The Church-Rosser Theorem

Recap from Last Time

Grammar of λ-terms:

$$M, N ::= x \mid \lambda x.M \mid M\,N$$

β-reduction: The computation rule:

$$(\lambda x.M)\,N \to_\beta M[x := N]$$

Compatibility: β-reduction applies anywhere in a term:

- $M \to_\beta M'$ implies $\lambda x.M \to_\beta \lambda x.M'$
- $M \to_\beta M'$ implies $M\,N \to_\beta M'\,N$ and $N\,M \to_\beta N\,M'$

Multi-step reduction: $M \twoheadrightarrow_\beta N$ means zero or more β-steps.

Notation

$\to_\beta$ always means *exactly one* reduction step. For example, $M \to_\beta M$ is never true, but $M \twoheadrightarrow_\beta M$ is always true.

β-equivalence: $M =_\beta N$ means M and N are connected by β-steps (in either direction).

Overview

The goal of this lecture is to prove the **Church-Rosser theorem**.

Theorem (Church-Rosser)

If $M \twoheadrightarrow_\beta M'$ *and* $M \twoheadrightarrow_\beta N'$*, then there exists a* λ*-term* P *such that*

$$M' \twoheadrightarrow_\beta P \quad \text{and} \quad N' \twoheadrightarrow_\beta P$$

$$\begin{tikzcd}
 & M \arrow[dl, "\twoheadrightarrow_\beta"'] \arrow[dr, "\twoheadrightarrow_\beta"] & \\
M' \arrow[dr, dashed, "\twoheadrightarrow_\beta"'] & & N' \arrow[dl, dashed, "\twoheadrightarrow_\beta"] \\
 & P &
\end{tikzcd}$$

This property is called **confluence**: any two reduction paths from the same term can eventually be rejoined. The characteristic shape gives it the alternative name **diamond property**.

Consequences of Church-Rosser

Recall: A term is in β-**normal form** if it contains no redex, i.e., no subterm of the form $(\lambda x.M)\, N$. We view β-normal forms as the *final outcomes of computations*.

Most important consequence of Church-Rosser: for any λ-term M, either

- there exists a *unique* P in β-normal form such that $M \twoheadrightarrow_\beta P$, or
- no such P exists (and the computation runs forever no matter which reduction path is taken).

Note

Even in the first case, the computation could run forever if a wrong reduction path is chosen. Only **strongly normalizing** terms terminate regardless of the evaluation order. **Example:** Let $\Omega = (\lambda x.x\, x)(\lambda x.x\, x)$. The term $(\lambda y.z)\, \Omega$ has a normal form z (reduce the outer redex), but reducing the inner redex first gives $(\lambda y.z)\, \Omega \to_\beta (\lambda y.z)\, \Omega \to_\beta \cdots$

Recap: Single-Step Reduction

Single-step reduction: $M \to_\beta N$

Exactly one β-redex is contracted.

Example:

$$(\lambda x.x\,y)\,z \to_\beta z\,y$$

Non-example: This is *not* a single step:

$$(\lambda x.x\,y)\,((\lambda y.y\,u)\,k) \not\to_\beta k\,u\,y$$

(This requires *two* β-reductions.)

Recap: Multi-Step Reduction

When we chain several β-reductions:

$$M_1 \rightarrow_\beta M_2 \rightarrow_\beta \cdots \rightarrow_\beta M_k$$

We write:

$$M_1 \twoheadrightarrow_\beta M_k$$

$$\begin{tikzcd}
M_1 \arrow[r] \arrow[rrr, bend left, "\twoheadrightarrow_\beta"] & M_2 \arrow[r] & \cdots \arrow[r] & M_k
\end{tikzcd}$$

Example:

$$(\lambda x.x\, y)\,((\lambda y.y\, u)\, k) \twoheadrightarrow_\beta k\, u\, y$$

Proof Strategy: What We Need

Strategy for proving Church-Rosser: Find an auxiliary relation $\to_R$ satisfying:

1. $\to_\beta \subseteq \to_R \subseteq \twoheadrightarrow_\beta$
2. $\to_R$ is reflexive, i.e., $A \to_R A$ for any A
3. $\to_R$ has the diamond property, i.e., if $M \to_R M'$ and $M \to_R N'$, then there exists P such that $M' \to_R P$ and $N' \to_R P$

If such $\to_R$ exists, Church-Rosser for $\twoheadrightarrow_\beta$ follows (next slides).

What Doesn't Work

- $\to_R = \to_\beta$? No: not reflexive (contracting a redex never returns the same term), and no diamond (see below).
- $\to_R = \twoheadrightarrow_\beta$? This *would* work—but that's what we're trying to prove!

So $\to_R$ must be strictly between $\to_\beta$ and $\twoheadrightarrow_\beta$.

$$
\begin{tikzcd}
& (\lambda x.x\,x)((\lambda y.y)z) \arrow[dl, "\text{inner}"'] \arrow[dr, "\text{outer}"] & \\
(\lambda x.x\,x)z \arrow[dr, dashed] & & ((\lambda y.y)z)((\lambda y.y)z) \arrow[dl, dashed] \\
& ? &
\end{tikzcd}
$$

$L \to_\beta z\,z$ only; $\quad R \to_\beta z((\lambda y.y)z)$ or $((\lambda y.y)z)z$

No common P reachable in one step!

