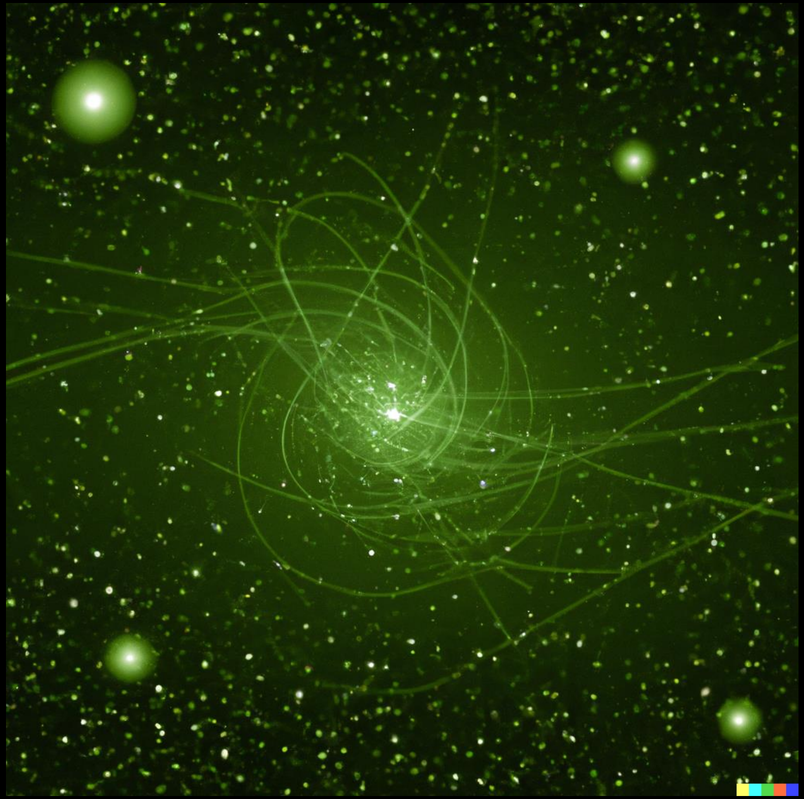
Λ-trees and the LEAN THEOREM PROVER

Raphael Appenzeller
Ventotene
September 2023



DALL-E AI result for the prompt:
`Λ-metric space`

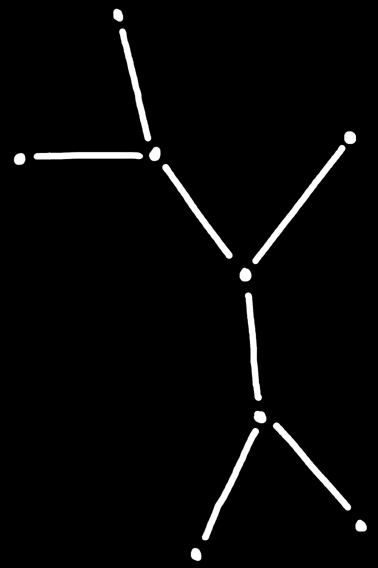
Λ-trees

Let Λ be an ordered abelian group.

Λ-trees

Let Λ be an ordered abelian group.

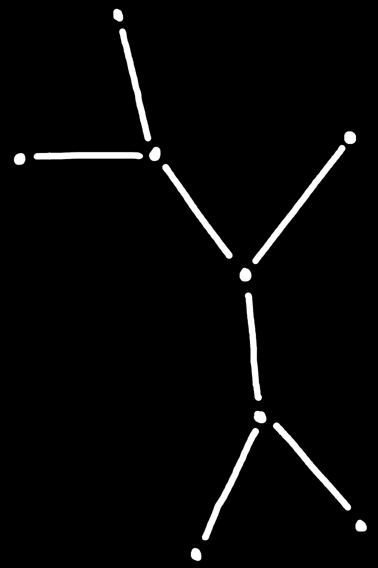
$$\Lambda = \mathbb{Z}$$



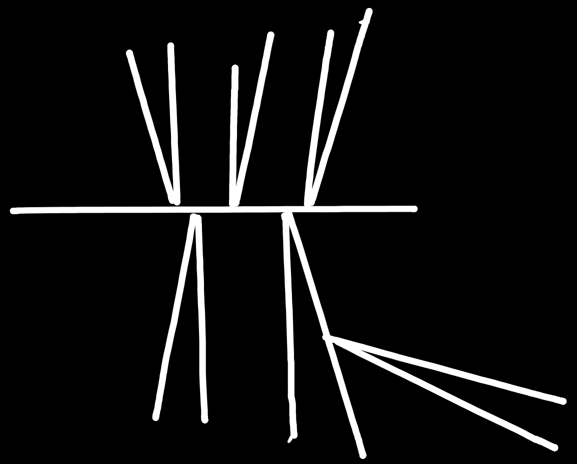
Λ-trees

Let Λ be an ordered abelian group.

