# Chapter 5

# Inductively defined sets; Structural induction

This chapter looks at a generalisation of the inductive definitions of the last section. An example of an inductively defined set is the following.

Suppose you want to define *by finite means*, and define *precisely*, the set of all *"simple" arithmetical expressions* that use the numbers $1, 2, 3$, the operations $+$ and $\times$, and round brackets. Then you would do it like this:

The set of said *simple arithmetical expressions* is the *smallest* set ($\subseteq$-smallest) that

1. Contains each of $1, 2$ and $3$.
2. If it contains expressions $E$ and $E'$, then it also contains $(E + E')$ and $(E \times E')$.

Some folks would add a 3rd requirement "nothing else is in the set unless so demonstrated using 1. 2. above" and omit "smallest". *Really*?

How exactly would you so "demonstrate"? In a recursive definition you ought to be able to make your recursive calls and not have to trace back why the object you constructed exists!

We will prove in Section 5.2.5 that indeed there *is* an iterative way to show that a *particular* simple arithmetic expression was formed correctly by our recursion, but that defeats the beauty of recursion. Besides, until we reach said section we don't *know* what "nothing else is in the set unless so demonstrated using 1. 2. above" *means* or *how* to "use" 1. and 2. do it! So it is nonsense to stick such a statement in the bottom of the definition as a (redundant) afterthought.

Before we get to the general definitions, let us finesse our construction and propose some terminology.

(a) First off, in step 1. above we say that $1, 2$ and $3$ are *the initial objects* of our recursive/inductive definition.

(b) In step 2. we say that $(E+E')$ is obtained by an *operation* (on strings) that is available to us, depicted as a "blackbox" below, which we named "+".

$$\begin{array}{c} E \\ \longrightarrow \\ \longrightarrow \\ E' \end{array} \boxed{\quad + \quad} \longrightarrow (E+E')$$

In words, the operation *concatenates from left to right the strings*

"(", "$E$", "+", "$E'$", and ")"

Similar comments for the operation "$\times$".

(c) Both operations in this example are single-valued, that is, functions. It is preferable to be slightly more general and allow operations that are just relations, but not necessarily functions. Such an operation $O(x_1, \ldots, x_n, y)$ is $n$-ary —$n$ inputs, $x_1, \ldots, x_n$— with output variable $y$.

(d) We say that a set of objects $S$ is *closed under* a **relation** (operation) —it could be a function— $O(x_1, \ldots, x_n, y)$ meaning that for *all* input values $x_1, \ldots, x_n$ in $S$, *all* the obtained values $y$ are also in $S$.

We are ready for the general definition:

**5.0.1 Definition.** Given a set of *initial objects* $\mathcal{I}$ and a *set* of *operations* $\mathcal{O} = \{O_1, O_2, O_2, \ldots\}$, the object $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is called the *closure of* $\mathcal{I}$ *under* $\mathcal{O}$ —or the set *inductively defined by the pair* $(\mathcal{I}, \mathcal{O})$— and *denotes* the $\subseteq$-smallest *class*[†] $S$ that satisfies

1. $\mathcal{I} \subseteq S$.
2. $S$ is *closed under all operations in* $\mathcal{O}$, or simply, closed under $\mathcal{O}$.
3. The "smallest" part: Any class $T$ that satisfies 1. and 2. also satisfies $S \subseteq T$.

The set $\mathcal{O}$ may be infinite. Each operation $O_i$ is a set. $\square$

Nice definition, but does $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *exist* given $\mathcal{I}$ and $\mathcal{O}$? Yes. But first,

**5.0.2 Theorem.** *For any choice of* $\mathcal{I}$ *and* $\mathcal{O}$*, if* $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *exists, then it is unique.*

*Proof.* Say $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O}) = T$. Then, letting $S$ pose as closure, we get $S \subseteq T$ from 5.0.1. Then, letting $T$ pose as closure, we get $T \subseteq S$, again from 5.0.1. Thus $S = T$. $\square$

[†] Let's say "class" until we learn that it is actually a *set*.

**5.0.3 Theorem.** *For any choice of $\mathcal{I}$ and $\mathcal{O}$ with the restrictions of Definition 5.0.1 the set* $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *exists.*

*Proof.* We have to check and note a few things.