How These Properties Imply Church-Rosser

Since $\rightarrow_\beta \subseteq \rightarrow_R \subseteq \twoheadrightarrow_\beta$, any $M \twoheadrightarrow_\beta M'$ can be written as a sequence of $\rightarrow_R$-steps:

$$M = M_0 \rightarrow_R M_1 \rightarrow_R \cdots \rightarrow_R M_n = M'$$

Since $\rightarrow_R$ is reflexive, we can "pad" shorter paths to make them the same length.

Since $\rightarrow_R$ has the diamond property, we can *tile* any two diverging $\twoheadrightarrow_\beta$ paths with diamonds, eventually reaching a common term.

We illustrate this tiling argument on the following slides.

Diamond Tiling: Setup

Suppose $M \twoheadrightarrow_\beta M'$ in 3 steps and $M \twoheadrightarrow_\beta N'$ in 3 steps.

Since $\rightarrow_\beta \subseteq \rightarrow_R$, we can express these as $\rightarrow_R$-paths. (If one path is shorter, we can pad it with reflexive steps $P \rightarrow_R P$.)

$$\begin{array}{c} \qquad M \\ \swarrow \qquad\qquad \searrow \\ \twoheadrightarrow_\beta \left\{ \begin{array}{c} A_1 \\ \downarrow \\ A_2 \\ \downarrow \\ M' \end{array} \right. \qquad ? \qquad \left. \begin{array}{c} B_1 \\ \downarrow \\ B_2 \\ \downarrow \\ N' \end{array} \right\} \twoheadrightarrow_\beta \end{array}$$

Goal: Find P with $M' \twoheadrightarrow_\beta P$ and $N' \twoheadrightarrow_\beta P$.

Diamond Tiling: Step 1

Apply diamond to the first divergence from M:

$$
\begin{tikzcd}
 & M \arrow[dl] \arrow[dr] & \\
A_1 \arrow[d] \arrow[dr, dashed] & & B_1 \arrow[d] \arrow[dl, dashed] \\
A_2 \arrow[d] & C_{11} & B_2 \arrow[d] \\
M' & & N'
\end{tikzcd}
$$

By diamond: $\exists C_{11}$ with $A_1 \to_R C_{11}$ and $B_1 \to_R C_{11}$.

Diamond Tiling: Step 2

Apply diamond to A_1, A_2, C_{11}:

$$
\begin{tikzcd}
 & & M \arrow[dll] \arrow[drr] & & \\
A_1 \arrow[d] \arrow[drr] & & & & B_1 \arrow[dll] \arrow[d] \\
A_2 \arrow[d] \arrow[dr, dashed] & & C_{11} \arrow[dl, dashed] & & B_2 \arrow[d] \\
M' & C_{21} & & & N'
\end{tikzcd}
$$

By diamond: $\exists C_{21}$ with $A_2 \to_R C_{21}$ and $C_{11} \to_R C_{21}$.

Diamond Tiling: Step 3

Apply diamond to B_1, B_2, C_{11}:

$$
\begin{tikzcd}
 & & M \arrow[dll] \arrow[drr] & & \\
A_1 \arrow[d] \arrow[drr] & & & & B_1 \arrow[dll] \arrow[d] \\
A_2 \arrow[d] \arrow[dr] & & C_{11} \arrow[dl] \arrow[dr, dashed] & & B_2 \arrow[dl, dashed] \arrow[d] \\
M' & C_{21} & & C_{12} & N'
\end{tikzcd}
$$

By diamond: $\exists C_{12}$ with $C_{11} \to_R C_{12}$ and $B_2 \to_R C_{12}$.

Diamond Tiling: Step 4

Apply diamond to C_{21}, C_{12}:

$$
\begin{tikzcd}
 & & M \arrow[dll] \arrow[drr] & & \\
A_1 \arrow[d] \arrow[drr] & & & & B_1 \arrow[dll] \arrow[d] \\
A_2 \arrow[d] \arrow[dr] & & C_{11} \arrow[dl] \arrow[dr] & & B_2 \arrow[dl] \arrow[d] \\
M' & C_{21} \arrow[dr, dashed] & & C_{12} \arrow[dl, dashed] & N' \\
 & & C_{22} & &
\end{tikzcd}
$$

By diamond: $\exists C_{22}$ with $C_{21} \to_R C_{22}$ and $C_{12} \to_R C_{22}$.

