Finite automata and formal languages (DIT323, TMV029)

Nils Anders Danielsson,
partly based on slides by Ana Bove

2024-01-22

Today

- Structural induction.
- Some concepts from automata theory.
- Inductively defined subsets
  (if we have time).

Structural induction

Structural induction

- For a given inductively defined set we have a corresponding induction principle.
- Example:

$$\frac{}{\textsf{zero} \in \mathbb{N}} \qquad\qquad \frac{n \in \mathbb{N}}{\textsf{suc}(n) \in \mathbb{N}}$$

In order to prove $\forall n \in \mathbb{N}.\ P(n)$:

- Prove $P(\textsf{zero})$.
- For all $n \in \mathbb{N}$, prove that $P(n)$ implies $P(\textsf{suc}(n))$.

Structural induction

- For a given inductively defined set we have a corresponding induction principle.
- Example:

$$\frac{}{\mathsf{true} \in Bool} \qquad \frac{}{\mathsf{false} \in Bool}$$

In order to prove $\forall b \in Bool.\ P(b)$:

- Prove $P(\mathsf{true})$.
- Prove $P(\mathsf{false})$.

Structural induction

- For a given inductively defined set we have a corresponding induction principle.
- Example:

$$\frac{}{\mathsf{nil} \in List(A)} \qquad \frac{x \in A \qquad xs \in List(A)}{\mathsf{cons}(x, xs) \in List(A)}$$

In order to prove $\forall xs \in List(A).\ P(xs)$:

- Prove $P(\mathsf{nil})$.
- For all $x \in A$ and $xs \in List(A)$, prove that $P(xs)$ implies $P(\mathsf{cons}(x, xs))$.

Pattern

- An inductively defined set:

$$\cdots \qquad \frac{x \in A \qquad \ldots \qquad d \in D(A)}{\mathsf{c}(x, \ldots, d) \in D(A)} \qquad \cdots$$

  Note that x is a non-recursive argument, and that d is recursive.
- In order to prove $\forall d \in D(A).\ P(d)$:
  - $\vdots$
  - For all $x \in A$, ..., $d \in D(A)$, prove that ... and $P(d)$ imply $P(\mathsf{c}(x, \ldots, d))$.
  - $\vdots$

  One inductive hypothesis for each *recursive* argument.

What is the induction principle for

$$\frac{n \in \mathbb{N}}{\mathsf{leaf}(n) \in \mathit{Tree}} \qquad \frac{l, r \in \mathit{Tree}}{\mathsf{node}(l, r) \in \mathit{Tree}}?$$

1. $(\forall n \in \mathbb{N}.\ P(\mathsf{leaf}(n))) \wedge$
   $(\forall l, r \in \mathit{Tree}.\ P(l) \wedge P(r) \Rightarrow P(\mathsf{node}(l, r)))$.
2. $(\forall n \in \mathbb{N}.\ P(\mathsf{leaf}(n))) \wedge$
   $(\forall l, r \in \mathit{Tree}.\ P(l) \wedge P(r) \Rightarrow P(\mathsf{node}(l, r))) \Rightarrow$
   $(\forall t \in \mathit{Tree}.\ P(t))$.
3. $(\forall n \in \mathbb{N}.\ P(\mathsf{leaf}(n))) \wedge$
   $(\forall t \in \mathit{Tree}.\ P(t) \Rightarrow P(\mathsf{node}(t, t))) \Rightarrow$
   $(\forall t \in \mathit{Tree}.\ P(t))$.

Respond at `https://pingo.coactum.de/729558`.

Some functions

Recall from last lecture:

$$
\begin{aligned}
& \mathit{length} \in \mathit{List}(A) \to \mathbb{N} \\
& \mathit{length}(\mathsf{nil}) && = \mathsf{zero} \\
& \mathit{length}(\mathsf{cons}(x, xs)) && = \mathsf{suc}(\mathit{length}(xs))
\end{aligned}
$$

$$
\begin{aligned}
& \mathit{append} \in \mathit{List}(A) \times \mathit{List}(A) \to \mathit{List}(A) \\
& \mathit{append}(\mathsf{nil}, \quad ys) && = ys \\
& \mathit{append}(\mathsf{cons}(x, xs), ys) && = \mathsf{cons}(x, \mathit{append}(xs, ys))
\end{aligned}
$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Let us prove the property

$$\begin{aligned} P(xs) := \forall ys \in List(A).& \\ & length(append(xs, ys)) = \\ & length(xs) + length(ys) \end{aligned}$$

by induction on the structure of the list.

