1 Arithmetic

1.1 Axioms for Arithmetic

*Example: Axioms for equality*

Recall the axioms of equality:

EQ1: \( \forall x \, x = x \) is an axiom.

EQ2: For each choice of formula \( \alpha \) and variable \( z \),

\[
\forall x_1 \forall x_2 \left( x_1 = x_2 \rightarrow (\alpha[x_1/z] \rightarrow \alpha[x_2/z]) \right)
\]

is an axiom.

The following two rules are consequences.

EqSubs: For each variable \( z \) and terms \( r, t_1 \) and \( t_2 \),

\[
t_1 = t_2 \Rightarrow r[t_1/z] = r[t_2/z].
\]

EqTrans(\( k \)): For each term \( t_1, \ldots, t_{k+1} \),

\[
t_1 = t_2 \Rightarrow \cdots \Rightarrow t_k = t_{k+1}.
\]

*Natural Numbers*

Fix the domain as \( \mathbb{N} \), the natural numbers. Interpret the constant symbol 0 as zero and the unary function symbol \( s \) as successor. Thus each number in \( \mathbb{N} \) has a term: \( 0 \), \( s(0) \), \( s(s(0)) \), \( s(s(s(0))) \), \( \ldots \).

To reason about natural numbers, we use the following Peano Axioms.\(^1\)

PA1: \( \forall x \, s(x) \neq 0 \). “Zero is not a successor.”

PA2: \( \forall x \forall y (s(x) = s(y) \rightarrow x = y) \). “Nothing has two predecessors.”

PA3: \( \forall x (x + 0 = x) \). Adding zero to any number yields the same number.

PA4: \( \forall x \forall y (x + s(y) = s(x + y)) \). Adding a successor yields the successor of adding the number.

PA5: \( \forall x \, x \cdot 0 = 0 \). Multiplying by zero yields zero.

PA6: \( \forall x \forall y \, x \cdot s(y) = x \cdot y + x \). Multiplication by a successor.

The six axioms above define + and \( \cdot \) for any particular numbers. They do not, however, allow us to reason adequately about all numbers. For that, we use an additional axiom: induction.

PA7: For each formula \( \varphi \) and variable \( v \),

\[
\varphi[0/v] \rightarrow \left( (\forall v (\varphi \rightarrow \varphi[s(v)/v])) \rightarrow (\forall v \varphi) \right)
\]

is an axiom.

\(^1\)Named in honor of Guiseppe Peano. His axioms were actually somewhat different, but credit is due anyway.
The formula $\varphi$ represents the “property” to be proved.

To prove $\varphi$ for every $x$, we can prove the base case $\varphi[0/x]$ and the inductive case $\forall x (\varphi \rightarrow \varphi[s(x)/x])$.

**A first proof in PA**

To illustrate the use of induction, we shall consider a proof that every non-zero number is a successor. Formally, this is

$$\vdash_{PA} \forall x \left( (x \neq 0) \rightarrow (\exists y (s(y) = x)) \right) .$$

Just as in the case of an informal induction, we must identify the statement to be used in the induction. In other words, we must select a formula $\varphi$ for use in Axiom PA7. It seems very natural to select $\varphi$ so that $\forall x \varphi$ is in fact what we want to prove; that is, to take $\varphi$ to be the formula

$$(x \neq 0) \rightarrow (\exists y (s(y) = x)) .$$

Given this choice, it suffices to prove $\varphi[0/x]$ and $\forall x (\varphi \rightarrow \varphi[s(x)/x])$; then two uses of *modus ponens* will complete the proof.

**Step I:** to prove $\varphi[0/x]$, which is $(0 \neq 0) \rightarrow (\exists y (s(y) = 0))$.

The second clause may look ominous, but the first comes to the rescue: it is false.