Diamond Tiling: Steps 5–6

Continue with diamonds involving M' and N':

$$
\begin{tikzcd}
 & & M \arrow[dll] \arrow[drr] & & \\
A_1 \arrow[d] \arrow[drr] & & & & B_1 \arrow[dll] \arrow[d] \\
A_2 \arrow[d] \arrow[dr] & & C_{11} \arrow[dl] \arrow[dr] & & B_2 \arrow[dl] \arrow[d] \\
M' \arrow[ddr, dashed] & C_{21} \arrow[dr] \arrow[dd, dashed] & & C_{12} \arrow[dl] \arrow[dd, dashed] & N' \arrow[ddl, dashed] \\
 & & C_{22} & & \\
 & C_{31} & & C_{13} &
\end{tikzcd}
$$

Diamonds 5 and 6 give us C_{31} and C_{13}.

Diamond Tiling: Final Step

$$\begin{tikzcd}
 & & M \arrow[dll] \arrow[drr] & & \\
A_1 \arrow[d] \arrow[drr] & & & & B_1 \arrow[d] \arrow[dll] \\
A_2 \arrow[d] \arrow[dr] & & C_{11} \arrow[dl] \arrow[dr] & & B_2 \arrow[dl] \arrow[d] \\
M' \arrow[ddr] \arrow[dddrr, bend right, blue, "\twoheadrightarrow_\beta"'] & C_{21} \arrow[dd] \arrow[dr] & & C_{12} \arrow[dl] \arrow[dd] & N' \arrow[ddl] \arrow[dddll, bend left, blue, "\twoheadrightarrow_\beta"] \\
 & & C_{22} \arrow[dd, dashed] & & \\
 & C_{31} \arrow[dr, dashed] & & C_{13} \arrow[dl, dashed] & \\
 & & \color{red}{P} & &
\end{tikzcd}$$

We have $M' \to_R C_{31} \to_R P$ and $N' \to_R C_{13} \to_R P$.

Since $\to_R \subseteq \twoheadrightarrow_\beta$: $M' \twoheadrightarrow_\beta P$ and $N' \twoheadrightarrow_\beta P$. Church-Rosser proved!

Parallel Reduction: The Solution

We have shown that *any* relation $\to_R$ satisfying the three properties implies Church-Rosser. Now we construct one.

The relation must satisfy:

1. $\to_\beta \subseteq \to_R \subseteq \twoheadrightarrow_\beta$
2. $\to_R$ is reflexive, i.e., $A \to_R A$ for any A
3. $\to_R$ has the diamond property

Such a relation exists! It is called **parallel β-reduction**, written $\Rrightarrow_\beta$.

Intuitively, $M \Rrightarrow_\beta M'$ means M' is obtained from M by contracting *any subset* of the redexes that *already exist* in M—but *not* redexes created by other contractions in the same step.

(In particular, choosing the empty subset gives $M \Rrightarrow_\beta M$, so $\Rrightarrow_\beta$ is reflexive.)

Parallel vs. Sequential Reduction

$$
\begin{tikzcd}
 & (\lambda x.x\,y)\,((\lambda z.z\,u)\,k) \arrow[dl, "\text{inner}"'] \arrow[dr, Rightarrow, "\text{both}"] & \\
(\lambda x.x\,y)\,(k\,u) \arrow[d, "\text{outer}"'] & & k\,u\,y \\
k\,u\,y & &
\end{tikzcd}
$$

Parallel: 1 step

Sequential: 2 steps

(Here y, u, k are free variables.)

Parallel reduction is *more powerful* than single-step, but *less powerful* than multi-step (can't reduce newly created redexes).

Parallel Reduction Cannot Reduce New Redexes

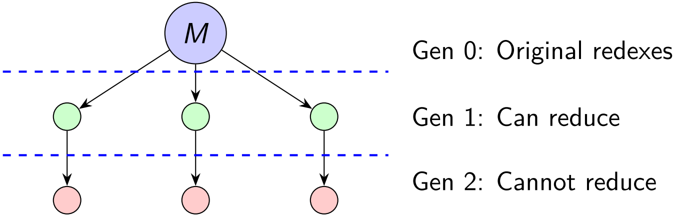
Example: Consider $M = (\lambda x.x\,x)\,(\lambda y.y)$

$$
\begin{array}{c}
(\lambda x.x\,x)\,(\lambda y.y) \\
\Big\downarrow{\scriptstyle \rightarrow_\beta} \\
\underbrace{(\lambda y.y)\,(\lambda y.y)}_{\text{new redex!}} \\
\Big\downarrow{\scriptstyle \rightarrow_\beta} \\
\lambda y.y
\end{array}
$$

- **Sequential:** Can reach $\lambda y.y$ in two steps.
- **Parallel:** From M, we can only reach $(\lambda y.y)(\lambda y.y)$ in one step.

The redex $(\lambda y.y)(\lambda y.y)$ was *created* by the first reduction—it didn't exist in M. Parallel reduction cannot contract it in the same step!

Parallel Reduction: One Generation



$\Rrightarrow_\beta$ reduces any subset of *existing* redexes simultaneously, but *cannot* reduce newly created redexes.

