Church-Rosser in Three Lines

A partial-function reduction relation is confluent. The TLC evaluator is a partial function. QED.

Abstract

The Church-Rosser theorem for a confluent reduction relation is usually proved via a one-step diamond, Tait-Martin-Löf parallel reduction, or a Takahashi translation. For any *deterministic* evaluator these machines are unnecessary. We give a three-line Lean proof of Church-Rosser for the ternary lambda calculus TLC: the one-step relation is a partial function, its reflexive transitive closure is therefore confluent, and Church-Rosser is an immediate corollary. The proof lives in `Proof/ChurchRosser.lean` with zero sorrys.

---

1. The three lines

Let `R $\subseteq$ $\alpha$ $\times$ $\alpha$` be a reduction relation. Call `R` *functional* (equivalently: *deterministic*, or a *partial function*) if

$$R(a,b) \wedge R(a,c) \implies b = c.$$

Call `R` *confluent* if

$$R^*(a,b) \wedge R^*(a,c) \implies \exists d.\ R^*(b,d) \wedge R^*(c,d).$$

Theorem. If `R` is functional, then `R*` is confluent.

Proof. By induction on `R^*(a, b)`. The reflexive case joins at `c`. The transitive step `a $\to$_R a' $\to$* b` inherits a second path `a $\to$_R a'' $\to$* c`. Functionality gives `a' = a''`, so the induction hypothesis on `a'` supplies the join.

That is the whole argument. In Lean, with `Relation.ReflTransGen`:

```
theorem confluent_of_partial_function
    {R : $\alpha$ $\to$ $\alpha$ $\to$ Prop} (hfun : $\forall$ a b c, R a b $\to$ R a c $\to$ 
    Confluent (Relation.ReflTransGen R) := by
  intro a b c hab hac
  induction hab with
  | refl => exact $\langle$c, hac, Relation.ReflTransGen.refl$\rangle$
  | tail _ hstep ih =>
      rcases ih with $\langle$d, hd$_1$, hd$_2$$\rangle$
      exact $\langle$d, hd$_1$, hd$_2$.tail hstep$\rangle$
```

Four lines of Lean, one line of idea.

2. TLC is a partial function

The ternary lambda calculus TLC uses a left-biased `betaStep` evaluator: at every beta-redex the leftmost outermost reduction fires. The evaluator is a total function `betaStep : Term $\to$ Option Term`. Its graph

$$\text{betaRel}(t, t') \;:\Longleftrightarrow\; \text{betaStep}(t) = \text{some } t'$$

is therefore a partial function in the relational sense. This is `betaRel_deterministic` in `ChurchRosser.lean`, proved by `rfl`-style case analysis on `betaStep`.

3. Corollary: Church-Rosser

Apply ?1 to `R = betaRel`:

```
theorem betaStar_confluent : Confluent (Relation.ReflTransGen betaRel) :=
  confluent_of_partial_function betaRel_deterministic

theorem church_rosser : ChurchRosser (Relation.ReflTransGen betaRel) :=
  betaStar_confluent
```

Normal forms are unique. Two successful bounded normaliser runs agree:

```
theorem nf_result_unique (t t$_1$ t$_2$ : Term) (fuel$_1$ fuel$_2$ : Nat) :
    nf fuel$_1$ t = some t$_1$ $\to$ nf fuel$_2$ t = some t$_2$ $\to$ t$_1$ = t$_2$
```

Again a direct corollary — functionality of `nf` follows from functionality of `betaStep`.

4. Why this is not cheating

The usual Church-Rosser proof handles non-deterministic relations: untyped lambda calculus with full beta reduction admits `($\lambda$x. x x) (($\lambda$y. y) z) $\to$ ($\lambda$x. x x) z $\to$ z z` and `($\lambda$x. x x) (($\lambda$y. y) z) $\to$ (($\lambda$y. y) z) (($\lambda$y. y) z) $\to$ z z`. There the diamond is content.

TLC fixes a left-biased evaluator. There is never a choice of redex. Confluence is trivial — but *non-trivially trivial*: the fact that the same theorem that costs 200 lines of parallel-reduction machinery in the general setting can be had for ten lines in the deterministic setting is precisely why deterministic evaluators are the right semantic primitive. You pay for non-determinism once, at the boundary where you pick an evaluation strategy, and from there on every metatheoretic fact is a one-liner.

The NilSquare residual calculus (`nilsquare_anti_comm`, `residualElt_mul_comm_zero` in the same file) is the algebraic shadow of this fact: two residuals of a common redex multiply to zero in the dual-number ring, and this is the one-line replacement for the Hindley-Rosen commutative square.

5. Three consequences

Uniqueness of bounded WHNF. `whnf_result_unique` is the same argument on the weak-head relation.

Monotone fuel. `nf_mono` and `whnf_mono`: once bounded normalisation succeeds, every larger fuel returns the same term. The proof is one call to `church_rosser`.

Replay-purity of the Zig runtime. `native/runtime/` steps TLC-compiled event streams. Because the underlying evaluator is a partial function, the stepper is replay-pure by construction — two identical event logs produce identical state. This is the Church-Rosser theorem in production.

6. Reproducibility

- One file: `Proof/ChurchRosser.lean`, 573 lines (most are the TLC term-level `betaStep` definition; the confluence proof itself is ?1 and ?3 above, ~30 lines).
- Zero sorrys.
- `lake build` verifies everything.

7. A remark on length

This paper is shorter than most Church-Rosser proofs. The theorem is shorter than the paper. Both are correct.