1. $0 \neq 0$ Assumption
2. $0 = 0$ EQ1 + $\forall$e
3. $\perp$ $\neg$e: 2, 1
4. $\exists y (s(y) = 0)$ $\perp$e
5. $(0 \neq 0) \rightarrow (\exists y (s(y) = 0))$ $\rightarrow$i

**Step II:** Since we require a $\forall$ quantifier; we take a fresh variable $x'$, for use in the $\forall$i rule.

We need to prove $\varphi[x'/x] \vdash_{PA} \varphi[s(x')/x]$, which is

$$(x' \neq 0) \rightarrow (\exists y (s(y) = x')) \vdash_{PA} (s(x') \neq 0) \rightarrow (\exists y (s(y) = s(x'))) .$$

Once again, a formidable-looking task becomes simple if we look at the final consequent: $\exists y (s(y) = s(x'))$. That, we can prove.

6. $x'$ fresh
7. $s(x') \neq 0$ Assumption
8. $s(x') = s(x')$ EQ1 + $\forall$e
9. $\exists y (s(y) = s(x'))$ $\exists$i [term $x'$]
10. $(s(x') \neq 0) \rightarrow (\exists y (s(y) = s(x'))) \rightarrow$i: 7–9
11. $\forall x (\exists y (s(y) = s(x'))) \forall$i: 6–10

With the parts in place, we can finish the proof.
12. \((0 \neq 0) \rightarrow (\exists y(s(y) = 0))\) \rightarrow
\left(\forall x\left(\left((x \neq 0) \rightarrow (\exists y(s(y) = x))\right) \rightarrow \left((s(x) \neq 0) \rightarrow (\exists y(s(y) = s(x)))\right)\right)\right) \rightarrow
\left(\forall x\left(\left((x \neq 0) \rightarrow (\exists y(s(y) = x))\right)\right)\right)

13. \(\forall x\left((x \neq 0) \rightarrow (\exists y(s(y) = x))\right) \rightarrow \left((s(x) \neq 0) \rightarrow (\exists y(s(y) = s(x)))\right)\)\)

14. \(\forall x((x \neq 0) \rightarrow (\exists y(s(y) = x)))\)

1.2 Associativity of Addition

Properties from the Peano Axioms

The Peano Axioms imply all of the familiar properties of the natural numbers. For example, addition is associative.

1. Lemma. Addition in Peano Arithmetic is associative; that is,
\[\vdash_{PA} \forall x \forall y \forall z (x + y) + z = x + (y + z)\].

(Notation “\(\vdash_{PA}\)” means “provable [in ND] using the EQ and PA axioms.”)

How can we find such a proof?

We must use induction (Axiom PA7). The key first step: choose a good formula \(\varphi\) for the induction property.

Setting Up the Induction

Recall Axiom PA7: \(\varphi[0/z] \rightarrow \left( (\forall z(\varphi \rightarrow \varphi[s(z)/z])) \rightarrow (\forall z \varphi) \right)\).

Choosing \(\varphi\) to be \((x + y) + z = x + (y + z)\) yields the instance
\((x + y) + 0 = x + (y + 0) \rightarrow \left( (\forall z((x + y) + z = x + (y + z) \rightarrow (x + y) + s(z) = x + (y + s(z)))) \rightarrow \forall z (x + y) + z = x + (y + z) \right)\)

There are some (generally unfamiliar) exceptions.
Thus we must prove the base case \((x + y) + 0 = x + (y + 0)\) and the inductive case
\[ \forall z \left( (x + y) + z = x + (y + z) \rightarrow (x + y) + s(z) = x + (y + s(z)) \right). \]

Then two uses of \(\rightarrow e\) yield the desired formula
\[ \forall z \ (x + y) + z = x + (y + z). \]

**The Base Case for Associativity**

To prove: \((x + y) + 0 = x + (y + 0)\).

How?

The axioms tell us how to add zero: \(\forall u(u + 0 = u)\). We can apply this to \(x + y\) and also to \(y\).

The rules for equality then give us what we want.