Parallel Reduction: Definition

Definition (Parallel reduction)

$\Rrightarrow_\beta$ is defined inductively:

1. **Variables:** $x \Rrightarrow_\beta x$
2. **Abstraction:** If $M \Rrightarrow_\beta M'$ then $\lambda x.M \Rrightarrow_\beta \lambda x.M'$
3. **Application:** If $M \Rrightarrow_\beta M'$ and $N \Rrightarrow_\beta N'$ then $M\,N \Rrightarrow_\beta M'\,N'$
4. β**-contraction:** If $M \Rrightarrow_\beta M'$ and $N \Rrightarrow_\beta N'$ then $(\lambda x.M)\,N \Rrightarrow_\beta M'[x := N']$

Rule Overlap: The Key to "Any Subset"

When $M \equiv (\lambda x.P)\,Q$ is a β-redex, **both Rules 3 and 4 apply**:

- **Rule 3** gives $(\lambda x.P)\,Q \Rrightarrow_\beta (\lambda x.P')\,Q'$ (head redex *not* contracted)
- **Rule 4** gives $(\lambda x.P)\,Q \Rrightarrow_\beta P'[x := Q']$ (head redex *contracted*)

This overlap is intentional! For each redex in M, we independently choose whether to contract it (Rule 4) or keep it (Rule 3).

Example: $(\lambda x.x)((\lambda y.y)z)$ has two redexes. We can:

- Contract neither: $(\lambda x.x)((\lambda y.y)z) \Rrightarrow_\beta (\lambda x.x)((\lambda y.y)z)$ (reflexivity)
- Contract inner only: $(\lambda x.x)((\lambda y.y)z) \Rrightarrow_\beta (\lambda x.x)z$
- Contract outer only: $(\lambda x.x)((\lambda y.y)z) \Rrightarrow_\beta (\lambda y.y)z$
- Contract both: $(\lambda x.x)((\lambda y.y)z) \Rrightarrow_\beta z$

Congruence: What Rules 1–3 Say

Rules 1–3 make $\Rrightarrow_\beta$ a **congruence**: a relation compatible with the inductive grammar of λ-calculus.

Using *only* Rules 1–3, we can *only* derive $M \Rrightarrow_\beta M$ for any term—that is, reflexivity. This is proved by induction on M:

- **Base case:** Rule 1 gives $x \Rrightarrow_\beta x$ for variables.
- **Abstraction:** If $M \Rrightarrow_\beta M$ (IH), then Rule 2 gives $\lambda x.M \Rrightarrow_\beta \lambda x.M$.
- **Application:** If $M \Rrightarrow_\beta M$ and $N \Rrightarrow_\beta N$ (IH), then Rule 3 gives $M\,N \Rrightarrow_\beta M\,N$.

Rule 4 is what allows the relation to be non-trivial: it permits $M \Rrightarrow_\beta M'$ with $M' \neq M$.

Proving the Diamond Property for $\Rrightarrow_\beta$

We have shown (via diamond tiling) that Church-Rosser follows from:

- $\to_\beta \subseteq \Rrightarrow_\beta \subseteq \twoheadrightarrow_\beta$
- $\Rrightarrow_\beta$ is reflexive, i.e., $M \Rrightarrow_\beta M$ for any M
- $\Rrightarrow_\beta$ has the diamond property

The inclusions and reflexivity follow easily from the definition. For now, we accept:

- **Substitution Lemma:** If $M \Rrightarrow_\beta M'$ and $N \Rrightarrow_\beta N'$ then $M[x := N] \Rrightarrow_\beta M'[x := N']$

We now focus on proving the diamond property for $\Rrightarrow_\beta$.

Main Idea: Complete Development

To prove the diamond property, we define the **complete development** $M^\star$ of a term M.

Theorem (Complete Development)

For each λ-term M, there exists $M^\star$ such that:

$$\textit{If } M \Rrightarrow_\beta M' \textit{ then } M' \Rrightarrow_\beta M^\star$$

$$\begin{tikzcd}
 & M \arrow[dl, Rightarrow] \arrow[dd, Rightarrow, "\text{all}"] \\
M' \arrow[dr, Rightarrow, dashed] & \\
 & M^\star
\end{tikzcd}$$

$M^\star$ is the *maximally parallel reduced* form: it reduces *all* generation-0 redexes at once.

Complete Development $\Rightarrow$ Diamond

If $M \Rrightarrow_\beta M'$ and $M \Rrightarrow_\beta N'$, take $P = M^\star$:

$$
\begin{tikzcd}
 & M \arrow[dl, Rightarrow, "\Rrightarrow_\beta"'] \arrow[dr, Rightarrow, "\Rrightarrow_\beta"] & \\
M' \arrow[dr, Rightarrow, "\Rrightarrow_\beta"'] & & N' \arrow[dl, Rightarrow, "\Rrightarrow_\beta"] \\
 & M^\star &
\end{tikzcd}
$$

Both M' and N' reduce to the *same* $M^\star$. Diamond is immediate!

Definition: Complete Development $M^\star$

Definition

The **complete development** $M^\star$ is defined inductively:

1. $x^\star = x$ (variables)
2. $(\lambda x.M)^\star = \lambda x.M^\star$ (abstractions)
3. $(M\,N)^\star = M^\star\,N^\star$ if M is not an abstraction
4. $((\lambda x.M)\,N)^\star = M^\star[x := N^\star]$ (β-redex: contract it!)

