [G] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • [C] : [A[J₀]] : ψ ↦ fix(θ ↦ i ↦ j ↦ (min (⟨[H], [I]⟩)))
    • [H] : (ψ i j)                                          [ R ] (((J × J) ∩ <) → R) → ((J₀ × J₀) ∩ <) → R ]
    • [I] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```

Synthesized recursive calls

In[6]

```
B[J₀,J₁] :-
 ψ ↦ fix (((? ↦ ψ) : (J×J∩<→R)→J₀×J₀→R) /
          ((θ i j ↦ min ⟨ψ i j,
                        min (k ↦ θ i k + θ k j + w i k j)⟩)
            : (J×J∩< → R) → J₀×J₁ → R) /
          ((? ↦ ψ) : (J×J∩<→R)→J₁×J₁→R))
```

Derived definition

Out[6]

```
• ψ ↦ fix((ψ' / [A]) / ψ')                                                           [ (J → J → R) → J → J →
  • [A] : θ ↦ i ↦ j ↦ (min (⟨[B], [C]⟩))                  [ (((J × J) ∩ <) → R) → J₀ → J₁ → R ]
    • [B] : (ψ i j)                                                                  [ R ]
    • [C] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```

In[7]

```
K₀ K₁ ⊆ J₀
K₂ K₃ ⊆ J₁
Slice (find (θ ↦ ?)) (`? ⟨`K₀,`K₁⟩ ⟨`K₂,`K₃⟩)
```

Out[7]

```
• program(ψ ↦ fix((ψ' / ([A] / ([B] / ([C] / [D])))) / ψ'))
  • [A] : θ ↦ i ↦ j ↦ (min (⟨[E], [F]⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₂ → R ]
    • [E] : (ψ i j)                                                                  [ R ]
    • [F] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • [B] : θ ↦ i ↦ j ↦ (min (⟨[G], [H]⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₃ → R ]
    • [G] : (ψ i j)                                                                  [ R ]
    • [H] : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```

Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

In[7]

```
Kø K₁ ⊆ Jø
K₂ K₃ ⊆ J₁
Slice (find (θ ↦ ?)) (`? ⟨`Kø,`K₁⟩ ⟨`K₂,`K₃⟩)
```

Out[7]

- program(ψ ↦ fix((ψ̈ / (A / (B / (C / D)))) / ψ̈))
  - A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩)) [(((J × J) ∩ <) → R) → Kø → K₂ → R]
    - E : (ψ i j) [R]
    - F : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - B : θ ↦ i ↦ j ↦ (min (⟨G, H⟩)) [(((J × J) ∩ <) → R) → Kø → K₃ → R]
    - G : (ψ i j) [R]
    - H : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - C : θ ↦ i ↦ j ↦ (min (⟨I, J⟩)) [(((J × J) ∩ <) → R) → K₁ → K₂ → R]
    - I : (ψ i j) [R]
    - J : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - D : θ ↦ i ↦ j ↦ (min (⟨K, L⟩)) [(((J × J) ∩ <) → R) → K₁ → K₃ → R]
    - K : (ψ i j) [R]
    - L : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]

In[8]

```
Stratify "/" (fixee C) C ψ
```

Out[8]

- program(ψ ↦ (let ψ := fix((ψ̈ / (ψ̈ / (ψ̈ / (D / ψ̈)))) / ψ̈) in fix((ψ̈ / (A / (B /
  - A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩)) [(((J × J) ∩ <) → R) → Kø → K₂ → R]
    - E : (ψ i j) [R]
    - F : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - B : θ ↦ i ↦ j ↦ (min (⟨G, H⟩)) [(((J × J) ∩ <) → R) → Kø → K₃ → R]
    - G : (ψ i j) [R]
    - H : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - C : θ ↦ i ↦ j ↦ (min (⟨I, J⟩)) [(((J × J) ∩ <) → R) → K₁ → K₃ → R]
    - I : (ψ i j) [R]
    - J : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - D : θ ↦ i ↦ j ↦ (min (⟨K, L⟩)) [(((J × J) ∩ <) → R) → K₁ → K₂ → R]
    - K : (ψ i j) [R]
    - L : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]

In[9]

```
Stratify "/" (fixee A) A ψ
```

Out[9]