3. \((x + y) + 0 = x + y\) \quad PA3 + \forall e
4. \(y + 0 = y\) \quad PA3 + \forall e
5. \(x + (y + 0) = x + y\) \quad EqSubs\([x + \cdot]: 4\)
6. \((x + y) + 0 = x + (y + 0)\) \quad EqTrans: 3, 5

**Inductive Case for Associativity**

The inductive step shows that associativity with \(z\) implies associativity with \(s(z)\):
\[ \forall z \left( (x + y) + z = x + (y + z) \rightarrow (x + y) + s(z) = x + (y + s(z)) \right). \]

Axiom PA4 — \(\forall u(\forall v(u + s(v) = s(u + v)))\) — gives addition of successors. We will need it three times: once for \(y + s(z)\), once for \((x + y) + s(z)\), and once for \(x + s(y + z)\).

To get the \(\forall\) quantifier, we select a fresh variable \(z'\), in place of \(z\).
Proof of the Inductive Case

8. $z'$ fresh
9. $(x + y) + z' = x + (y + z')$ Assumption
10. $s(x + (y + z')) = s((x + y) + z')$ EqSubs[$s(\cdot)$]: 9
11. $(x + y) + s(z') = s((x + y) + z')$ PA4 + ∀e ($\times$2)
12. $y + s(z') = s(y + z')$ PA4 + ∀e ($\times$2)
13. $x + (y + s(z')) = x + s(y + z')$ EqSubs[$x + \cdot$]: 12
14. $x + s(y + z') = s(x + (y + z'))$ PA4 + ∀e ($\times$2)
15. $(x + y) + s(z') = x + (y + s(z'))$ EqTrans: 13, 14, 10, 11
16. $A(x, y, z') \rightarrow A(x, y, s(z'))$ →i: 9–15
17. $\forall z (A(x, y, z) \rightarrow A(x, y, s(z)))$ ∀i: 8–16

(Note: $A(a, b, c)$ abbreviates $(a + b) + c = a + (b + c)$.)

Completing the Proof

Now that we have the base case and the inductive case, we must

- Combine them and complete the induction, and
- Add the quantifiers onto $x$ and $y$.

The full proof appears next.
**Associativity: the full proof**

1. \( x, y \) fresh

2. \[ A(x, y, 0) \rightarrow \left( \left( \forall z (A(x, y, z) \rightarrow A(x, y, s(z))) \right) \rightarrow \left( \forall z A(x, y, z) \right) \right) \] PA7

3. \( (x + y) + 0 = x + y \) PA3 + ∀e

4. \( y + 0 = y \) PA3 + ∀e

5. \( x + (y + 0) = x + y \) EqTrans: 3, 5

6. \( x + (y + 0) = (x + y) + 0 \) EqTrans: 3, 5

7. \[ \left( \forall z (A(x, y, z) \rightarrow A(x, y, s(z))) \right) \rightarrow \left( \forall z A(x, y, z) \right) \rightarrow e: 6, 2 \]

8. \( z \) fresh

9. \( (x + y) + z = x + (y + z) \) Assumption

10. \( s(x + (y + z)) = s((x + y) + z) \) EqSubs[\( s(\cdot) \)]; 9

11. \( (x + y) + s(z) = s((x + y) + z) \) PA4 + ∀e(×2)

12. \( y + s(z) = s(y + z) \) PA4 + ∀e(×2)

13. \( x + (y + s(z)) = x + s(y + z) \) EqTrans: 12

14. \( x + s(y + z) = s(x + (y + z)) \) PA4 + ∀e(×2)

15. \( (x + y) + s(z) = x + (y + s(z)) \) EqTrans: 13, 14, 10, 11

16. \[ \forall z \left( \left( (x + y) + z = x + (y + z) \right) \rightarrow \left( (x + y) + s(z) = x + (y + s(z)) \right) \right) \rightarrow i: 9–15 \]

17. \[ \forall z \left( \left( (x + y) + z = x + (y + z) \right) \rightarrow \left( (x + y) + s(z) = x + (y + s(z)) \right) \right) \rightarrow i: 8–16 \]

18. \( \forall z ((x + y) + z = x + (y + z)) \rightarrow e: 7, 17 \)

19. \( \forall x \left( \forall y \left( \forall z ((x + y) + z = x + (y + z)) \right) \right) \rightarrow i(\times 2): 1–18 \)

**Recapitulation**

The formula to be proven had three universal quantifiers: on \( x, y, \) and \( z \).