1. By 3.1.5, for each $O_i$, $\mathrm{ran}(O_i)$ is a set.
2. The class $F = \{\mathrm{ran}(O_i) : i = 1, 2, 3 \ldots\}$ is a set. This is so by **Principle** 2, since I can index all members of $F$ by assigning unique indices from $\mathbb{N}$ to each of its members (and $\mathbb{N}$ is a set by **Principle** 0).
3. By 2. above and 2.4.16, $\bigcup F$ is a set, and so is $T = \mathcal{I} \cup \bigcup F$ □
4. $T$ contains $\mathcal{I}$ as a subset (by the way $T$ was defined) and is $\mathcal{O}$-closed since any $O_i$-output —no matter where the inputs come from— is in $\mathrm{ran}(O_i) \subseteq \bigcup F$.
5. The family $\mathbb{G} = \{S : \mathcal{I} \subseteq S \;\&\; S \text{ is } \mathcal{O}\text{-closed}\}$ contains the set $T$ as a member. Thus (cf. 2.4.17)

$$C \stackrel{Def}{=} \left(\bigcap \mathbb{G}\right) \subseteq T$$

   is a set. Since all sets $S$ in $\mathbb{G}$ contain $\mathcal{I}$ and are $\mathcal{O}$-closed, so is $C$. But $C \subseteq S$ for all such sets $S$ the way it is defined. So it is $\subseteq$-smallest.

Thus, $C = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. We proved existence. □

## *5.1. Induction over a closure*

**5.1.1 Definition.** Let a pair $(\mathcal{I}, \mathcal{O})$ be given as above.

We say that a property $P[x]$ <u>propagates with $\mathcal{O}$</u> *iff* for each $O_i(x_1, \ldots, x_n, y) \in \mathcal{O}$, <u>if whenever all the inputs in the $x_i$</u> satisfy $P[x]$ (i.e., $P[x_i]$ is true for each argument $x_i$), then all output values returned by $y$ —for said inputs— satisfy $P[x]$ as well. Recall that for each assignment of values to the inputs $x_1, \ldots, x_n$ we may have more than one output values in $y$; for all such values $P[y]$ is true. □

**5.1.2 Lemma.** *For all $(\mathcal{I}, \mathcal{O})$ and a property $P[x]$, if the latter propagates with $\mathcal{O}$, then the class $\mathbb{A} = \{x : P[x]\}$ is closed under $\mathcal{O}$ (is $\mathcal{O}$-closed).*

*Proof.* So let $O_i(x_1, \ldots, x_n, y) \in \mathcal{O}$. Let $a_1, \ldots, a_n$ be all in $\mathbb{A}$. Thus

$$P[a_i], \text{ for all } i = 1, \ldots, n$$

By assumption, if $O_i(a_1, \ldots, a_n, b)$, then $P[b]$ is true, hence $b \in \mathbb{A}$. □

**5.1.3 Theorem.** *Let* $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ *and a property* $P[x]$ *be given. Suppose we have done the following steps:*

1. *We showed that for each $a \in \mathcal{I}$, $P[a]$ is true.*

2. *We showed that $P[x]$ propagates $\mathcal{O}$.*

*Then $\underline{\textit{every } a \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})}$ has property $P[x]$.*

Naturally, the technique encapsulated by 1. and 2. of 5.1.3 is called "induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$" or "structural induction" over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

Note that for each $O_i \in \mathcal{O}$ the "propagation of property $P[x]$" will take the form of an I.H. followed by an I.S.:

- **Assume** for the unspecified fixed inputs $a_1, \ldots, a_n$ of $O_i$ that *all* satisfy $P[x]$. This is the *I.H.* for $O_i$.

- Then **prove** that any output $b$ of $O_i$ caused by said input also satisfies the property.