Understanding the Definition

Rules 1–3 just recurse through the structure—no reduction happens:

- $x^\star = x$ (nothing to do)
- $(\lambda x.M)^\star = \lambda x.M^\star$ (recurse into body)
- $(M\,N)^\star = M^\star\,N^\star$ (recurse into both parts)

All the reduction happens in Rule 4:

- $((\lambda x.M)\,N)^\star = M^\star[x := N^\star]$

This rule detects a β-redex and contracts it. Notice: we compute $M^\star$ and $N^\star$ *before* substituting—so the definition recurses into subterms first, then performs the substitution.

Example: Computing $M^\star$ (Part 1)

Let $M = (\lambda x.x\, y)\,((\lambda z.z)\, w)$. This term has two redexes: outer and inner.

Step 1: Spread the stars using the definition rules

$M^\star = ((\lambda x.x\, y)\,((\lambda z.z)\, w))^\star$

$= (x\, y)^\star[x := ((\lambda z.z)\, w)^\star]$ (Rule 4: outer redex)

$= (x^\star\, y^\star)[x := ((\lambda z.z)\, w)^\star]$ (Rule 3: x not abstraction)

$= (x^\star\, y^\star)[x := z^\star[z := w^\star]]$ (Rule 4: inner redex)

Now all the stars are on variables.

Example: Computing $M^\star$ (Part 2)

We have: $M^\star = (x^\star\, y^\star)[x := z^\star[z := w^\star]]$

Step 2: Evaluate the leaves (all variables, so Rule 1)

$x^\star = x, \quad y^\star = y, \quad z^\star = z, \quad w^\star = w$

Step 3: Substitute back from inside out

$= (x\, y)[x := z[z := w]]$

$= (x\, y)[x := w]$

$= w\, y$

So $M^\star = w\, y$.

Warm-up: Why $M \Rrightarrow_\beta M^\star$?

Before proving the main theorem, let's verify that $M \Rrightarrow_\beta M^\star$ for any M.

Proof: By induction on the structure of M:

- **Case $M \equiv x$:** We have $x^\star = x$ and $x \Rrightarrow_\beta x$ by Rule 1. ✓
- **Case $M \equiv \lambda x.M_1$:** By IH, $M_1 \Rrightarrow_\beta M_1^\star$. By Rule 2, $\lambda x.M_1 \Rrightarrow_\beta \lambda x.M_1^\star = (\lambda x.M_1)^\star$. ✓
- **Case $M \equiv M_1\, M_2$ where M_1 is not an abstraction:** By IH, $M_i \Rrightarrow_\beta M_i^\star$. By Rule 3, $M_1\, M_2 \Rrightarrow_\beta M_1^\star\, M_2^\star = (M_1\, M_2)^\star$. ✓
- **Case $M \equiv (\lambda x.M_1)\, M_2$ (a β-redex):** By IH, $M_i \Rrightarrow_\beta M_i^\star$. By Rule 4, $(\lambda x.M_1)\, M_2 \Rrightarrow_\beta M_1^\star[x := M_2^\star] = M^\star$. ✓

Visualizing Complete Development

$$
\begin{tikzcd}
M = & \boxed{(\lambda x.(\lambda y.y\,x)\,z)\,w} \arrow[d] \arrow[ddd, bend right, Rrightarrow, "\Rrightarrow_\beta"'] & \\
\to_\beta & (\lambda x.z\,x)\,w \arrow[d] & \text{(inner redex)} \\
\to_\beta & z\,w & \text{(outer redex)} \\
M^\star = & z\,w &
\end{tikzcd}
$$

Sequential: two $\to_\beta$ steps. Parallel: one $\Rrightarrow_\beta$ step to $M^\star$.

Main Theorem

Theorem
If $M \Rrightarrow_\beta M'$ then $M' \Rrightarrow_\beta M^\star$.

Meaning: Whatever subset of redexes we choose to reduce (getting M'), we can always "catch up" to the complete development $M^\star$ by reducing the remaining redexes.

Proof: By induction on the structure of M.

We consider what form M can take: variable, abstraction, or application.

Proof: Case 1 (Variable)

Case: $M \equiv x$.

Given that $x \Rrightarrow_\beta M'$, what does this imply about M'? Specifically, which rule could have led from x to M'?

- Rule 2 ($\lambda y.A \Rrightarrow \lambda y.B$): only for abstractions — does not apply
- Rule 3 ($A\,B \Rrightarrow A'\,B'$): only for applications — does not apply
- Rule 4 ($(\lambda y.A)\,B \Rrightarrow A'[y := B']$): only for redexes — does not apply
- Rule 1 ($x \Rrightarrow x$): applies!

So $M' \equiv x$.

This proof technique is called **rule inversion**: given a derivation, we ask which rule(s) could have produced it. It is reminiscent of what one does with sequent calculus.

By definition: $x^\star = x$.