Q: Why did we do induction on only one?

A: We only needed one. Rule ∀i worked fine for the other two.

Q: Did it matter which variable we chose for the induction?

A: Yes!

Consider what would happen in an induction on \( x \). The base would be to derive

\[(0 + y) + z = 0 + (y + z) .\]
But that’s not easy—Axiom PA3 has \( u + 0 \), not \( 0 + u \).

1.3 Commutativity of Addition

**Commutativity of Addition**

_Theorem:_ Addition in Peano Arithmetic is commutative; that is,

\[
\vdash_{PA} \forall x \forall y (x + y = y + x) .
\]

To do the proof, we will need to use induction both on \( x \) and on \( y \).

I have chosen to do \( x \) first. Thus I let \( \varphi \) be the formula \( \forall y (x + y = y + x) \). The corresponding instance of PA7 is

\[
\forall y (0 + y = y + 0) \rightarrow \\
\left( \forall x (\forall y (x + y = y + x) \rightarrow \forall y (s(x) + y = y + s(x))) \rightarrow \\
\forall x \forall y (x + y = y + x) \right)
\]

Thus we must prove the base case, \( \forall y (0 + y = y + 0) \), and the inductive case,

\[
(\forall x (\forall y (x + y = y + x) \rightarrow \forall y (s(x) + y = y + s(x))) .
\]

**The Base Case for Commutativity**

To prove: \( \forall y (0 + y = y + 0) \).

To prove this, we must use induction on \( y \). Let’s make it a lemma:

2. **Lemma.** Peano Arithmetic has a proof of \( \forall y (0 + y = y + 0) \).

**Proof** of lemma. We use induction (PA7) with \( 0 + y = y + 0 \) for \( \varphi \).

**Basis:** Prove \( 0 + 0 = 0 + 0 \). Immediate from EQ1.

**Inductive step:** Prove \( \forall y ((0 + y = y + 0) \rightarrow (0 + s(y) = s(y) + 0)) \).

Pick \( y' \) fresh. Then assume \( 0 + y' = y' + 0 \) (i.e., start a subproof). We get

\[
\begin{align*}
0 + s(y') &= s(0 + y') & \text{PA4} \\
&= s(y' + 0) & \text{assumption + EqSubs[\(s(\cdot)\)]} \\
&= s(y') + 0 & \text{PA3} .
\end{align*}
\]

Applying \( \rightarrow i \) and generalization \( (\forall i) \) yield the required formula.

Using PA7 and \( \rightarrow e \) (twice) completes the proof of the lemma.
1. \[ 0 + 0 = 0 + 0 \]  
   EQ1 + \forall e

2. \[ y \text{ fresh} \]

3. \[ 0 + y = y + 0 \]  
   Assumption

4. \[ 0 + s(y) = s(0 + y) \]  
   PA4 + \forall e

5. \[ s(0 + y) = s(y + 0) \]  
   EqSubs[s(\cdot)]; 3

6. \[ y + 0 = y \]  
   PA3 + \forall e

7. \[ s(y + 0) = s(y) \]  
   EqSubs[s(\cdot)]; 6

8. \[ s(y) + 0 = s(y) \]  
   PA3 + \forall e

9. \[ 0 + s(y) = s(y) + 0 \]  
   EqTrans (\times 3): 4, 5, 7, 8

10. \[ 0 + y = y + 0 \rightarrow 0 + s(y) = s(y) + 0 \]  
   \rightarrow i: 3–9

11. \[ \forall y (0 + y = y + 0 \rightarrow 0 + s(y) = s(y) + 0) \]  
   \forall i: 10

12. \[ \forall y (0 + y = y + 0) \]  
   PA7 + \rightarrow e (\times 2): 1, 11

**Inductive Case (on variable \( x \)) for Commutativity**

3. **Lemma.** \[ \forall y (x + y = y + x) \vdash_{PA} \forall y (s(x) + y = y + s(x)) \]. (“If \( x \) commutes with everything, then \( s(x) \) also does.”)

Plan of proof: induction on variable \( y \), for formula \( s(x) + y = y + s(x) \).