- program(ψ ↦ (let ψ := fix((ψ̈ / (ψ̈ / (ψ̈ / (D / ψ̈)))) / ψ̈) in let ψ := fix((ψ̈ / (C
  - A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩)) [(((J × J) ∩ <) → R) → Kø → K₃ → R]
    - E : (ψ i j) [R]
    - F : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - B : θ ↦ i ↦ j ↦ (min (⟨G, H⟩)) [(((J × J) ∩ <) → R) → K₁ → K₃ → R]
    - G : (ψ i j) [R]
    - H : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - C : θ ↦ i ↦ j ↦ (min (⟨I, J⟩)) [(((J × J) ∩ <) → R) → Kø → K₂ → R]
    - I : (ψ i j) [R]
    - J : (min k ↦ ((θ i k) + (θ k j) + (w i k j))) [R]
  - D : θ ↦ i ↦ j ↦ (min (⟨K, L⟩)) [(((J × J) ∩ <) → R) → K₁ → K₂ → R]
    - K : (ψ i j) [R]

Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

```
    • J : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                    [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨K, L⟩))                  [ (((J × J) ∩ <) → R) → K₁ → K₂ → R ]
    • K : (ψ i j)                                                     [ R ]
    • L : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
```

In[10]

```
Stratify "/" (fixee B) B ψ
```

Out[10]

```
• program(ψ ↦ (let ψ := fix((ψ' / (ψ' / (ψ' / (D / ψ')))) / ψ') in let ψ := fix((ψ' / (C
  • A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₃ → R ]
    • E : (ψ i j)                                                     [ R ]
    • F : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨G, H⟩))                  [ (((J × J) ∩ <) → R) → K₁ → K₃ → R ]
    • G : (ψ i j)                                                     [ R ]
    • H : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨I, J⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₂ → R ]
    • I : (ψ i j)                                                     [ R ]
    • J : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨K, L⟩))                  [ (((J × J) ∩ <) → R) → K₁ → K₂ → R ]
    • K : (ψ i j)                                                     [ R ]
    • L : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
```

In[11]

```
⟨Slice (find B (k ↦ ?)) ⟨`K₁, `K₂, `K₃⟩,
 Slice (find C (k ↦ ?)) ⟨`K₀, `K₁, `K₂⟩,
 Slice (find A (k ↦ ?)) ⟨`K₀, `K₁, `K₂, `K₃⟩⟩
```

Out[11]

```
• program(ψ ↦ (let ψ := fix((ψ' / (ψ' / (ψ' / (D / ψ')))) / ψ') in let ψ := fix((ψ' / (C
  • A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₃ → R ]
    • E : (ψ i j)                                                     [ R ]
    • F : (min (G / (H / (I / J))))                                   [ R ]
      • G : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₀ → R ]
      • H : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₁ → R ]
      • I : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₂ → R ]
      • J : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₃ → R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨K, L⟩))                  [ (((J × J) ∩ <) → R) → K₁ → K₃ → R ]
    • K : (ψ i j)                                                     [ R ]
    • L : (min (M / (N / O)))                                         [ R ]
      • M : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₁ → R ]
      • N : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₂ → R ]
      • O : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₃ → R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨P, Q⟩))                  [ (((J × J) ∩ <) → R) → K₀ → K₂ → R ]
    • P : (ψ i j)                                                     [ R ]
    • Q : (min (R / (S / T)))                                         [ R ]
      • R : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₀ → R ]
      • S : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₁ → R ]
      • T : k ↦ ((θ i k) + (θ k j) + (w i k j))                       [ K₂ → R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨U, V⟩))                  [ (((J × J) ∩ <) → R) → K₁ → K₂ → R ]
    • U : (ψ i j)                                                     [ R ]
    • V : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                   [ R ]
```

Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

In[12]

```
Distrib min
```

Out[12]

```
• program(ψ ↦ (let ψ := fix((ψ̈ / (ψ̈ / (ψ̈ / (D / ψ̈)))) / ψ̈) in let ψ := fix((ψ̈ / (C                    (ψ
  • A : θ ↦ i ↦ j ↦ (min (⟨E, F⟩))                                  [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • E : (ψ i j)                                                                                    [ R ]
    • F : (min (⟨G, H, I, J⟩))                                                                       [ R ]
      • G : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • H : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • I : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • J : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨K, L⟩))                                  [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • K : (ψ i j)                                                                                    [ R ]
    • L : (min (⟨M, N, O⟩))                                                                          [ R ]
      • M : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • N : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • O : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨P, Q⟩))                                  [ (((J × J) ∩ <) → R) → K0 → K2 → R ]
    • P : (ψ i j)                                                                                    [ R ]
    • Q : (min (⟨R, S, T⟩))                                                                          [ R ]
      • R : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • S : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
      • T : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨U, V⟩))                                  [ (((J × J) ∩ <) → R) → K1 → K2 → R ]
    • U : (ψ i j)                                                                                    [ R ]
    • V : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
```