Therefore $M' \equiv x \Rrightarrow_\beta x = M^\star$ (by Rule 1). ✓

Proof: Case 2 (Abstraction)

Case: $M \equiv \lambda x.M_1$.

By rule inversion on $\lambda x.M_1 \Rrightarrow_\beta M'$:

- Rule 1 ($y \Rrightarrow y$): only for variables — does not apply
- Rule 3 ($A\,B \Rrightarrow A'\,B'$): only for applications — does not apply
- Rule 4 ($(\lambda y.A)\,B \Rrightarrow A'[y := B']$): only for redexes — does not apply
- Rule 2 ($\lambda y.A \Rrightarrow \lambda y.B$ if $A \Rrightarrow B$): applies!

So $M' \equiv \lambda x.N_1$ for some N_1 with $M_1 \Rrightarrow_\beta N_1$.

By IH on M_1: $N_1 \Rrightarrow_\beta M_1^\star$.

Hence by Rule 2:

$$M' = \lambda x.N_1 \Rrightarrow_\beta \lambda x.M_1^\star = (\lambda x.M_1)^\star = M^\star \quad \checkmark$$

Proof: Case 3 (Application, non-redex)

Case: $M \equiv M_1\, M_2$ where M_1 is not an abstraction.

By rule inversion on $M_1\, M_2 \Rrightarrow_\beta M'$:

- Rule 1 ($x \Rrightarrow x$): only for variables — does not apply
- Rule 2 ($\lambda y.A \Rrightarrow \lambda y.B$): only for abstractions — does not apply
- Rule 4 ($(\lambda y.A)\, B \Rrightarrow A'[y := B']$): requires M_1 to be an abstraction — does not apply
- Rule 3 ($A\, B \Rrightarrow A'\, B'$ if $A \Rrightarrow A'$, $B \Rrightarrow B'$): applies!

So $M' \equiv N_1\, N_2$ for some N_1, N_2 with $M_1 \Rrightarrow_\beta N_1$ and $M_2 \Rrightarrow_\beta N_2$.

By IH: $N_1 \Rrightarrow_\beta M_1^\star$ and $N_2 \Rrightarrow_\beta M_2^\star$.

Hence by Rule 3 (since M_1 not an abstraction implies $M_1^\star$ not an abstraction):

$$M' = N_1\, N_2 \Rrightarrow_\beta M_1^\star\, M_2^\star = (M_1\, M_2)^\star = M^\star \quad \checkmark$$

Proof: Case 4 (β-redex)

Case: $M \equiv (\lambda x.M_1)\, M_2$ (a β-redex).

By rule inversion on $(\lambda x.M_1)\, M_2 \Rrightarrow_\beta M'$:

- Rule 1 ($y \Rrightarrow y$): only for variables — does not apply
- Rule 2 ($\lambda y.A \Rrightarrow \lambda y.B$): only for abstractions — does not apply
- Rule 3 ($A\, B \Rrightarrow A'\, B'$): applies! Gives $M' \equiv (\lambda x.N_1)\, N_2$ with $M_1 \Rrightarrow_\beta N_1$, $M_2 \Rrightarrow_\beta N_2$
- Rule 4 ($(\lambda y.A)\, B \Rrightarrow A'[y := B']$): applies! Gives $M' \equiv N_1[x := N_2]$ with $M_1 \Rrightarrow_\beta N_1$, $M_2 \Rrightarrow_\beta N_2$

So we have two subcases depending on which rule was used.

Proof: Case 4, Subcase 4.1

Recall: $M \equiv (\lambda x.M_1)\, M_2$.

Subcase 4.1: $M' \equiv (\lambda x.N_1)\, N_2$ with $M_1 \Rrightarrow_\beta N_1$ and $M_2 \Rrightarrow_\beta N_2$ (Rule 3 used; head redex not contracted).

By IH: $N_1 \Rrightarrow_\beta M_1^\star$ and $N_2 \Rrightarrow_\beta M_2^\star$.

Hence by Rule 4:

$$M' = (\lambda x.N_1)\, N_2 \Rrightarrow_\beta M_1^\star[x := M_2^\star] = ((\lambda x.M_1)\, M_2)^\star = M^\star \quad \checkmark$$

Proof: Case 4, Subcase 4.2

Recall: $M \equiv (\lambda x.M_1)\, M_2$.

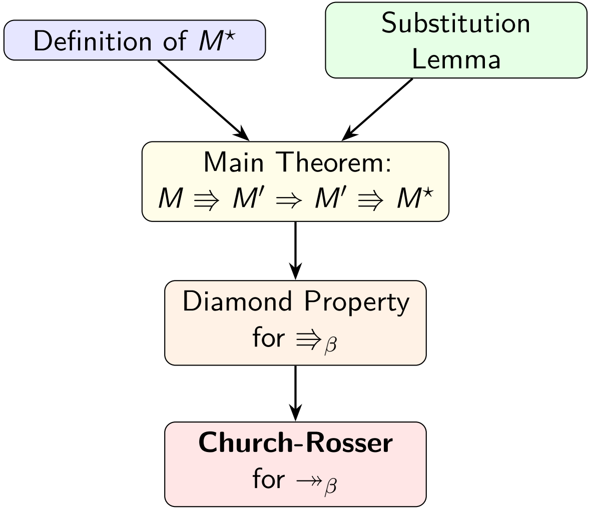
Subcase 4.2: $M' \equiv N_1[x := N_2]$ with $M_1 \Rrightarrow_\beta N_1$ and $M_2 \Rrightarrow_\beta N_2$ (Rule 4 used; head redex contracted).