*Proof.* (of 5.1.3) Let us write

$$\mathbb{A} \stackrel{Def}{=} \{x : P[x]\}$$

Thus, 1. in 5.1.3 translates to

$$\mathcal{I} \subseteq \mathbb{A} \tag{$*$}$$

2. and 5.1.3 yield

$$\mathbb{A} \text{ is } \mathcal{O}\text{-closed} \tag{$**$}$$

If $\mathbb{A}$ were a set —a hypothesis we *cannot* make because of Russell's paradox— then $(*)$ and $(**)$ would immediately yield $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq \mathbb{A}$ and we would be done. So we have a tiny bit more work to do:

By 5.0.3, item 4, the set $T$ built for our $\mathcal{I}$ and $\mathcal{O}$ contains $\mathcal{I}$ and is $\mathcal{O}$-closed. Thus so is $T \cap \mathbb{A}$! Moreover the latter is a set, as we know (2.4.2). Hence, by 5.0.1,

$$\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq T \cap \mathbb{A} \subseteq \mathbb{A}$$

The last implication immediately translates to

$$\underline{\text{“}x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O}) \text{ implies } P[x] \text{ is true”}}$$ □

**5.1.4 Example.** Let $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where $\mathcal{I} = \{0\}$ and $\mathcal{O}$ contains just one operation, $x + 1 = y$, where $y$ is the output variable. That is,

$$n \longrightarrow \boxed{x + 1 = y} \longrightarrow n + 1 \tag{1}$$

is our only operation. By induction over $S$, I can show $S \subseteq \mathbb{N}$.

The "$P[x]$" is "$x \in \mathbb{N}$".

So $P[0]$ is true. I verified the property for $\mathcal{I}$. That the property propagates with our operation is captured by (1) above (if $n \in \mathbb{N}$, then $n + 1 \in \mathbb{N}$). Done!

Can we show also $\mathbb{N} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$? **Yes**: In this direction I do SI over $\mathbb{N}$ on variable $n$. The property, let's call it $Q[x]$, now is "$x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$".

For $n = 0$, $n \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ since $0 \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 5.0.1.

Now, say (*I.H.*) $n \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Since $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ is closed under the operation $x + 1 = y$, we have $n + 1 \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 5.0.1.

So,

$$\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = \mathbb{N}$$ □

Thus the induction over a closure generalises *SI*.

## 5.2. Closure vs. definition by stages

We will see in this section that there is also a *by-stages* or *by-steps* way to obtain $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

**5.2.1 Definition. (Derivations)** An $(\mathcal{I}, \mathcal{O})$-*derivation* —or just *derivation* if we know which $(\mathcal{I}, \mathcal{O})$ we are talking about— is a *finite sequence of objects*

$$d_1, d_2, d_3, \ldots, d_i, \ldots, d_n \tag{1}$$

satisfying:

Each $d_i$ is

1. A member of $\mathcal{I}$,

   or

2. For some $j$, one of the results of $O_j(x_1, \ldots, x_k, y)$ with inputs $a_1, \ldots, a_k$ that are found in the derivation (1) *to the left of* $d_i$.

$n$ is called the *length of the derivation*. Every $d_i$ is called an $(\mathcal{I}, \mathcal{O})$-*derived* object, or just *derived*, if the $(\mathcal{I}, \mathcal{O})$ is understood. □

Clearly, the concept of a derivation abstracts, thus generalises, the concept of *proof*, while a derived object abstracts the concept of a *theorem*.

**5.2.2 Example.** For the $(\mathcal{I}, \mathcal{O})$ of 5.1.4, here are some derivations:

$$0$$

$$0, 0, 0$$

$$0, 1, 0, 1, 0, 1, 1, 1, 1, 0$$

Nothing says we cannot repeat a $d_i$ in a derivation! Lastly here is an "efficient" derivation with no redundant steps: $0, 1, 2, 3, 4, 5$. □

**5.2.3 Proposition.** *If $d_1, d_2, d_3, \ldots, d_i, \ldots, d_n, d_{n+1}, \ldots, d_m$ is a $(\mathcal{I}, \mathcal{O})$-derivation, then so is $d_1, d_2, d_3, \ldots, d_i, \ldots, d_n$.*

*Proof.* Each $d_i$ is validated in a derivation either outright (i.e., is in $\mathcal{I}$) or by looking to the left! What we may remove to the *right* of $d_i$ does not affect the validity of that entry. $\square$

**5.2.4 Proposition.** *If* $d_1, d_2, \ldots, d_n$ *and* $e_1, e_2, \ldots, e_m$ *are* $(\mathcal{I}, \mathcal{O})$*-derivations, then so is* $d_1, d_2, \ldots, d_n, e_1, e_2, \ldots, e_m$.