$\Lambda = \mathbb{Z}$



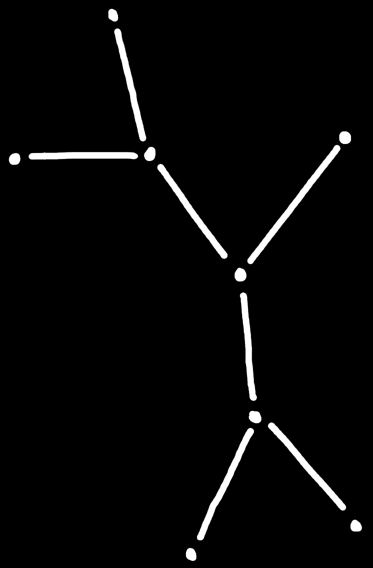
$\Lambda = \mathbb{R}$



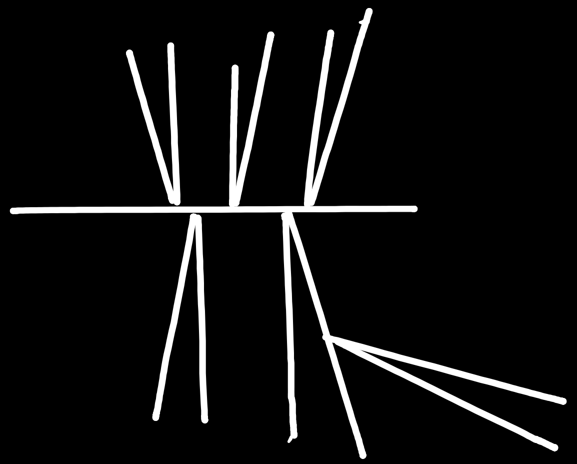
Λ-trees

Let Λ be an ordered abelian group.

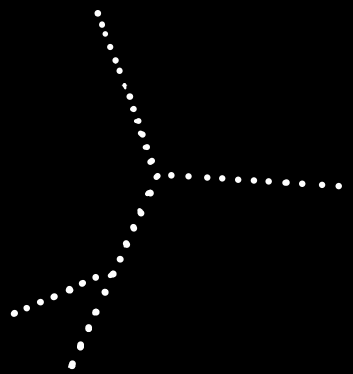
$\Lambda = \mathbb{Z}$



$\Lambda = \mathbb{R}$



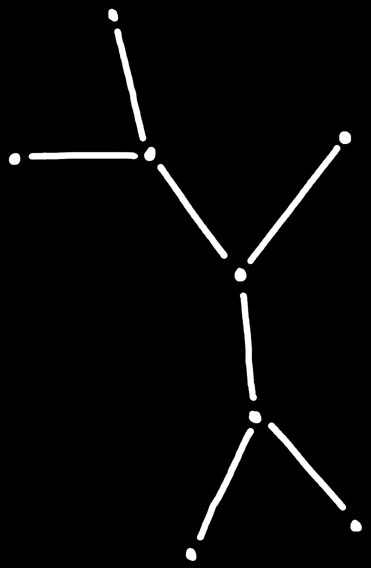
$\Lambda = \mathbb{Q}$



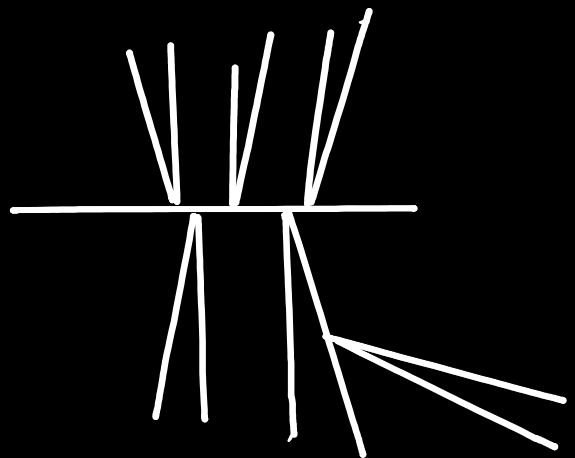
Λ-trees

Let Λ be an ordered abelian group.