Basis: \( s(x) + 0 = 0 + s(x) \). Already proven (Lemma 2).

Ind. step: for a fresh variable \( y' \), we need to show

\[
(s(x) + y' = y' + s(x)) \rightarrow (s(x) + s(y') = s(y') + s(x)) .
\]

Note that the premise of the Lemma implies both \( x + y' = y' + x \) and \( x + s(y') = s(y') + x \).
That is, we use \( \forall e \) on the premise *twice*, with different terms.

We calculate the sums in our goal:

\[
s(x) + s(y') = s(s(x) + y') \quad \text{PA4}
\]
\[
= s(y' + s(x)) \quad \text{assumption + EqSubs[s(\cdot)]}
\]
\[
= s(s(y' + x)) \quad \text{PA4 + EqSubs[s(\cdot)]}
\]

and

\[
s(y') + s(x) = s(s(y') + x) \quad \text{PA4}
\]
\[
= s(x + s(y')) \quad \text{premise of Lemma + EqSubs[s(\cdot)]}
\]
\[
= s(s(x + y')) \quad \text{PA4 + EqSubs[s(\cdot)]} .
\]
Recall we have \(x + y' = y' + x\); thus \(\text{EqSubs}[s(s(\cdot))]\) yields the required
\[
s(x) + s(y') = s(y') + s(x).
\]

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>12.</td>
<td>(\forall y \ x + y = y + x)</td>
<td>Premise</td>
</tr>
<tr>
<td>13.</td>
<td>(s(x) + 0 = 0 + s(x))</td>
<td>(\forall e: 12)</td>
</tr>
<tr>
<td>14.</td>
<td>(y') fresh</td>
<td></td>
</tr>
<tr>
<td>15.</td>
<td>(x + y' = y' + x)</td>
<td>(\forall e: 12)</td>
</tr>
<tr>
<td>16.</td>
<td>(x + s(y') = s(y') + x)</td>
<td>(\forall e: 12)</td>
</tr>
<tr>
<td>17.</td>
<td>(s(x) + y' = y' + s(x))</td>
<td>Assumption</td>
</tr>
<tr>
<td>18.</td>
<td>(Uses of PA4 omitted)</td>
<td></td>
</tr>
<tr>
<td>22.</td>
<td>(Uses of EqSubs on 17–21 omitted)</td>
<td></td>
</tr>
<tr>
<td>27.</td>
<td>(s(x) + s(y') = s(y') + s(x))</td>
<td>EqTrans ((\times 6): 17–26)</td>
</tr>
<tr>
<td>28.</td>
<td>(s(x) + y' = y' + s(x))</td>
<td>(\rightarrow i: 17–27)</td>
</tr>
<tr>
<td>29.</td>
<td>((s(x) + y = y + s(x)) \rightarrow (s(x) + s(y) = s(y) + s(x)))</td>
<td>(\forall i: 14–28)</td>
</tr>
</tbody>
</table>

**Putting It All Together**

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>0 + 0 = 0</td>
<td>PA3 + (\forall e)</td>
</tr>
<tr>
<td>2.</td>
<td>Lemma 2</td>
<td></td>
</tr>
<tr>
<td>12.</td>
<td>(\forall y \ 0 + y = y + 0)</td>
<td>PA7 + (\rightarrow e (\times 2): 1, 11)</td>
</tr>
<tr>
<td>13.</td>
<td>(\forall y \ x + y = y + x)</td>
<td>Assumption</td>
</tr>
<tr>
<td>29.</td>
<td>((s(x) + y = y + s(x)) \rightarrow (s(x) + s(y) = s(y) + s(x)))</td>
<td>(\forall i: 14–28)</td>
</tr>
</tbody>
</table>

The other familiar properties of addition and multiplication have similar proofs. One can continue: divisibility, primeness, etc.
1.4 Definability

4. Definition. Let formula \( \varphi \) have free variables \( x_1, \ldots, x_k \).

Given an interpretation \( I \), a formula \( \varphi \) defines the \( k \)-ary relation of tuples that make \( \varphi \) true — that is, the relation

\[
R_\varphi = \left\{ (a_1, \ldots, a_k) \in \text{dom}(I)^k \mid \varphi(I, E[x_1 \mapsto a_1] \cdots [x_k \mapsto a_k]) = T \right\}.
\]

A relation \( R \) is definable (in \( I \)) iff \( R = R_\varphi \) for some formula \( \varphi \).

