In this supplementary lecture we prove that the λ-calculus is confluent. This is result is due to Alonzo Church (1903–1995) and J. Barkley Rosser (1907–1989) and is known as the *Church–Rosser theorem.* The proof given here takes an alternative approach to the standard proof due to Tait and Martin-Löf as given for example in Barendregt [1] (see also [15, 16]).

1 Terms as Labeled Trees

In the proof of confluence, we will need to talk about *occurrences* of subterms in a term as designated by a path from the root. Thus we need to develop notation for terms viewed as labeled trees. We will refer to this view of terms as the *coalgebraic* view, although the rationale for this terminology will only become clear much later in the course.

1.1 Algebraic View of Terms

A *ranked alphabet* is a set Σ together with an *arity function* $\mathsf{arity} : \Sigma \to \mathbb{N}$. The letters $f \in \Sigma$ are viewed as operator symbols, and $\mathsf{arity}(f) \in \mathbb{N}$ denotes the *arity* of f, or the number of inputs of f. The symbol f is called *unary*, *binary*, *ternary*, or *n-ary* according as its arity is $1, 2, 3$, or n, respectively. It is called a *constant* if its arity is 0.

Finite terms over Σ can be defined by induction:

- If $t_0, \ldots, t_{n-1}$ are terms and $f \in \Sigma$ with $\mathsf{arity}(f) = n$, then $f\, t_0\ \ldots,\ t_{n-1}$ is a term.

Note the base case $n = 0$ is included; the first premise is vacuous in that case. This defines terms in prefix notation, but really we are interested in the abstract syntax.

1.2 Coalgebraic View of Terms

Alternatively, we can define terms as labeled trees. Terms represented in this way are called *coterms.* Let $\mathbb{N}^*$ denote the set of finite-length strings of natural numbers. A subset $t \subseteq \mathbb{N}^*$ is a *tree* if it is nonempty and prefix-closed (that is, if $\alpha\beta \in t$, then $\alpha \in t$). Any tree must contain the empty string, which is the *root* of the tree.

A *coterm* is then a partial function $e : \mathbb{N}^* \rightharpoonup \Sigma$ such that

- $\mathsf{dom}\, e$ is a tree;
- if $\alpha \in \mathsf{dom}\, e$, then $\alpha\, i \in \mathsf{dom}\, e$ iff $i < \mathsf{arity}(e(\alpha))$.

The second condition says that a node α of the tree has exactly n children if the arity of its label is n.

A coterm is *finite* if its domain is a finite set. Thus one advantage of this definition is that it admits infinite terms.[1]

[1]Infinite terms exist in OCaml. Type the following at the interpreter and see what happens: `type x = C of x;; let rec x = C x;;`

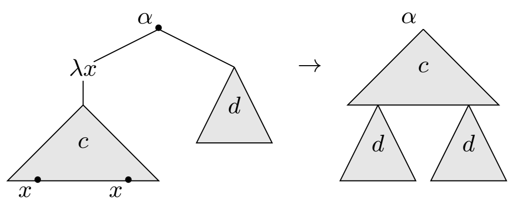

Figure 1: A β-reduction at α

If $\alpha \in \mathsf{dom}\, e$, the *subterm of e rooted at α* is the coterm $e \upharpoonright \alpha = \lambda\beta.\, e(\alpha\beta)$.[2] The domain of $e \upharpoonright \alpha$ is $\{\beta \mid \alpha\beta \in \mathsf{dom}\, e\}$. Note that if α is a prefix of β, then $e \upharpoonright \beta$ is a subterm of $e \upharpoonright \alpha$.

1.3 Substitution

Another advantage of the coalgebraic view is that we have a more general notion of substitution. We can speak of substituting a term for a single occurrence of a subterm of e as specified by its location in the tree.

Formally, if $\alpha \in \mathsf{dom}\, e$ and d is a term, define

$$e[d/\alpha] \;\triangleq\; \lambda\beta\,.\begin{cases} d(\gamma), & \text{if } \beta = \alpha\gamma, \\ e(\beta), & \text{if } \alpha \text{ is not a prefix of } \beta. \end{cases}$$

This is not safe substitution, but raw substitution; we merely replace the subterm of e at α with d.

2 λ-Coterms

A λ-coterm is just a labeled tree as described in the preceding section over a ranked alphabet consisting of variables x of arity 0, unary binding operators λx, and the binary application operator. Ordinary λ-terms are just finite λ-coterms.

Now the β-reduction rule takes the following form. Suppose α is a redex in e, say $e \upharpoonright \alpha = (\lambda x.\, c)\, d$. The corresponding contractum obtained by applying the β-reduction rule is $c\{d/x\}$, which we substitute for the redex at α. The new term is $e[c\{d/x\}/\alpha]$. This is illustrated in Fig. 1.