*Proof.* Traversing $d_1, d_2, \ldots, d_n$ and $e_1, e_2, \ldots, e_m$ in

$$d_1, d_2, \ldots, d_n, e_1, e_2, \ldots, e_m$$

from left to right we validate each $d_i$ and each $e_j$ giving precisely the same validation *reason* as we would in each sequence $d_1, d_2, \ldots, d_n$ and $e_1, e_2, \ldots, e_m$ separately. These reasons are local to each sequence. $\square$

We now prove that defining a set $S$ as a $(\mathcal{I}, \mathcal{O})$-closure is equivalent with defining $S$ as the set of all $(\mathcal{I}, \mathcal{O})$-derived objects.

**5.2.5 Theorem.** *For any initial sets of objects and operations on objects ($\mathcal{I}$ and $\mathcal{O}$) we have that* $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$.

*Proof.* Let us write $D = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$ and prove that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = D$. We have two directions:

1. $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq D$: By induction over $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$. The property to prove is "$x \in D$".

   - Let $x \in \mathcal{I}$. Then $x$ is derived via the one-member derivation

     $$x$$

     So $x \in D$. Thus all $x \in \mathcal{I}$ have the property.

   - The property "$x \in D$" propagates with each $O_k(\vec{x}_n, y) \in \mathcal{O}$: So let each of the $x_i$ have a derivation $\boxed{\ldots, x_i}$. We show that so does $y$.

     Concatenating all these derivations we get a derivation (5.2.4)

     $$\boxed{\ldots, x_1}, \ldots, \boxed{\ldots, x_i}, \ldots, \boxed{\ldots, x_n} \tag{1}$$

     But then so is

     $$\boxed{\ldots, x_1}, \ldots, \boxed{\ldots, x_i}, \ldots, \boxed{\ldots, x_n}, y \tag{2}$$

     by 5.2.1, case 2. That is, $y$ is *derived*, hence $y \in D$ is proved (I.S.).

2. $D \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$: Let $x \in D$. This time we do good old-fashioned CVI over $\mathbb{N}$ on the length $n$ of a derivation of $x$, toward showing that $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ —this is the "property of $x$" that we prove.

   *Basis.* $n = 1$. The only way to have a 1-element derivation is that $x \in \mathcal{I}$. Thus, $x \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ by 5.0.1.

*I.H.* Assume the claim for $x$ derived with length $k < n$.

*I.S.* Prove that the claim holds when $x$ has a derivation of length $n$.

Consider such a derivation

$$a_1, \ldots a_i, \ldots, a_k, \ldots, \begin{array}{c} a_n \\ \| \\ x \end{array}$$

If $x \in \mathcal{I}$, then we are done by the *Basis*. Otherwise, say $x$ is the result of an operation (relation) $O_r \in \mathcal{O}$, applied on entries to the left of $x$, that is, say that $O_r(\ldots, x)$ is true —where we did not (have to) specify the inputs.

By the I.H. the inputs of $O_r$ all are in $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$. Now, since this closure is closed under $O_r(\ldots, x)$, we have that the output $x$ is in $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ too. □

So now we have two *equivalent* (5.2.5) approaches to defining inductively defined sets $S$: As $S = \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ or as $S = \{x : x \text{ is } (\mathcal{I}, \mathcal{O})\text{-derived}\}$.

The first approach is best when you want to prove properties of all members of the set $S$. The second is best when you want to show $x \in S$, for some specific $x$.

**5.2.6 Example.** Let $A = \{a, b\}$. We call $A$ an "*alphabet*".

Let $\mathcal{I} = \{\lambda\}$, $\lambda$ being (the name of) the *empty string*. Let us denote string concatenations by putting the strings we want to concatenate next to each other. E.g., concatenate *aaa* and *bbbaa* to obtain *aaabbbaa*. Also, if $X$ denotes a string, and so does $Y$, then $XY$ denotes the concatenation of the strings (denoted by) $X$ and $Y$ in that order. Similarly, $Xa$ means the result of concatenating string $X$ with the (length-1) string $a$, in that order. The *length* of a string over $A$ is the number of occurrences in the string (with repetitions) of $a$ and $b$.