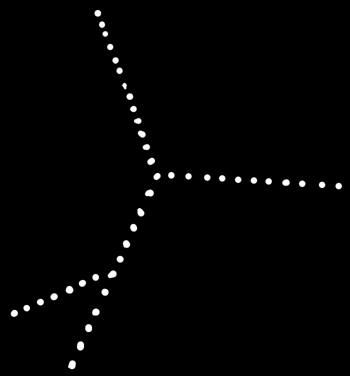
$\Lambda = \mathbb{Z}$



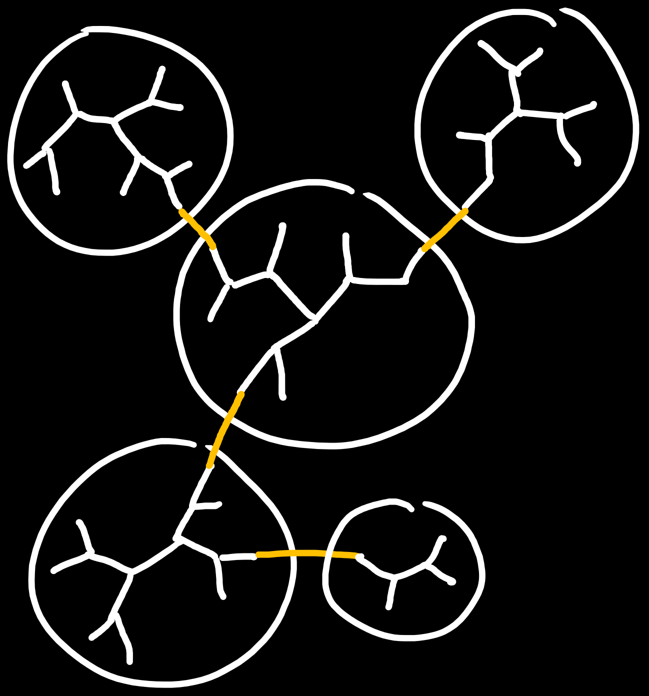
$\Lambda = \mathbb{R}$



$\Lambda = \mathbb{Q}$



$\Lambda = \mathbb{Z} \times \mathbb{Z}$



Λ-trees

Let Λ be an ordered abelian group.

Def: A Λ-metric space is a Λ-*tree* if

(1) connected

(2) no cycles

(3) no holes

Λ-trees

Let Λ be an ordered abelian group.

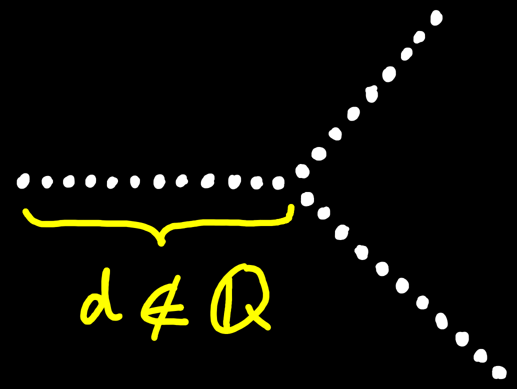
Def: A Λ-metric space is a Λ-*tree* if

(1) connected

(2) no cycles

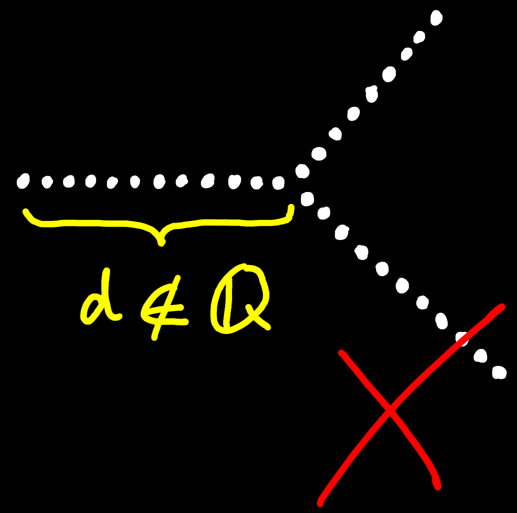
(3) no holes

Example: $\Lambda = \mathbb{Q}$



Λ-trees

Example: $\Lambda = \mathbb{Q}$



Let Λ be an ordered abelian group.

Def: A Λ-metric space is a Λ-*tree* if

(1) connected
(2) no cycles
(3) no holes

Theorem: If $\Lambda = 2\Lambda$, then $(1) \wedge (2) \implies (3)$.

LEAN THEOREM PROVER

LEAN is a formal proof verifier

LEAN is a formal proof verifier

```
class lambda_tree (Λ : Type*)[linear_ordered_add_comm_group Λ]
                  (X : Type*) extends lambda_metric_space Λ X :=
(a1 : axiom_1 Λ X)
(a2 : axiom_2 Λ X)
(a3 : axiom_3 Λ X)
```

LEAN

THEOREM PROVER

Theorem: If $\Lambda = 2\Lambda$, then $(1) \wedge (2) \implies (3)$.

```
theorem thm1 (Λ : Type*)[linear_ordered_add_comm_group Λ]
             (X : Type*)[lambda_metric_space Λ X] :
Lambda_is_two_Lambda Λ → axiom_1 Λ X ∧ axiom_2 Λ X → axiom_3 Λ X
```

THEOREM PROVER

Theorem: If $\Lambda = 2\Lambda$, then $(1) \wedge (2) \implies (3)$.

```
theorem thm1 (Λ : Type*)[linear_ordered_add_comm_group Λ]
             (X : Type*)[lambda_metric_space Λ X] :
Lambda_is_two_Lambda Λ → axiom_1 Λ X ∧ axiom_2 Λ X → axiom_3 Λ X
```

goals accomplished 🎉