2.1 Acceptable Orderings

If α and β are both redexes in e and α is a proper prefix of β, then $e \upharpoonright \beta$ is a proper subterm of $e \upharpoonright \alpha$. If we reduce α before reducing β, then β will in general no longer be a redex; indeed, it may no longer even exist in the resulting tree. However, if we reduce β first, then α is still a redex in the resulting tree (although the subterm rooted at α may have changed), and we can reduce it.

More generally, if $A \subseteq \mathsf{dom}\, e$ is a set of redexes in e, and we reduce the redexes in A in some order consistent with the subterm relation—that is, we reduce $\alpha \in A$ only if all proper extensions $\alpha\beta \in A$ have already been reduced—then every redex in A will still be available when it is time to reduce it, and it will be possible to reduce all of them. Moreover, the actual order does not matter, as long as it is consistent with the subterm relation.

[2]Here we are using λ as a meta-operator. Thus $\lambda\beta.\, e(\alpha\beta)$ is the function that on input $\beta \in \mathbb{N}^*$ returns the value of e applied to $\alpha\beta$.

Formally, we say that a linear ordering $\alpha_1, \ldots, \alpha_n$ of the elements of A is *acceptable* if $\alpha_i = \alpha_j\beta$ implies $i \leq j$. In other words, the sequence $\alpha_1, \ldots, \alpha_n$ is a subsequence of some total extension of the partial order $\{(\alpha\beta, \alpha) \mid \alpha,\ \beta \in \mathbb{N}^*\}$ (or, if you like, a topological sort of A with respect to the edges $(\alpha\beta, \alpha)$).

Acceptable orderings of a given set of redexes are not unique. However, it does not matter which one we pick. One can easily prove inductively that for all acceptable orderings of A, the length of the reduction sequence will be the same, namely the cardinality of A, and the resulting terms will be identical. Let us call this term $\theta_A(e)$, as it depends only on the starting term e and the set of redexes A, not on the order of the reductions.

For $\alpha \in \mathbb{N}^*$, let $\alpha{\downarrow} = \{\alpha\beta \mid \beta \in \mathbb{N}^*\}$. If $\alpha \in \mathsf{dom}\, e$, then $\alpha{\downarrow} \cap \mathsf{dom}\, e$ represents the set of subterms of $e \restriction \alpha$.

For $A, X \in \mathbb{N}^*$, write $A \rhd X$ if there exists an α such that $A \subseteq \alpha{\downarrow}$ and $\alpha{\downarrow} \cap X = \varnothing$. If $A \rhd X$, then A and X are disjoint, and there exists an acceptable ordering of $A \cup X$ such that all elements of A come before all elements of X.

3 Confluence

We start by proving confluence in some special cases, building up to the general result.

Lemma 1. *Let A and B be two sets of redexes of e such that all elements of A are prefix-incomparable to all elements of B. Then $\theta_B(\theta_A(e))$ and $\theta_A(\theta_B(e))$ both exist and are equal. This gives the confluent diagram illustrated in Fig. 2.*

$$\begin{tikzcd}
& e \arrow[dl, "B"'] \arrow[dr, "A"] & \\
\theta_B(e) \arrow[dr, "A"] & & \theta_A(e) \arrow[dl, "B"'] \\
& \begin{array}{c}\theta_B(\theta_A(e)) \\ = \theta_A(\theta_B(e))\end{array} &
\end{tikzcd}$$

Figure 2

Proof. Both $\theta_B(\theta_A(e))$ and $\theta_A(\theta_B(e))$ represent the reduction of the redexes in $A \cup B$ in different acceptable orders, thus both terms are equal to $\theta_{A\cup B}(e)$. ☐

Lemma 2. *Let α be a redex of e, and let A be a set of redexes of e such that $A \subseteq \alpha{\downarrow}$. Then there exists a set B of redexes of $\theta_\alpha(e)$ such that $B \subseteq \alpha{\downarrow}$ and*

$$\theta_C(\theta_A(e)) \;=\; \theta_B(\theta_\alpha(e)),$$

where $C = \{\alpha\}$ if $\alpha \notin A$ and $C = \varnothing$ if $\alpha \in A$. This gives the confluent diagram illustrated in Fig. 3.

$$\begin{tikzcd}
& e \arrow[dl, "A"'] \arrow[dr, "\alpha"] & \\
\theta_A(e) \arrow[dr, "C"] & & \theta_\alpha(e) \arrow[dl, "B"'] \\
& \begin{array}{c}\theta_C(\theta_A(e)) \\ = \theta_B(\theta_\alpha(e))\end{array} &
\end{tikzcd}$$

Figure 3

Proof. Suppose first that $\alpha \notin A$. Let $e \restriction \alpha = (\lambda x.\, c)\, d$. The set A may contain redexes in c and d. Reducing α first, a copy of d replaces each free occurrence of x in c (see Fig. 1). If we then reduce the redexes in these copies of d in some acceptable order, then reduce the remaining redexes in c in some acceptable order, this yields the same result as reducing the redexes in d and c in some acceptable order before reducing α, then reducing α.