By IH: $N_1 \Rrightarrow_\beta M_1^\star$ and $N_2 \Rrightarrow_\beta M_2^\star$.

Hence by the Substitution Lemma:

$$M' = N_1[x := N_2] \Rrightarrow_\beta M_1^\star[x := M_2^\star] = ((\lambda x.M_1)\, M_2)^\star = M^\star \quad \checkmark \quad \square$$

Church-Rosser: Complete Proof Summary



This is the **Takahashi proof** (1995), based on the Tait–Martin-L"of method (1970s).

Corollary: Uniqueness of Normal Forms

Corollary

If $M \twoheadrightarrow_\beta N_1$ and $M \twoheadrightarrow_\beta N_2$ where N_1 and N_2 are both in β-normal form, then $N_1 = N_2$.

Proof: By Church-Rosser, there exists P with $N_1 \twoheadrightarrow_\beta P$ and $N_2 \twoheadrightarrow_\beta P$.

But N_1 and N_2 are normal forms, so $N_1 = P = N_2$. $\square$

Corollary: Consistency

Corollary

Not all λ-terms are β-equivalent. In particular, $x \neq_\beta y$ for distinct variables x, y.

Proof: Both x and y are in β-normal form (no redexes).

If $x =_\beta y$, then by Church-Rosser there exists P with $x \twoheadrightarrow_\beta P$ and $y \twoheadrightarrow_\beta P$.

Since x and y are normal: $x = P = y$. Contradiction. $\square$

This ensures the λ-calculus is a *sensible* system—not everything is equal!

Parallel Reduction and Concurrency

Beyond proving Church-Rosser, parallel reduction has a natural interpretation:

Concurrent computation: $\Rrightarrow_\beta$ identifies which redexes can be reduced *simultaneously*—they are the redexes that exist in the original term.

- Redexes in generation 0 are *independent*: contracting one does not affect the others
- They can in principle be computed in parallel
- Newly created redexes (generation $1+$) must wait for the previous reductions to complete

This gives a formal basis for reasoning about which parts of a computation are independent.

Summary

1. **Parallel reduction** $\Rrightarrow_\beta$ reduces multiple redexes simultaneously (but only existing ones)
2. Key properties: $\rightarrow_\beta \subseteq \Rrightarrow_\beta \subseteq \twoheadrightarrow_\beta$ and the diamond property
3. **Complete development** $M^\star$ is the maximally parallel reduced form
4. **Main theorem:** Any partial development can be completed to the full development $M^\star$
5. Diamond $+$ inclusions $\Rightarrow$ **Church-Rosser** for $\twoheadrightarrow_\beta$
6. **Consequences:** Unique normal forms, consistency of λ-calculus

Appendix: Proof of Property (1)

Lemma

If $M \rightarrow_\beta N$ then $M \Rrightarrow_\beta N$.

Proof idea: A single β-reduction is a special case of parallel reduction where we reduce exactly one redex and leave everything else unchanged.

The congruence rules (1–3) let us "do nothing" to parts of the term (since $P \Rrightarrow_\beta P$ for any P).

Appendix: Proof of Property (2)

Lemma
If $M \Rrightarrow_\beta N$ then $M \twoheadrightarrow_\beta N$.

Proof: By induction on the structure of M:

- **Case $M \equiv x$:** Only Rule 1 applies, so $N \equiv x$. Then $x \twoheadrightarrow_\beta x$. ✓
- **Case $M \equiv \lambda x.M_1$:** Only Rule 2 applies, so $N \equiv \lambda x.N_1$ with $M_1 \Rrightarrow_\beta N_1$. By IH, $M_1 \twoheadrightarrow_\beta N_1$. By compatibility, $\lambda x.M_1 \twoheadrightarrow_\beta \lambda x.N_1$. ✓
- **Case $M \equiv M_1\, M_2$ where M_1 is not an abstraction:** Only Rule 3 applies, so $N \equiv N_1\, N_2$ with $M_i \Rrightarrow_\beta N_i$. By IH, $M_i \twoheadrightarrow_\beta N_i$. By compatibility, $M_1\, M_2 \twoheadrightarrow_\beta N_1\, N_2$. ✓
- **Case $M \equiv (\lambda x.M_1)\, M_2$:** Both Rules 3 and 4 apply. Either $N \equiv (\lambda x.N_1)\, N_2$ or $N \equiv N_1[x := N_2]$, with $M_i \Rrightarrow_\beta N_i$. By IH, $M_i \twoheadrightarrow_\beta N_i$. In both subcases, $M \twoheadrightarrow_\beta N$. ✓

Appendix: Substitution Lemma

Lemma (Substitution Lemma)

If $M \Rrightarrow_\beta M'$ and $N \Rrightarrow_\beta N'$ then

$$M[x := N] \Rrightarrow_\beta M'[x := N']$$

Proof: By induction on the structure of M. We unpack M into four cases.