We denote by $A^+$ the set of all strings of non zero length formed using the symbols $a$ and $b$. $A^*$ is defined to be $A^+ \cup \{\lambda\}$. Let $\mathcal{O}$ consist of the operations $O_a$ and $O_b$:

$$X \longrightarrow \boxed{O_a} \longrightarrow Xa \tag{1}$$

and

$$X \longrightarrow \boxed{O_b} \longrightarrow Xb \tag{2}$$

We claim that $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) = A^*$.

1. For $\mathrm{Cl}(\mathcal{I}, \mathcal{O}) \subseteq A^*$ we do induction over the closure to prove that any $x \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ satisfies $x \in A^*$ ("the property").

    - Well, if $x \in \mathcal{I}$ then $x = \lambda$. But $\lambda \in A^*$.
    - The property propagates with each of $O_a$ and $O_b$. For example, if $X \in A^*$, then since $Xa$ is also a string over the alphabet $A$, we have $Xa \in A^*$. Similarly for $O_b$. Done.

2. For $\mathrm{Cl}(\mathcal{I},\mathcal{O}) \supseteq A^*$ we do induction over $\mathbb{N}$ on $n = |Y|$ —the length of $Y$— to prove that any $Y \in A^*$ satisfies $Y \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$ ("the property").
   - Basis. $n = 0$. Then $Y = \lambda \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I},\mathcal{O})$. Done.
   - *I.H.* **Assume** claim for fixed $n$.
   - *I.S.* **Prove** for $n+1$. If $|Y| = n+1$ then $Y = Xa$ or $Y = X'b$ for some $X$ or $X'$ of length $n$. Say, it is $Y = Xa$. By I.H. $X \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$. But since $\mathrm{Cl}(\mathcal{I},\mathcal{O})$ is $\mathcal{O}$-closed, we have $Y = Xa \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$ by (1). The $Y = X'b$ case is entirely similar. □

**5.2.7 Example.** Let $A = \{a, b\}$ again.

Let $\mathcal{I} = \{\lambda\}$, let $\mathcal{O}$ consist of one operation $R$:

$$X \longrightarrow \boxed{R} \longrightarrow aXb \tag{3}$$

We claim that $\mathrm{Cl}(\mathcal{I},\mathcal{O}) = \{a^n b^n : n \geq 0\}$, where for any string $X$,

$$X^n \stackrel{Def}{=} \underbrace{XX\ldots X}_{n \text{ copies of } X}$$

If $n = 0$, "0 copies of $X$" means $\lambda$.

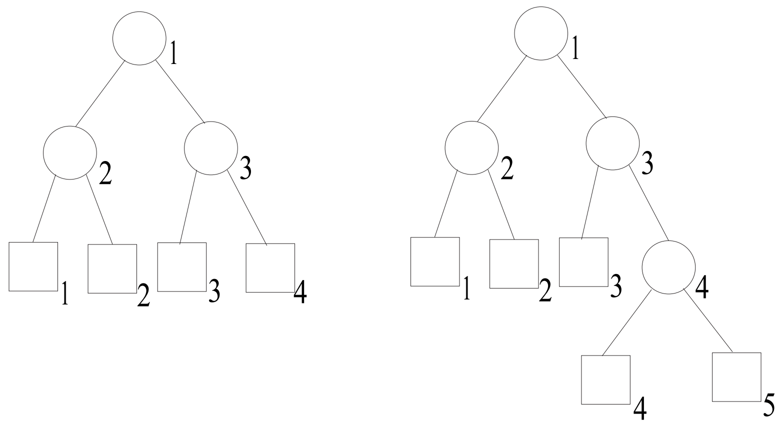
Let us write $S = \{a^n b^n : n \geq 0\}$.