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case nil:

$$length(append(\mathsf{nil}, ys))$$

$$length(\mathsf{nil}) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case nil:

$$length(append(\mathsf{nil}, ys)) =$$
$$length(ys)$$

$$length(\mathsf{nil}) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case nil:

$$
\begin{aligned}
& length(append(\mathsf{nil}, ys)) = \\
& length(ys) \\
& 0 + length(ys) = \\
& length(\mathsf{nil}) + length(ys)
\end{aligned}
$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case nil:

$$
\begin{aligned}
&length(append(\mathsf{nil}, ys)) = \\
&length(ys) = \\
&0 + length(ys) = \\
&length(\mathsf{nil}) + length(ys)
\end{aligned}
$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$length(append(\mathsf{cons}(x, xs), ys))$$

$$length(\mathsf{cons}(x, xs)) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$length(append(\mathsf{cons}(x, xs), ys)) =$$
$$length(\mathsf{cons}(x, append(xs, ys)))$$

$$length(\mathsf{cons}(x, xs)) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$length(append(\mathsf{cons}(x, xs), ys)) =$$
$$length(\mathsf{cons}(x, append(xs, ys))) =$$
$$1 + length(append(xs, ys))$$

$$length(\mathsf{cons}(x, xs)) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$length(append(\mathsf{cons}(x, xs), ys)) =$$
$$length(\mathsf{cons}(x, append(xs, ys))) =$$
$$1 + length(append(xs, ys))$$

$$(1 + length(xs)) + length(ys) =$$
$$length(\mathsf{cons}(x, xs)) + length(ys)$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$
\begin{aligned}
& length(append(\mathsf{cons}(x, xs), ys)) = \\
& length(\mathsf{cons}(x, append(xs, ys))) = \\
& 1 + length(append(xs, ys)) \\
& 1 + (length(xs) + length(ys)) = \\
& (1 + length(xs)) + length(ys) = \\
& length(\mathsf{cons}(x, xs)) + length(ys)
\end{aligned}
$$

Lemma

$\forall xs, ys \in List(A).$
$length(append(xs, ys)) = length(xs) + length(ys).$

Proof.

Case $\mathsf{cons}(x, xs)$:

$$
\begin{aligned}
& length(append(\mathsf{cons}(x, xs), ys)) = \\
& length(\mathsf{cons}(x, append(xs, ys))) = \\
& 1 + length(append(xs, ys)) = \{\text{By the IH, } P(xs).\} \\
& 1 + (length(xs) + length(ys)) = \\
& (1 + length(xs)) + length(ys) = \\
& length(\mathsf{cons}(x, xs)) + length(ys)
\end{aligned}
$$

Prove $\forall xs \in List(A).append(xs, \textsf{nil}) = xs$ and $\forall xs \in List(A).append(\textsf{nil}, xs) = xs$. Which proof is "easiest"?

1. The first.
2. The second.

Respond at `https://pingo.coactum.de/729558`.

Induction/recursion

- Inductively defined sets:
  inference rules with constructors.
- Recursion (primitive recursion):
  recursive calls only for recursive arguments
  $(f(\mathsf{c}(x,d)) = \ldots f(d) \ldots)$.
- Structural induction:
  inductive hypotheses for recursive arguments
  $(P(d) \Rightarrow P(\mathsf{c}(x,d)))$.

Some concepts from automata theory

Alphabets and strings

- An *alphabet* is a finite, nonempty set.
  - $\{ a, b, c, ..., z \}$.
  - $\{ 0, 1, ..., 9 \}$.
- A *string (or word) over the alphabet* Σ is a member of $List(\Sigma)$.

Some conventions

Following the course text book:

- Σ: An alphabet.
- a, b, c: Elements of alphabets.
- u, v, w: Words over an alphabet.