Example: in Peano Arithmetic, the relation \( \leq \) is defined by the formula \( \exists z (x_1 + z = x_2) \).

Given a defined relation \( R_\varphi \), we can use it in a formula as if it were a symbol of the language. An “atom” \( R_\varphi(t_1, \ldots, t_k) \) is simply an alternate notation for the formula \( (t_1/x_1) \cdots (t_k/x_k) \varphi \).

Properties of Defined Relations

The PA axioms allow one to show that the defined relation \( \leq \) has the expected properties.

- \( x \leq y \) and \( y \leq z \) imply \( x \leq z \) (transitivity).
- If \( x \leq y \) and \( y \leq x \) then \( x = y \) (asymmetry).

We can also define further relations using \( \leq \); e.g.,

\[
x < y \quad \text{iff} \quad (x \leq y) \land (x \neq y).
\]

Transitivity of Less-or-Equal

To show: \( \{ x \leq y, y \leq z \} \vdash x \leq z \).

1. \( \exists w (x + w = y) \) \hspace{1cm} \text{Premise [} x \leq y \text{]} \\
2. \( \exists w (y + w = z) \) \hspace{1cm} \text{Premise [} y \leq z \text{]} \\
3. \( x + u = y, \ u \text{ fresh} \hspace{1cm} \text{Assumption} \\
4. \( y + v = z, \ v \text{ fresh} \hspace{1cm} \text{Assumption} \\
5. \( x + (u + v) = (x + u) + v \hspace{1cm} \text{Assoc} iativity \text{ of } + \\
6. \( (x + u) + v = y + v \hspace{1cm} \text{EqSubs}[: + v]: 3 \\
7. \( x + (u + v) = z \hspace{1cm} \text{EqTrans(2): 5, 6, 4} \\
8. \( \exists w (x + w = z) \hspace{1cm} \exists i: 7 \\
9. \( \exists w (x + w = z) \hspace{1cm} \exists: 2, 4–8 \\
10. \( \exists w (x + w = z) \hspace{1cm} \exists: 1, 3–9 
\)
Defining Functions

To define a \( k \)-ary function, use its \((k + 1)\)-ary relation.

Example: Let \( R_{sq} \) ("square-of") be defined by \( x_1 \cdot x_1 = x_2 \).
Then we can get the effect of having the squaring function:

if \( \varphi \) contains a free variable \( x \), but \( u \) is fresh, then the formula

\[
\exists u (R_{sq}(t, u) \land \varphi[u/x])
\]

expresses “the square of \( t \) satisfies \( \varphi \).”

We must, however, ensure that \( R_{sq} \) really does define a function;
that is, that every number has exactly one square:

\[
\forall x \left( (\exists y (R_{sq}(x, y)) \land \forall y \forall z ((R_{sq}(x, y) \land R_{sq}(x, z)) \rightarrow y = z) ) \right).
\]

We leave this proof as an exercise.

2 Lists

Lists are a basic structure in computer science. We shall give axioms for lists, similar to the PA axioms.

Vocabulary for lists:

- a constant symbol \( e \), for the empty list, and
- a binary function symbol \( \text{cons} \).

The term \( \text{cons}(a, b) \) will denote the list with \( a \) as its first element and \( b \) as the remainder of the list.

Lists are Sequences

A list is a sequence of zero or more things, called “items”.

Informally, we shall write a list by writing its items between angle brackets. Examples:

\[ \langle a, b \rangle \]: the list containing \( a \) and \( b \), only.
\[ \langle b, a \rangle \]: a different list—the order matters.
\[ \langle \rangle \]: the empty list.

What can be an item? Anything!

\[ \langle p \land q \rangle \]: an expression is, technically, a list—also denotable as \( \langle (, p, \land, q, ) \rangle \).