1. For $\mathrm{Cl}(\mathcal{I},\mathcal{O}) \subseteq S$ we do induction over the closure to prove that any $x \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$ satisfies $x \in S$ ("the property").
   - Well, if $x \in \mathcal{I}$ then $x = \lambda = a^0 b^0$. Done.
   - The property propagates with each of $R$. For example, say $x = a^n b^n \in S$. Using (3) we see that the output, $axb$, is $a^{n+1} b^{n+1} \in S$. The property does propagate! Done.
2. For $\mathrm{Cl}(\mathcal{I},\mathcal{O}) \supseteq S$ we do induction over $\mathbb{N}$ on $n$ of $x = a^n b^n$ (arbitrary member of $S$) to prove that any $x \in S$ satisfies $x \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$ ("the property").
   - Basis. $n = 0$. Then $x = \lambda \in \mathcal{I} \subseteq \mathrm{Cl}(\mathcal{I},\mathcal{O})$. Done.
   - *I.H.* **Assume** claim for fixed $n$.
   - *I.S.* **Prove** for $n+1$. Thus $x = a^{n+1} b^{n+1} = aa^n b^n b$. By the I.H., $a^n b^n \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$. By (3) —recall that $\mathrm{Cl}(\mathcal{I},\mathcal{O})$ is $\mathcal{O}$-closed— we get the output $aa^n b^n b = a^{n+1} b^{n+1} \in \mathrm{Cl}(\mathcal{I},\mathcal{O})$. □

**5.2.8 Example. (Extended Binary Trees)** This is a longish example with some preliminary discussion up in front. We want to define the term known as "Tree". This term refers to a structure, which uses as building blocks —called **nodes**— the members of the enumerable set below

$$A = \{\bigcirc_0, \bigcirc_1, \bigcirc_2, \ldots; \square_0, \square_1, \square_2, \ldots\}$$

Trees look something like this:




The qualifier "extended" is due to the presence of square nodes. We will not define simple trees (they have round nodes only).

These *nodes* are made *distinct* by the use of *subscripts*. The symbols in the set $A$ are *distinguished* by their *type*, "round" vs. "square", and *within each type* by their natural number index. Thus, $\bigcirc_i \neq \bigcirc_j$ iff $i \neq j$, $\square_i \neq \square_j$ iff $i \neq j$, and $\bigcirc_i \neq \square_j$, *for all* $i, j$.

One feature in both of the above drawings is essential to note (blue type below):

**Circular** or **square** nodes are connected by line segments. Walking in the vertical direction from the top of the page towards the bottom, **no nodes are *ever* shared.** In particular, in all the examples above where we have more than one node, you will notice that the two *sets* of nodes that "hang below" the top node (left and right of it) are *disjoint*. **We need to include this requirement in our definition**.

But clearly these sets of notes have "geometric structure" (*position*: left/right; and *connections*: via line segments)! They are not "flat" sets like $\{\bigcirc_5, \square_{11}\}$.

And yet, in the *mathematical definition below* we will need to *state* the blue condition: the left and right, when you "forget" the lines and positions, become disjoint flat sets. This observation is what *imposes* some complexity in the definition, which defines the "structure" *and* the "flat" set that supports the structure (the set of nodes in the tree) *simultaneously*.

We define an *extended binary tree* as a *member* of the inductively defined set of *E-Trees*. It is intended that each e-tree of the inductively defined set of all trees is an *ordered pair*:

(flat set of its nodes, geometric tree structure)

The "geometric tree structure" is **mathematically given** in a *one-dimensional depiction* of the trees.

For example, the first tree in the figure above is linearly represented by

$$\Big(\big(\square_1, \bigcirc_2, \square_2\big), \bigcirc_1, \big(\square_3, \bigcirc_3, \square_4\big)\Big)$$

To appreciate the issue of "structure vs. flat set of nodes" let us first write the above as

$$\Big(\big(a, b, c\big), d, \big(f, g, h\big)\Big) \tag{2}$$

How easy is to obtain the flat set of nodes $a$–$h$? Easy via naked eye for very small trees, hard for large ones.

So, why not forget flat structure and just say "left and right parts of a tree must be disjoint"? Because such parts are not sets (sets do not have "structure", like "edges"), and the term "disjoint" refers to sets.

Here (2) is shorthand for something really complex, namely

$$(a, b, c) = \Big\{\{a, \{a, b\}\}, \{a, c, \{a, b\}\}\Big\}$$

Suppose now that $b = g$, but all other letters $(a, c, d, f, h)$ are distinct. Thus $(f, g, h) = (f, b, h) = \Big\{\{f, \{f, b\}\}, \{f, h, \{f, b\}\}\Big\}$ and hence

$$\big(a, b, c\big) \cap \big(f, g, h\big) = \emptyset \tag{$t$}$$

So test $(t)$ does *NOT* give me the information I *need* before I build the tree in (2). Apparently it is wrong to do so, as $b = g$.

