# Lecture Notes

August 27, 2023

# Contents

# 1 Lecture 3

## 1.1 Properties of abstract reduction systems

In this lecture, we explore some properties of abstract reduction systems and look at the relationships between them.

**Definition 1.1.** An abstract reduction system $(A, \underset{A}{\longrightarrow})$ is said to be **Church-Rosser** if

$$\forall x, y \in A, \quad x \overset{*}{\underset{A}{\longleftrightarrow}} y \implies x \downarrow_A y$$

Consider abstract reduction systems $A$ and $B$ as shown in the figure below. System $A$ is not Church-Rosser, since $a \overset{*}{\underset{A}{\longleftrightarrow}} b$ but $a$ and $b$ are not joinable. System $B$ is a simple example of a Church-Rosser system.

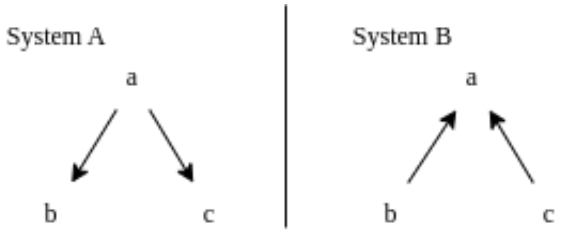


Figure 1: Examples of abstract reduction systems

**Definition 1.2.** An abstract reduction system $(A, \underset{A}{\longrightarrow})$ is said to be **Confluent** if

$$\forall a, b, c \in A, \quad a \overset{*}{\underset{A}{\longrightarrow}} b \text{ and } a \overset{*}{\underset{A}{\longrightarrow}} c \implies b \downarrow_A c$$

**Definition 1.3.** An abstract reduction system $(A, \underset{A}{\longrightarrow})$ is said to be **Semi-confluent** if

$$\forall a, b, c \in A, \quad a \underset{A}{\longrightarrow} b \text{ and } a \overset{*}{\underset{A}{\longrightarrow}} c \implies b \downarrow_A c$$

**Theorem 1.4.** For an abstract reduction system, **Church-Rosser, Confluence and Semi-confluence are equivalent**.

**Proof:** We will prove this theorem in the following three stages. Clearly, the three of them combined result in the theorem stated above.

1. Church-Rosser $\implies$ Confluence
2. Confluence $\implies$ Semi-confluence
3. Semi-confluence $\implies$ Church-Rosser

First, we will prove that **if a system is Church-Rosser, it is confluent**. Let $(A, \underset{A}{\longrightarrow})$ be an abstract reduction system that is Church-Rosser.

Let $a, b, c \in A : a \underset{A}{\overset{*}{\rightarrow}} b$ and $a \underset{A}{\overset{*}{\rightarrow}} c$

By definition, $b \underset{A}{\overset{*}{\leftrightarrow}} c$

Since A is Church-Rosser, $b \underset{A}{\overset{*}{\leftrightarrow}} c \implies b \downarrow_A c$

i.e. $a \underset{A}{\overset{*}{\rightarrow}} b$ and $a \underset{A}{\overset{*}{\rightarrow}} c \implies b \downarrow_A c$, proving that A is confluent

Next, we will prove that **if a system is Confluent, it is also semi-confluent**. Let $(A, \underset{A}{\rightarrow})$ be a confluent abstract reduction system.

Let $a, b, c \in A : a \underset{A}{\rightarrow} b$ and $a \underset{A}{\overset{*}{\rightarrow}} c$

Since $A \subseteq A^*$, $a \underset{A}{\rightarrow} b \implies a \underset{A}{\overset{*}{\rightarrow}} b$

Since A is confluent, $a \underset{A}{\overset{*}{\rightarrow}} b$ and $a \underset{A}{\overset{*}{\rightarrow}} c \implies b \downarrow_A c$

i.e. $a \underset{A}{\rightarrow} b$ and $a \underset{A}{\overset{*}{\rightarrow}} c \implies b \downarrow_A c$, proving that A is semi-confluent

Finally, we will prove that **if a system is semi-confluent, it is also Church-Rosser**. Let $(A, \underset{A}{\rightarrow})$ be a semi-confluent abstract reduction system.

Let $a, b \in A : a \underset{A}{\overset{*}{\leftrightarrow}} b$

Let $p$ be the shortest path connecting $a$ and $b$ in $A^{\leftrightarrow^*}$. We will use induction on $|p|$ to prove that $a$ and $b$ are joinable.

**Base case:** For $p = 0$, we have $a = b$ which makes them trivially joinable.

**Induction step:** Let it be true that if the shortest path connecting $a$ and $b$ in $A^{\leftrightarrow^*}$ is $|p|$, then $a$ and $b$ are joinable in $A$. We will prove that this is also true for $|p| + 1$.

Let $a, b' \in A$ such that the shortest path connecting them in $A^{\leftrightarrow^*}$ is of length $|p| + 1$. Then, $\exists b \in A$ : the shortest path connecting $a$ and $b$ in $A^{\leftrightarrow^*}$ is $|p|$ and $b \underset{A}{\leftrightarrow} b'$. Since our induction hypothesis holds true for $|p|$, $a$ and $b$ are joinable (they both reduce to some $c \in A$).

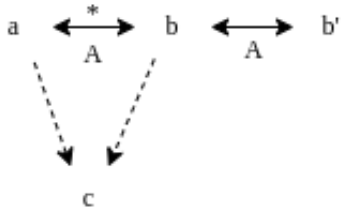


Figure 2: Abstract reduction system representing the induction step

We now have 2 cases.

**Case 1:** $b \underset{A}{\leftarrow} b'$

$b' \underset{A}{\rightarrow} b \underset{A}{\overset{*}{\rightarrow}} c$. Therefore, $a \downarrow_A b'$.