(This assumes that \( (, \), etc., are in the domain—either as themselves or in the form of “codes” for them.)
\langle\langle\rangle, \langle\rangle\rangle$: items can themselves be lists.
\langle a, a, a \rangle$: items may repeat—this list has three, all the same.

Basic lists

Lists and Natural Numbers

We define lists by an analogy to the natural numbers.
Recall that the natural numbers start with the constant \(0 \in \mathbb{N}\) and use the successor function \(s\) to generate any natural number.

\[0, s(0), s(s(0)), s(s(s(0))), \ldots\]

Similarly, lists start with the empty list \(e\), and the \(\text{cons}\) function generates new lists out of previous ones.

\[e, \text{cons}(e, e), \text{cons}(e, \text{cons}(e, e)), \text{cons}(e, \text{cons}(e, \text{cons}(e, e))), \ldots\]

We also get other objects, when the first argument to \(\text{cons}\) is not \(e\).

What’s a List?

A domain of lists must contain the empty list \(e\). It must also contain \(\text{cons}(e, e)\), and so on:

\[
\text{cons}(e, \text{cons}(e, e)), \\
\text{cons}(e, \text{cons}(e, \text{cons}(e, e))), \ldots
\]

We shall regard the above objects as lists whose elements are all \(e\). In general, if we want a list containing \(a_1, a_2, \ldots, a_k\) we use the object formed as

\[
\text{cons}(a_1, \text{cons}(a_2, \text{cons}(\ldots \text{cons}(a_k, e) \ldots)))
\]

The values \(a_i\) can be anything in the domain.
For now, the values on a list will themselves be lists.
Examples

For example, the list containing the three objects \( \text{cons}(e, e), e \) and \( \text{cons}(\text{cons}(e, e), e) \), in that order, is

\[
\text{cons}\left(\text{cons}(e, e), \text{cons}\left(e, \text{cons}(\text{cons}(e, e), e)\right)\right).
\]

A short-hand notation: “angle brackets.”

- \( \langle \rangle \) denotes the empty list \( e \).
- For any term \( a \), \( \langle a \rangle \) denotes the list whose single item is \( a \), i.e., the object \( \text{cons}(a, e) \) (which is also \( \text{cons}(a, \langle \rangle) \)).
- Suppose that \( \ell \) is an expression such that \( \langle \ell \rangle \) denotes a non-empty list. For a term \( a \), \( \langle a, \ell \rangle \) denotes the list whose first item is \( a \) and whose remaining items are the items on the list \( \langle \ell \rangle \). That is, \( \langle a, \ell \rangle \) denotes the list \( \text{cons}(a, \langle \ell \rangle) \).

Note: \( \langle a, \ell \rangle \) is NOT the same as \( \langle a, \langle \ell \rangle \rangle \)!

Examplecises

1. What is the “angle bracket” form of the list \( \text{cons}(\text{cons}(e, e), \text{cons}(\text{cons}(e, e), e)) \)?

The sub-term \( \text{cons}(e, e) \) is the one-element list \( \langle e \rangle \).
The whole term contains that list twice, yielding \( \langle \langle e \rangle, \langle e \rangle \rangle \).

2. What is the explicit term denoted by \( \langle e, e, e \rangle \)?

The list \( \text{cons}(e, \text{cons}(e, \text{cons}(e, e))) \) that we saw previously.

3. Which list is longer: \( \langle e, e, e \rangle \) or \( \langle e, \langle e, \langle e, e \rangle \rangle \rangle \)?

The first list is longer. It has three items; the second only two.
Axioms of Basic Lists

We take the following set of axioms for basic lists.

List 1: \( \forall x \forall y \text{cons}(x, y) \neq e. \)

List 2: \( \forall x \forall y \forall z \forall w (\text{cons}(x, y) = \text{cons}(z, w) \rightarrow (x = z \land y = w)). \)

List 3: For each formula \( \varphi(x) \) and each variable \( y \) not free in \( \varphi \),

\[
\varphi[e/x] \rightarrow \left( \forall x (\varphi \rightarrow \forall y \varphi[\text{cons}(y, x)/x]) \rightarrow \forall x \varphi \right)
\]

Note the close analogy to natural numbers!