In[13]

```
Assoc min
```

Out[13]

```
• program(ψ ↦ (let ψ := fix((ψ̈ / (ψ̈ / (ψ̈ / (D / ψ̈)))) / ψ̈) in let ψ := fix((ψ̈ / (C                    (ψ
  • A : θ ↦ i ↦ j ↦ (min (⟨E, F, G, H, I⟩))                         [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • E : (ψ i j)                                                                                    [ R ]
    • F : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • G : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • H : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • I : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨J, K, L, M⟩))                            [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • J : (ψ i j)                                                                                    [ R ]
    • K : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • L : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • M : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨N, O, P, Q⟩))                            [ (((J × J) ∩ <) → R) → K0 → K2 → R ]
    • N : (ψ i j)                                                                                    [ R ]
    • O : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • P : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
    • Q : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨R, S⟩))                                  [ (((J × J) ∩ <) → R) → K1 → K2 → R ]
    • R : (ψ i j)                                                                                    [ R ]
    • S : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                                  [ R ]
```

Bellmania

file:///Users/corwin/var/workspace/Bellmania.Frontend/repl.html

In[14]

```
⟨Stratify min (fixee A) ⟨E,G⟩ ψ,
 Stratify min (fixee B) ⟨J,L⟩ ψ,
 Stratify min (fixee C) ⟨N,P⟩ ψ⟩
```

Out[14]

```
• program(ψ ↦ (let ψ := fix((ψ / (ψ / (ψ / (G / ψ)))) / ψ) in let ψ := let ψ
  • A : θ ↦ i ↦ j ↦ (min (⟨H, I, J, K⟩))                    [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • H : (ψ i j)                                                                  [ R ]
    • I : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • J : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • K : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨L, M⟩))                          [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • L : (ψ i j)                                                                  [ R ]
    • M : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨N, O, P⟩))                       [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • N : (ψ i j)                                                                  [ R ]
    • O : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • P : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨Q, R⟩))                          [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • Q : (ψ i j)                                                                  [ R ]
    • R : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • E : θ ↦ i ↦ j ↦ (min (⟨S, T, U⟩))                       [ (((J × J) ∩ <) → R) → K0 → K2 → R ]
    • S : (ψ i j)                                                                  [ R ]
    • T : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • U : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • F : θ ↦ i ↦ j ↦ (min (⟨V, W⟩))                          [ (((J × J) ∩ <) → R) → K0 → K2 → R ]
    • V : (ψ i j)                                                                  [ R ]
    • W : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • G : θ ↦ i ↦ j ↦ (min (⟨X, Y⟩))                          [ (((J × J) ∩ <) → R) → K1 → K2 → R ]
    • X : (ψ i j)                                                                  [ R ]
    • Y : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```

In[15]

```
Stratify min (fixee A) ⟨H,J⟩ ψ
```

Out[15]

```
• program(ψ ↦ (let ψ := fix((ψ / (ψ / (ψ / (H / ψ)))) / ψ) in let ψ := let ψ
  • A : θ ↦ i ↦ j ↦ (min (⟨I, J, K⟩))                       [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • I : (ψ i j)                                                                  [ R ]
    • J : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • K : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • B : θ ↦ i ↦ j ↦ (min (⟨L, M⟩))                          [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • L : (ψ i j)                                                                  [ R ]
    • M : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • C : θ ↦ i ↦ j ↦ (min (⟨N, O⟩))                          [ (((J × J) ∩ <) → R) → K0 → K3 → R ]
    • N : (ψ i j)                                                                  [ R ]
    • O : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • D : θ ↦ i ↦ j ↦ (min (⟨P, Q, R⟩))                       [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • P : (ψ i j)                                                                  [ R ]
    • Q : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
    • R : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
  • E : θ ↦ i ↦ j ↦ (min (⟨S, T⟩))                          [ (((J × J) ∩ <) → R) → K1 → K3 → R ]
    • S : (ψ i j)                                                                  [ R ]
    • T : (min k ↦ ((θ i k) + (θ k j) + (w i k j)))                                [ R ]
```