Formally, take $B = \{\alpha\gamma_i \mid 1 \leq i \leq m\} \cup \{\alpha\delta_i\beta_j \mid 1 \leq i \leq k,\ 1 \leq j \leq n\}$, where $A = \{\alpha 00\gamma_i \mid 1 \leq i \leq m\} \cup \{\alpha 1\beta_j \mid 1 \leq j \leq n\}$ and the free occurrences of x in c are located at $\{\alpha 00\delta_1, \ldots, \alpha 00\delta_k\}$. The elements of A of the form $\alpha 00\gamma_i$ represent the redexes in c, which after reducing α become the elements of B of the form $\alpha\gamma_i$. The elements of A of the form $\alpha 1\beta_j$ represent the redexes in d, which after reducing α become the elements of B of the form $\alpha\delta_i\beta_j$ representing the corresponding redexes in the copies of d that replaced the free occurrences of x in c. Fig. 1 illustrates the case $k = 2$.

If $\alpha \in A$, then it must appear last in any acceptable ordering of A. By the previous argument, there exists $B \subseteq \alpha{\downarrow}$ such that $\theta_\alpha(\theta_{A-\{\alpha\}}(e)) = \theta_B(\theta_\alpha(e))$, therefore $\theta_\varnothing(\theta_A(e)) = \theta_\alpha(\theta_{A-\{\alpha\}}(e)) = \theta_B(\theta_\alpha(e))$. □

Lemma 3. *Let A and X be sets of redexes of e such that $A \rhd X$. There exists a set B of redexes of $\theta_X(e)$ such that*

$$\theta_X(\theta_A(e)) \;=\; \theta_B(\theta_X(e)).$$

Proof. This follows easily by induction on the cardinality of X using Lemmas 1 and 2. Starting with $X_0 = X$ and $B_0 = A$, construct a sequence of sets X_i and B_i by taking the elements of X one at a time in some acceptable order, maintaining the invariant $B_i \rhd X_i$. Fig. 4 illustrates the case $X = \{\beta_1, \beta_2, \beta_3\}$. □

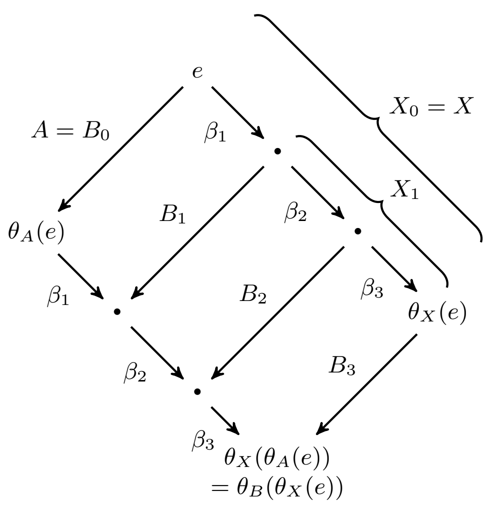

Figure 4

Lemma 4. *Let A be an arbitrary set of redexes of e, and let α be a redex of e. Then there exist redex sets C of $\theta_A(e)$ and B of $\theta_\alpha(e)$ such that*

$$\theta_C(\theta_A(e)) \;=\; \theta_B(\theta_\alpha(e)).$$

This gives the confluent diagram illustrated in Fig. 3 (the same diagram as for Lemma 2, but with a different interpretation of the symbols.)

Proof. Partition A into $A_1 = \alpha{\downarrow} \cap A$ and $A_2 = A - A_1$. Then $A_1 \rhd A_2$. By Lemma 2, there exist a set $B_1 \subseteq \alpha{\downarrow}$ of redexes of $\theta_\alpha(e)$ and $C_1 \subseteq \{\alpha\}$ such that

$$\theta_{C_1}(\theta_{A_1}(e)) \;=\; \theta_{B_1}(\theta_\alpha(e)). \tag{1}$$

Take $B = B_1 \cup A_2$. Since $B_1 \subseteq \alpha{\downarrow}$, $C_1 \subseteq \alpha{\downarrow}$, and $\alpha{\downarrow} \cap A_2 = \varnothing$, we have $B_1 \rhd A_2$ and $C_1 \rhd A_2$. By Lemma 3, there exists a set C of redexes of $\theta_{A_2}(\theta_{A_1}(e)) = \theta_A(e)$ such that

$$\theta_C(\theta_{A_2}(\theta_{A_1}(e))) \;=\; \theta_{A_2}(\theta_{C_1}(\theta_{A_1}(e))). \tag{2}$$