Appendix: Substitution Lemma (Case 1 – Variable)

Case $M \equiv y$:

By rule inversion on $y \Rrightarrow_\beta M'$: only Rule 1 applies.

So $M' \equiv y$.

Subcase $y = x$:

$$M[x := N] = N \quad \text{and} \quad M'[x := N'] = N'$$

Since $N \Rrightarrow_\beta N'$ by assumption, we are done. ✓

Subcase $y \neq x$:

$$M[x := N] = y \quad \text{and} \quad M'[x := N'] = y$$

By Rule 1, $y \Rrightarrow_\beta y$. ✓

Appendix: Substitution Lemma (Case 2 – Abstraction)

Case $M \equiv \lambda y.M_1$:

By rule inversion on $\lambda y.M_1 \Rrightarrow_\beta M'$: only Rule 2 applies.

So $M' \equiv \lambda y.M_1'$ with $M_1 \Rrightarrow_\beta M_1'$.

(Assume $y \neq x$ and $y \notin FV(N) \cup FV(N')$, renaming if necessary.)

$$M[x := N] = \lambda y.M_1[x := N]$$

$$M'[x := N'] = \lambda y.M_1'[x := N']$$

By IH on M_1: $M_1[x := N] \Rrightarrow_\beta M_1'[x := N']$.

By Rule 2: $\lambda y.M_1[x := N] \Rrightarrow_\beta \lambda y.M_1'[x := N']$. ✓

Appendix: Substitution Lemma (Case 3 – Non-redex Application)

Case $M \equiv M_1\, M_2$ where M_1 is not an abstraction:

By rule inversion: only Rule 3 applies.

So $M' \equiv M_1'\, M_2'$ with $M_i \Rrightarrow_\beta M_i'$.

$$M[x := N] = M_1[x := N]\, M_2[x := N]$$
$$M'[x := N'] = M_1'[x := N']\, M_2'[x := N']$$

By IH on M_1: $M_1[x := N] \Rrightarrow_\beta M_1'[x := N']$.

By IH on M_2: $M_2[x := N] \Rrightarrow_\beta M_2'[x := N']$.

By Rule 3: $M_1[x := N]\, M_2[x := N] \Rrightarrow_\beta M_1'[x := N']\, M_2'[x := N']$. ✓

Appendix: Substitution Lemma (Case 4 – Redex, Subcase a)

Case $M \equiv (\lambda y.P)\, Q$:

By rule inversion: both Rules 3 and 4 apply, giving two subcases.

Subcase (a): $M' \equiv (\lambda y.P')\, Q'$ with $P \Rrightarrow_\beta P'$ and $Q \Rrightarrow_\beta Q'$ (Rule 3; head redex not contracted).

(Assume $y \neq x$ and $y \notin FV(N) \cup FV(N')$.)

$$M[x := N] = (\lambda y.P[x := N])\, Q[x := N]$$
$$M'[x := N'] = (\lambda y.P'[x := N'])\, Q'[x := N']$$

By IH: $P[x := N] \Rrightarrow_\beta P'[x := N']$ and $Q[x := N] \Rrightarrow_\beta Q'[x := N']$.

By Rule 2: $\lambda y.P[x := N] \Rrightarrow_\beta \lambda y.P'[x := N']$.

By Rule 3: $(\lambda y.P[x := N])\, Q[x := N] \Rrightarrow_\beta (\lambda y.P'[x := N'])\, Q'[x := N']$. ✓

Appendix: Substitution Lemma (Case 4 – Redex, Subcase b)

Subcase (b): $M' \equiv P'[y := Q']$ with $P \Rrightarrow_\beta P'$ and $Q \Rrightarrow_\beta Q'$ (Rule 4; head redex contracted).

(Assume $y \neq x$ and $y \notin FV(N) \cup FV(N')$.)

$$M[x := N] = (\lambda y.P[x := N])\, Q[x := N]$$
$$M'[x := N'] = P'[y := Q'][x := N']$$

By IH: $P[x := N] \Rrightarrow_\beta P'[x := N']$ and $Q[x := N] \Rrightarrow_\beta Q'[x := N']$.

By Rule 4:

$$(\lambda y.P[x := N])\, Q[x := N] \Rrightarrow_\beta P'[x := N'][y := Q'[x := N']]$$

By the **substitution commutation lemma** (since $y \neq x$, $y \notin FV(N')$):

$$P'[x := N'][y := Q'[x := N']] = P'[y := Q'][x := N'] = M'[x := N']$$

Hence $M[x := N] \Rrightarrow_\beta M'[x := N']$. $\square$

Thank You

Questions?