Notation

- Σ^* instead of $List(\Sigma)$.
- ε instead of nil or $[\,]$.
- aw instead of cons(a, w).
- a instead of cons(a, nil) or $[a]$.
- abc instead of $[a, b, c]$.
- uv instead of $append(u, v)$.
- $|w|$ instead of $length(w)$.
- Σ^+: Nonempty strings, $\{\, w \in \Sigma^* \mid w \neq \varepsilon \,\}$.

Exponentiation

- Σ^n: Strings of length n, $\{ w \in \Sigma^* \mid |w| = n \}$.
- An example: $\{ a, b \}^2 = \{ aa, ab, ba, bb \}$.
- Alternative definition of $\Sigma^n \subseteq \Sigma^*$:

$$\begin{aligned}
\Sigma^0 &= \{ \varepsilon \} \\
\Sigma^{n+1} &= \{ aw \mid a \in \Sigma, w \in \Sigma^n \}
\end{aligned}$$

Exponentiation

- w^n: w repeated n times.
- An example: $(ab)^3 = ababab$.
- A recursive definition:

$$\begin{aligned} w^0 &= \varepsilon \\ w^{n+1} &= ww^n \end{aligned}$$

Which of the following propositions are valid? The alphabet is $\{ a, b, c \}$.

1. $|uv| = |u| + |v|$.
2. $|uv| = |u||v|$.
3. $|w^n| = n$.
4. $uv = vu$.
5. $\varepsilon v = v\varepsilon$.

Respond at `https://pingo.coactum.de/729558`.

Languages

A *language* over an alphabet Σ is a set $L \subseteq \Sigma^*$.

- Typical programming languages.
- Typical natural languages?
  (Are they well-defined?)
- Other examples, for instance the odd natural numbers expressed in binary notation (without leading zeros), which is a language over $\{0, 1\}$.

Another convention

Following the course text book:

- L, M, N: Languages.

Operations

- Concatenation: $LM = \{ uv \mid u \in L, v \in M \}$.
- An example:

$$\{ a, bc \} \{ de, f \} = \{ ade, af, bcde, bcf \}$$

Operations

- Exponentiation:

$$\begin{aligned} L^0 &= \{\, \varepsilon \,\} \\ L^{n+1} &= LL^n \end{aligned}$$

- An example:

$$\begin{aligned} &\{\, a, bc \,\}^2 = \\ &\{\, a, bc \,\}(\{\, a, bc \,\}^1) = \\ &\{\, a, bc \,\}(\{\, a, bc \,\}\{\, \varepsilon \,\}) = \\ &\{\, a, bc \,\}\{\, a, bc \,\} = \\ &\{\, aa, abc, bca, bcbc \,\} \end{aligned}$$

Operations

- Exponentiation:

$$\begin{aligned} L^0 &= \{\, \varepsilon \,\} \\ L^{n+1} &= LL^n \end{aligned}$$

- This definition is consistent with a previous one:

$$\Sigma^n = \{\, w \in \Sigma^* \mid |w| = 1 \,\}^n$$

Operations

- The Kleene star $L^* = \bigcup_{n \in \mathbb{N}} L^n$.
- An example:

$$\{\, a, bc \,\}^* =$$
$$\{\, a, bc \,\}^0 \cup \{\, a, bc \,\}^1 \cup \{\, a, bc \,\}^2 \cup \ldots =$$
$$\{\, \varepsilon, a, bc, aa, abc, bca, bcbc, \ldots \,\}$$

- This definition is consistent with a previous one:

$$\Sigma^* = \{\, w \in \Sigma^* \mid |w| = 1 \,\}^*$$

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

1. $\forall w \in L^n.\ |w| = n$.
2. $LM = ML$.
3. $L(M \cup N) = LM \cup LN$.
4. $LM \cap LN \subseteq L(M \cap N)$.
5. $L^* L^* \subseteq L^*$.

Respond at `https://pingo.coactum.de/729558`.

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

1. $\forall w \in L^n.\ |w| = n$.

No. Counterexample: $L = \{ \varepsilon \}$, $n = 1$.

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

2. $LM = ML$.

No. Counterexample: $L = \{ 0 \}$, $M = \{ 1 \}$.

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

3. $L(M \cup N) = LM \cup LN$.

