Synthesis via Tactics

George Stelle
Los Alamos National Laboratory

```
Definition synthesis
   (specLang progLang : Type)
   (proof : progLang → specLang → Prop) :=
     ∀ s:specLang, {p:progLang | proof p s}.
```

```
Definition synthesis
   (specLang progLang : Type)
   (proof : progLang → specLang → Prop) :=
     ∀ s:specLang, {p:progLang | proof p s}.

Definition synthesis_via_tactics :=
   synthesis coqTerm coqTerm coqTypeJudgement
```

```
Definition gcd : ∀ x y,
   { d | (d | x) ∧ (d | y) ∧
         (∀ d’, (d’ | x) ∧ (d’ | y) → d’ ≤ d) }.
```

```
Definition synthesis :
   ∀ spec : Type, spec.

Definition decidable_synthesis :
   ∀ spec : Type, spec ∨ ¬spec.

Definition partial_synthesis :
   ∀ spec : Type, option spec.
```

"So... what does the thinking?"

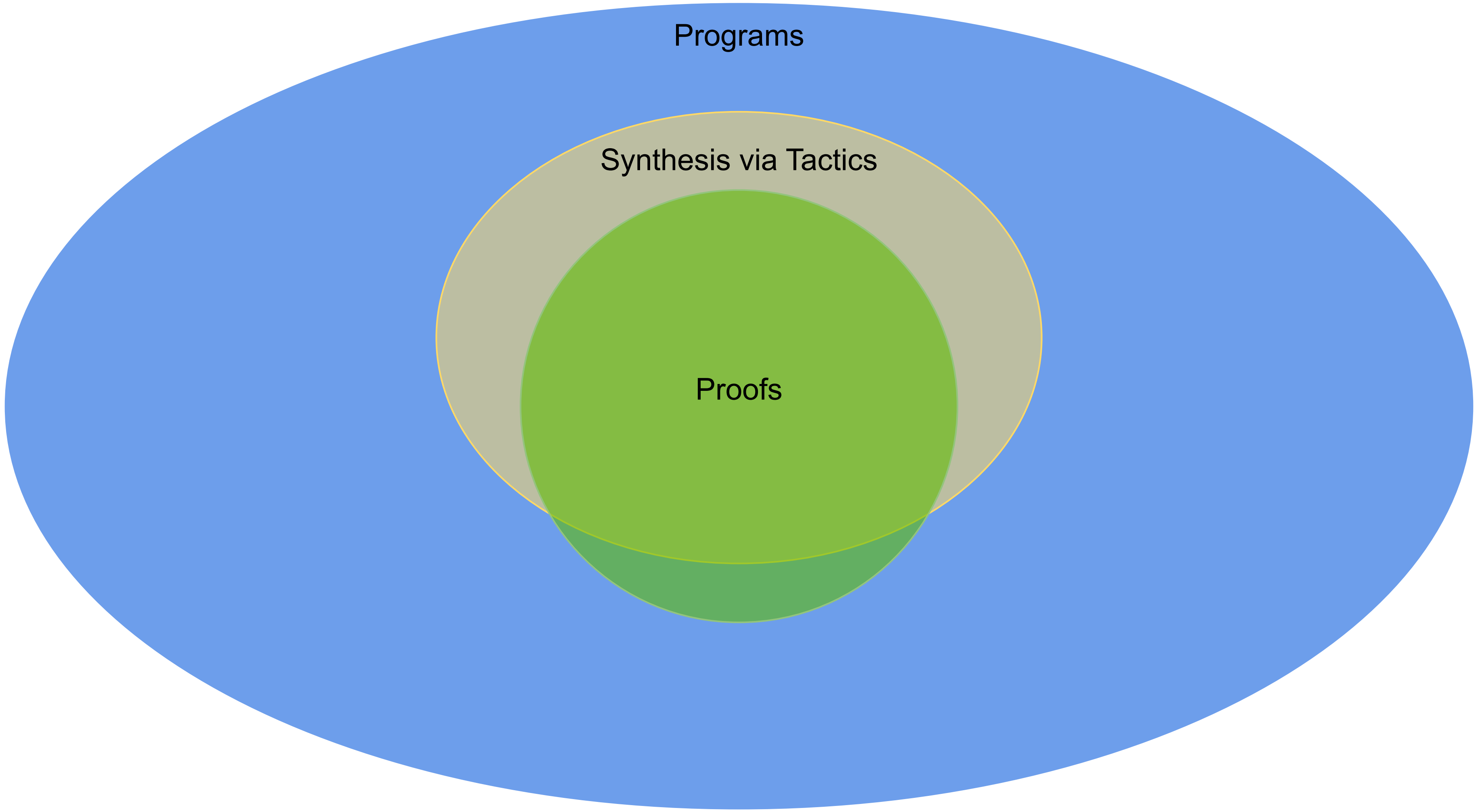
"You're not understanding, are you? The brain does the thinking. The meat."

"Thinking meat! You're asking me to believe in thinking meat!"

-Terry Bisson

```
Lemma compose_and : ∀ a b c d : Prop,
  (a → c) → (c → d) → (a ∧ b) → d.
intuition.
Qed.

Definition compose_prod : ∀ a b c d : Type,
  (a → c) → (c → d) → (a * b) → d.
intuition.
Defined.
```


Programs
Synthesis via Tactics
Proofs

```
Require Import PolikarpovaTactics.
From SMTCoq Require Import SMTCoq.

Ltac HotNewSynthesis x :=
  repeat match goal with =>
    | SeparationLogic _ _ => separationLogicSynthesis
    | SAT _ => smt
    …
  end.
```

https://coq.inria.fr/