Then

$$\begin{aligned}
\theta_C(\theta_A(e)) &= \theta_C(\theta_{A_2}(\theta_{A_1}(e))) && \text{since } A_1 \rhd A_2\\
&= \theta_{A_2}(\theta_{C_1}(\theta_{A_1}(e))) && \text{by (2)}\\
&= \theta_{A_2}(\theta_{B_1}(\theta_\alpha(e))) && \text{by (1)}\\
&= \theta_B(\theta_\alpha(e)) && \text{since } B_1 \rhd A_2.
\end{aligned}$$

□

Lemma 5. *Let $e \to e'$ by some arbitrary sequence of reductions, and let A be a set of redexes of e. Then there exists a set B of redexes of e' such that $\theta_A(e) \to \theta_B(e')$.*

Proof. This follows in a straightforward fashion by induction on the length of the reduction $e \to e'$ by composing the reductions of Lemma 4. □

Theorem 6 (Church–Rosser Theorem). *Let $e \to e_1$ and $e \to e_2$ by some arbitrary sequences of reductions. Then there exists an e_3 such that $e_1 \to e_3$ and $e_2 \to e_3$.*

Proof. Decompose the reduction sequence $e \to e_1$ into maximal segments such that the ordering of redexes is acceptable in each segment. Lemma 5 gives a confluent diagram for each segment, and these can be composed to get a confluent diagram for the entire reduction sequence $e \to e_1$. □

References

[1] H. P. Barendregt. *The Lambda Calculus, Its Syntax and Semantics.* North-Holland, 2nd edition, 1984.

[2] Henk P. Barendregt and Jan Willem Klop. Applications of infinitary lambda calculus. *Inf. and Comput.*, 207(5):559–582, 2009.

[3] James Gosling, Bill Joy, Jr. Guy L. Steele, and Gilad Bracha. *The Java Language Specification.* Prentice Hall, 3rd edition, 2005.

[4] John E. Hopcroft and Richard M. Karp. A linear algorithm for testing equivalence of finite automata. Technical Report 71-114, University of California, 1971.

[5] Jean-Baptiste Jeannin. Capsules and closures. *Electron. Notes Theor. Comput. Sci.*, 276:191–213, September 2011.

[6] Jean-Baptiste Jeannin and Dexter Kozen. Capsules and separation. In Nachum Dershowitz, editor, *Proc. 27th ACM/IEEE Symp. Logic in Computer Science (LICS'12)*, pages 425–430, Dubrovnik, Croatia, June 2012. IEEE.

[7] Jean-Baptiste Jeannin and Dexter Kozen. Computing with capsules. In Martin Kutrib, Nelma Moreira, and Rogério Reis, editors, *Proc. Conf. Descriptional Complexity of Formal Systems (DCFS 2012)*, volume 7386 of *Lecture Notes in Computer Science*, pages 1–19, Braga, Portugal, July 2012. Springer.

[8] J. W. Klop and R. C. de Vrijer. Infinitary normalization. In S. Artemov, H. Barringer, A. S. d'Avila Garcez, L. C. Lamb, and J. Woods, editors, *We Will Show Them: Essays in Honour of Dov Gabbay*, volume 2, pages 169–192. College Publications, 2005.

[9] Peter J. Landin. The mechanical evaluation of expressions. *Computer Journal*, 6(4):308–320, 1964.

[10] Saunders MacLane. *Categories for the Working Mathematician.* Springer-Verlag, New York, 1971.

[11] John McCarthy. History of LISP. In Richard L. Wexelblat, editor, *History of programming languages I*, pages 173–185. ACM, 1981.

[12] Robin Milner and Mads Tofte. Co-induction in relational semantics. *Theoretical Computer Science*, 87(1):209–220, 1991.

[13] Eugenio Moggi. Notions of computation and monads. *Information and Computation*, 93(1), 1991.

[14] Benjamin C. Pierce. *Types and Programming Languages.* MIT Press, 2002.

[15] Robert Pollack. Polishing up the Tait–Martin-Löf proof of the Church–Rosser theorem. In *Proc. De Wintermöte '95.* Department of Computing Science, Chalmers University, Göteborg, Sweden, January 1995.

[16] Masako Takahashi. Parallel reductions in λ-calculus (revised version). *Information and Computation*, 118(1):120–127, April 1995.

[17] Robert Endre Tarjan. Efficiency of a good but not linear set union algorithm. *J. ACM*, 22(2):215–225, 1975.

[18] Philip Wadler. Monads for functional programming. In M. Broy, editor, *Marktoberdorf Summer School on Program Design Calculi*, volume 118 of *NATO ASI Series F: Computer and systems sciences.* Springer Verlag, August 1992.

[19] Glynn Winskel. *The Formal Semantics of Programming Languages.* MIT Press, 1993.