Yes. The set $L(M \cup N)$ consists exactly of the strings in LM and the strings in LN.

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

4. $LM \cap LN \subseteq L(M \cap N)$.

No. With $L = \{ \varepsilon, 1 \}$, $M = \{ 1 \}$ and $N = \{ \varepsilon \}$ we get that

$$\begin{aligned}
& LM \cap LN & = \\
& \{ 1, 11 \} \cap \{ \varepsilon, 1 \} & = \\
& \{ 1 \} & \nsubseteq \\
& \emptyset & = \\
& L\emptyset & = \\
& L(M \cap N).
\end{aligned}$$

Which of the following propositions are valid? The alphabet is $\{ 0, 1, 2 \}$.

5. $L^*L^* \subseteq L^*$.

Yes. Any string in L^*L^* consists of

- a string in L^* followed by a string in L^*,
- i.e. m strings in L followed by n strings in L (for some $m, n \in \mathbb{N}$),
- i.e. $m + n$ strings in L,
- and such a string is a member of L^*.

In fact, $(L^*)^* = L^*$.

Inductively defined subsets

Inductively defined subsets

- One can define subsets of (say) Σ^* inductively.
- For instance, for $L \subseteq \Sigma^*$ we can define $L^* \subseteq \Sigma^*$ inductively:

$$\frac{}{\varepsilon \in L^*} \qquad \frac{u \in L \qquad v \in L^*}{uv \in L^*}$$

- Note that there are no constructors (but in some cases it might make sense to name the rules).

$$\frac{}{\varepsilon \in L^*} \qquad \frac{u \in L \qquad v \in L^*}{uv \in L^*}$$

$aba \in \{ a, ab \}^*$

Proof:

$$\frac{\dfrac{}{ab \in \{ a, ab \}} \qquad \dfrac{\dfrac{}{a \in \{ a, ab \}} \qquad \dfrac{}{\varepsilon \in \{ a, ab \}^*}}{a \in \{ a, ab \}^*}}{aba \in \{ a, ab \}^*}$$

$bab \notin \{ a, ab \}^*$

Proof:

- Because $bab \neq \varepsilon$ a derivation of $bab \in \{ a, ab \}^*$ would have to end in the following way, with $uv = bab$:

$$\frac{u \in \{ a, ab \} \qquad v \in \{ a, ab \}^*}{uv \in \{ a, ab \}^*}$$

- Because $u \in \{ a, ab \}$ we get that $u = a$ or $u = ab$.
- In either case we get a contradiction, because u must be empty or start with b.

Inductively defined subsets

- What about recursion?

$$
\begin{aligned}
&f \in L^* \to Bool\\
&f(\varepsilon) \quad = \mathsf{false}\\
&f(uv) = not(f(v))
\end{aligned}
$$

- If $\varepsilon \in L$, do we have

$$f(\varepsilon) = f(\varepsilon\varepsilon) = not(f(\varepsilon))\text{?}$$

Inductively defined subsets

- Induction works
  (assuming "proof irrelevance").
- $P(\varepsilon) \wedge (\forall u \in L, v \in L^*.\ P(v) \Rightarrow P(uv)) \Rightarrow \forall w \in L^*.\ P(w)$.

Another example

$L \subseteq \{ a, b \}^*$ is defined inductively in the following way:

$$\frac{}{a \in L} \qquad \frac{u, v \in L}{ubv \in L}$$

An induction principle for L:

$$P(a) \wedge (\forall u, v \in L.\ P(u) \wedge P(v) \Rightarrow P(ubv)) \Rightarrow \forall w \in L.\ P(w)$$

$L \subseteq \{ a, b \}^*$ is defined inductively in the following way:

$$\frac{}{a \in L} \qquad \frac{u, v \in L}{ubv \in L}$$

Which of the following propositions are valid?

1. $\varepsilon \in L$.
2. $aba \in L$.
3. $bab \in L$.
4. $aabaa \in L$.
5. $ababa \in L$.

Respond at `https://pingo.coactum.de/729558`.

Today

- Structural induction.
- Some concepts from automata theory.
- Inductively defined subsets.

Next lecture

- Deterministic finite automata.