- \( e \) corresponds to 0.
- \( \text{cons} \) is a binary analogue of \( s \).
- An induction axiom applies.

Exercise: Reasoning about lists

Exercise:

Prove that every non-\( e \) object is a \( \text{cons} \); that is, show that

\[
\vdash_{\text{List}} \forall x (x \neq e \rightarrow \exists y \exists z \text{cons}(y, z) = x).
\]

(Hint: the corresponding statement for natural numbers is that every non-zero number is a successor:

\[
\vdash_{\text{PA}} \forall x (x \neq 0 \rightarrow \exists z s(z) = x).
\]
Relations and Functions on Lists

Relations on Lists

We can define relations and functions on lists.

Example: the function first(·), where the first item on list x is first(x).

BUT: That’s not a function! The list e has no first.

Solution: use the relation

\[ R_{\text{first}} = \{ (\text{cons}(a, b), a) \mid \text{a and b are lists} \} \]

\( R_{\text{first}}(x, y) \) is definable by the formula \( \exists z (x = \text{cons}(y, z)) \).

Similarly, formula \( \exists z (x = \text{cons}(z, y)) \) defines

\[ R_{\text{rest}}(x, y) = \{ (\text{cons}(a, b), b) \mid \text{a and b are lists} \} \]

Properties of “First” and “Rest”

5. Lemma. Any object has at most one first:

\[ \forall x \forall y \forall z ((R_{\text{first}}(x, y) \land R_{\text{first}}(x, z)) \rightarrow y = z) \]

Sketch of proof: Assume \( R_{\text{first}}(x, y) \land R_{\text{first}}(x, z) \); that is,

\[ \exists u x = \text{cons}(y, u) \quad \text{and} \quad \exists u x = \text{cons}(z, u) \]

Thus there are items \( u_1 \) and \( u_2 \) (rule \( \exists e \)) for which

\[ x = \text{cons}(y, u_1) \quad \text{and} \quad x = \text{cons}(z, u_2) \]

Then List2 yields \( y = z \).
Length of Lists

Example: Constrain a relation EqLen so that EqLen(x, y) means that x and y have the same length.

- The empty list has the same length as itself: EqLen(e, e).
- A non-empty list does not have the same length as the empty list:
  \[ \forall x \left( x \neq e \rightarrow (\neg EqLen(x, e) \land \neg EqLen(e, x)) \right) . \]
- Adding an element to each of two lists preserves (in)equality of length:
  \[ \forall x \forall y \forall u \forall v \left( EqLen(x, y) \leftrightarrow EqLen(\text{cons}(u, x), \text{cons}(v, y)) \right) . \]

Length of Lists, continued

To show:

Every list has the same length as itself; that is, \( \forall x \, \text{EqLen}(x, x) \).

We prove this using induction, via List3.

Basis: \( \vdash_{\text{List}} \text{EqLen}(e, e) \). Given.

Inductive step: Need to prove, for arbitrary (“fresh”) x, that

\[ \text{EqLen}(x, x) \vdash_{\text{List}} \forall y \, \text{EqLen}(\text{cons}(y, x), \text{cons}(y, x)) . \]

This follows directly from the third formula for EqLen.

Exercise on Defined Relations

Similarly to EqLen, we can characterize UnEqLen, “unequal lengths”, by

\[ \neg \text{UnEqLen}(e, e) , \]

\[ \forall x \left( x \neq e \rightarrow (\neg \text{UnEqLen}(e, x) \land \neg \text{UnEqLen}(x, e)) \right) , \]

and

\[ \forall x \forall y \forall u \forall v \left( \text{UnEqLen}(x, y) \leftrightarrow \text{UnEqLen}(\text{cons}(u, x), \text{cons}(v, y)) \right) . \]

Exercise: Show that \( \vdash_{\text{List}} \forall x \forall y \left( \text{UnEqLen}(x, y) \leftrightarrow \neg \text{EqLen}(x, y) \right) . \)