**Case 2:** $b \underset{A}{\rightarrow} b'$

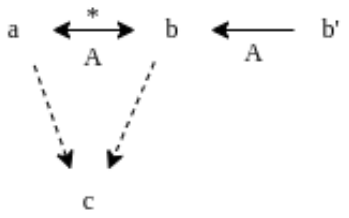


Figure 3: Abstract reduction system in case 1

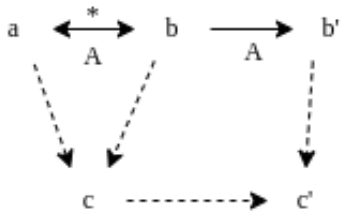


Figure 4: Abstract reduction system in case 2

Here we have $b \underset{A}{\longrightarrow} b'$ and $b \underset{A}{\overset{*}{\leftrightarrow}} c$. Since $A$ is semi-conflient, $b'$ and $c$ must be joinable. That is, $\exists c' \in A : b' \underset{A}{\overset{*}{\rightarrow}} c'$ and $c \underset{A}{\overset{*}{\rightarrow}} c'$. Since $a \underset{A}{\overset{*}{\rightarrow}} c \underset{A}{\overset{*}{\rightarrow}} c'$, we have $a \underset{A}{\overset{*}{\rightarrow}} c'$. Hence, $a \downarrow_A b'$

In either case, we have shown that $a \downarrow_A b'$, which completes our induction. We have proven that $a \underset{A}{\overset{*}{\leftrightarrow}} b \implies a \downarrow_A b$, which means A is also Church-Rosser.

This completes our proof that Church-Rosser, Confluence and Semi-confluence are equivalent properties of an abstract reduction system. While it may seem redundant to have multiple terms to refer to the same thing, they each give us a different perspective of looking at the same property which can prove to be helpful.

## 1.2 Address space

Data structures provide a way to organize and address data. For a data structure, a valid set of addresses form its address space. This is **not** a formal definition of address spaces and is only meant to give a broad idea. We will look at an example below to illustrate one way of addressing a binary tree. Let us consider the following addressing of a binary tree: each edge is labelled 1 or 2 depending on whether it leads to the left or right descendant of a node; the address of each node is obrained by appending the label of the edge leading into it to the address of its parent, with the root being $\epsilon$. Look at the figure below to bettter understand this.

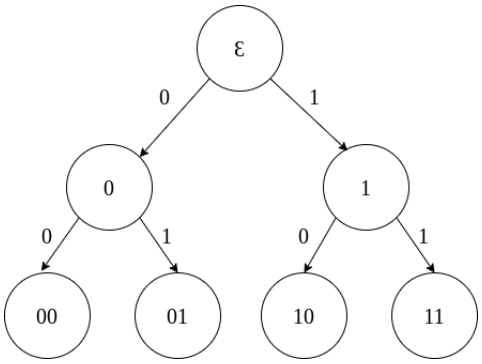


Figure 5: Addressing a binary tree

The address space for this binary tree would be the set $S = \{\epsilon, 0, 00, 01, 1, 10, 11\}$. The set $S_1 = \{\epsilon, 0, 00, 01, 1\}$ would also be a valid address space for some binary tree but the set $S_2 = \{0, 00, 1\}$ would not, since it does not contain $\epsilon$, the address of the root.

# 2 Lecture 5

## 2.1 Terms

**ARS** Abstract Reduction System

**Terminating ARS** An ARS is terminating *iff.* it has no infinite runs.

## 2.2 Well Founded Induction(WFI)

This is property of an abstract reduction system. A system having this property implies that,

$$\forall x \in A.\, \Big(\forall y \in A.\, x \xrightarrow{+} y \implies P(y)\Big) \implies P(x)$$

in other words, $P(x)$ is satisfied if $\forall y. x \xrightarrow{+} y \implies P(y)$ is satisfied.

**Theorem 2.1.** Let, $(A, \longrightarrow)$ be an ARS, then $A$ satisfies WFI iff. $(A, \longrightarrow)$ is terminating.

*Proof.*
**1. If $\longrightarrow$ terminates, then $A$ satisfies WFI.**
*Proof by contraposition.* Assume that WFI does not hold.
That implies that $\neg P(a_0)$ for some $a_0 \in A$. Since we assumed that WFI does not hold, $\exists a_1$, such that, $a_0 \xrightarrow{+} a_1$ and $\neg P(a_1)$. Using the same arguement, $\exists a_2$, such that, $a_1 \xrightarrow{+} a_2$ and $\neg P(a_2)$. Hence there is an infinite chain

$a_0 \xrightarrow{+} a_1 \xrightarrow{+} a_2 \xrightarrow{+} ...$, i.e. $\longrightarrow$ does not terminate.

**2. If $A$ satisfies WFI, then $\longrightarrow$ terminates.**
*Proof by WFI.* Let,

$$P(x) := \text{there is no infinite chain starting from x.}$$

Clearly, if there is no infinite chain starting from any successor of $x$, then there is no infinite chain starting from $x$. Hence, WFI holds and we can conclude that $P(x)$ holds for all $x$, i.e., $\longrightarrow$ terminates. □

## 2.3 Confluence

Order of evaluation does not matter.

**Joinable** $x$ and $y$ are joinable, denoted by $\downarrow$ *iff.* they have the same normal form.

**Local Confluence** An element $x \in A$ is said to be locally confluent if $\forall y, z \in A, z \xleftarrow{+} x \xrightarrow{+} y \; \exists w : y \xrightarrow{*} w \xleftarrow{*} z$, in other words, $y \downarrow z$.

$\rightarrow$ If a system is terminating and has local confluence for all vertices, then the system has **confluence**.

## 2.4 Interative Evaluation

**TODO**