I do need the information the flat set of nodes gives me, for the decision. See definition below for the details!

Thus our definition below builds the flat set —called the *support* of the tree— of nodes of a tree *at the same time as it builds the structure of the tree.*

**5.2.9 Definition.** We define the *set* of *all extended trees* —or just *trees*— $ET$, as $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$ where:

1. First, chose as the set of initial objects

$$\mathcal{I} = \Big\{(\emptyset, \square_0), (\emptyset, \square_1), (\emptyset, \square_2), \ldots\}$$

2. $\mathcal{O}$ has just one rule with a constraint on the input: If $F_X \cap F_Y = \emptyset$ **and** $\bigcirc_i \notin F_X \cup F_Y$, then

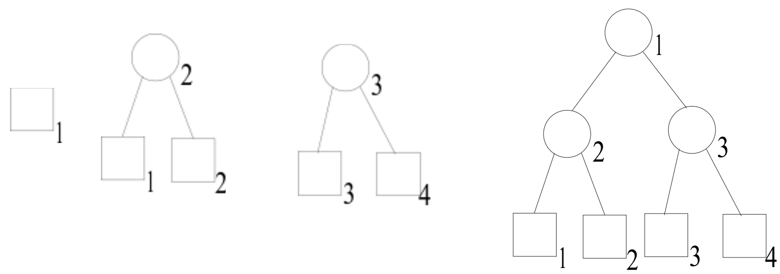
$$\left.\begin{array}{c} (F_X, X) \longrightarrow \\ \bigcirc_i \longrightarrow \\ (F_Y, Y) \longrightarrow \end{array}\right\} \boxed{\text{form tree}} \longrightarrow \Big(F_X \cup F_Y \cup \{\bigcirc_i\},\ (X, \bigcirc_i, Y)\Big)$$

3. For *each* $(S, T) \in \mathrm{Cl}(\mathcal{I}, \mathcal{O})$ we say that $T$ is an extended tree, and $S$ is its *support*, that is, the "flat" set nodes from the set $A$ used to build $T$. We indicate this relationship by

$$S = \sup(T)^{\dagger}$$

If $T = (X, \bigcirc_i, Y)$, then we say that $\bigcirc_i$ is the **root** of $T$, while $X$ is its **left** and $Y$ is its **right subtree**. □

Thus, some immediate examples of trees are



Indeed, using 5.2.5, the leftmost example is a tree since it is the right component of the pair $(\emptyset, \Box_1)$. The next tree is built via the derivation —written linearly,

$$(\emptyset, \Box_1), (\emptyset, \Box_2), \Big(1, (\Box_1, \bigcirc_2, \Box_2)\Big)$$

The next derivation builds both the 2nd and 3rd trees:

$$(\emptyset, \Box_1), (\emptyset, \Box_2), \Big(1, (\Box_1, \bigcirc_2, \Box_2)\Big), (\emptyset, \Box_3), (\emptyset, \Box_4), \Big(1, (\Box_3, \bigcirc_3, \Box_4)\Big)$$

The 4th tree has this as a derivation:[‡]

$$(\emptyset, \Box_1), (\emptyset, \Box_2), \Big(1, (\Box_1, \bigcirc_2, \Box_2)\Big), (\emptyset, \Box_3), (\emptyset, \Box_4), \Big(1, (\Box_3, \bigcirc_3, \Box_4)\Big),$$
$$\Bigg(3, \Big(1, (\Box_1, \bigcirc_2, \Box_2)\Big), \bigcirc_1, \Big(1, (\Box_3, \bigcirc_3, \Box_4)\Big)\Bigg)$$

The support of the 4th tree is the flat set $\{\bigcirc_1, \bigcirc_2, \bigcirc_3\}$. □

**5.2.10 Example. (Trees —continued)** Hmm! Seems like we are not including square nodes in the support. See how the *support* of all nodes in $\mathcal{I}$ is $\emptyset$ for each entry. Why so?

In the words of Knuth ([Knu73]) "trees is the most important nonlinear structure arising in computing algorithms". The extended tree is an abstraction of trees that we implement with computer programs, where round nodes are the

[†]**Caution**: As for many other symbols, "sup" means something else in the context of POsets. We will not get into this!

[‡]Derivations are not unique as is clear from Example 5.2.2.

*only ones that can carry data.* The lines are (implicitly) pointing downwards. They are *pointers*, in computer jargon. For example, the topmost leftmost line in the fourth tree above points to the node $\bigcirc_2$. Practically it means that if your program is processing node $\bigcirc_1$, then it can transfer to and process node $\bigcirc_2$ if it wishes. It knows the address of $\bigcirc_2$. The pointer holds this address as value.

Which brings me to square nodes! Together with the line planted on them, they are notation for *null* pointers! They point nowhere. So square nodes cannot hold information, *that is why they do not contribute to the support of the tree.*

The computer scientist calls round nodes "internal" and calls square nodes "external".

Finally, how do the lines —called *edges*— get inserted? We defined "root" for trees, as well as "left subtree" and "right subtree". So, to draw lines and draw a tree that is given mathematically as $(X, \bigcirc_r, Y)$, we *call* recursively the process that does it on (inputs) $X$ and $Y$. Then add two more edges: One from $\bigcirc_r$ to the root of $X$ and one from $\bigcirc_r$ to the root of $Y$.

How does the recursion terminate? Well, if your tree is just $\Box_j$, then there is nothing to draw. $\Box_j$ is the root. This is the basis of the recursive procedure: do nothing. □

Here is something interesting about all extended trees:

**5.2.11 Proposition.** *In any extended tree, the number of square nodes exceeds by one the number of round nodes.*

*Proof.* Induction over the set of all trees (5.2.9) $\mathrm{Cl}(\mathcal{I}, \mathcal{O})$.

1. *Basis.* For any $(\emptyset, \Box_i)$, the tree-part (structure-part) is just $\Box_i$. One square node, 0 round nodes. Done.

2. The property propagates with the only tree-builder operation:

$$\left.\begin{array}{r}(F_X, X) \longrightarrow \\ \bigcirc_i \longrightarrow \\ (F_Y, Y) \longrightarrow\end{array}\right\} \boxed{\text{form tree}} \longrightarrow \Big(F_X \cup F_Y \cup \{\bigcirc_i\},\ (X, \bigcirc_i, Y)\Big)$$

Indeed, *suppose* that $X$ has $\phi$ internal (round) and $\varepsilon$ external (square) nodes. Let also $Y$ have $\phi'$ internal and $\varepsilon'$ external nodes.

The assumption *on the input side* is then (I.H.) that

$$\phi + 1 = \varepsilon \tag{1}$$

and

$$\phi' + 1 = \varepsilon' \tag{2}$$

The *output side* of the operation has the tree $(X, \bigcirc_i, Y)$. This has $\Phi = \phi + \phi' + 1$ internal nodes and $E = \varepsilon + \varepsilon'$ external ones. Using (1) and (2) we have

$$\Phi = \varepsilon + \varepsilon' - 1 = E - 1$$

Seeing that this is the property we want to prove on the output side, indeed the property propagates with the rule. Done. □

# Bibliography


[Dav65] M. Davis, *The undecidable*, Raven Press, Hewlett, NY, 1965.

[Hin78] P. G. Hinman, *Recursion-theoretic hierarchies*, Springer-Verlag, New York, 1978.

[Kle43] S.C. Kleene, *Recursive predicates and quantifiers*, Transactions of the Amer. Math. Soc. **53** (1943), 41–73, [Also in [Dav65], 255–287].

[Knu73] Donald E. Knuth, *The Art of Computer Programming; Fundamental Algorithms*, 2nd ed., vol. 1, Addison-Wesley, 1973.

[Kur63] A.G. Kurosh, *Lectures on General Algebra*, Chelsea Publishing Company, New York, 1963.

[Tou03a] G. Tourlakis, *Lectures in Logic and Set Theory, Volume 1: Mathematical Logic*, Cambridge University Press, Cambridge, 2003.

[Tou03b] ———, *Lectures in Logic and Set Theory, Volume 2: Set Theory*, Cambridge University Press, Cambridge, 2003.

[Tou08] ———, *Mathematical Logic*, John Wiley & Sons, Hoboken, NJ, 2008.