

Event B: Modeling and Reasoning with Data Structures¹

Manuel Carro

manuel.carro@upm.es

Universidad Politécnica de Madrid &
IMDEA Software Institute

¹With material from J. R. Abrial book *Modeling in Event-B: system and software engineering*.

Infinite Lists	s. 4
Finite Lists	s. 15
Infinite Trees	s. 16
Finite Trees	s. 17

- Data structures involving pointers / references described with relations, functions.
- Specific axioms of these specific data structures give *properties* of the functions that model the data structures.
- These properties are necessary for theorem provers to discharge proofs on data structures.
- Specific forms of these axioms (capturing induction on the data structures) are well-suited to be used in automated proofs.
- We will focus on formalizing:
 - Infinite lists.
 - Finite lists.
 - Infinite trees.
 - Finite trees.

- Set V of list nodes.
- Initial node f .
- Bijective *next* function.

There is a *next* and a *previous* (with the exception of f)

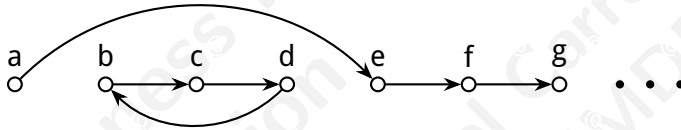
$$\text{axm_1 : } f \in V$$

$$\text{axm_2 : } n \in V \mapsto V \setminus \{f\}$$

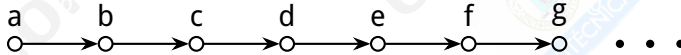


Characterizing (and avoiding) cycles

Cycles:

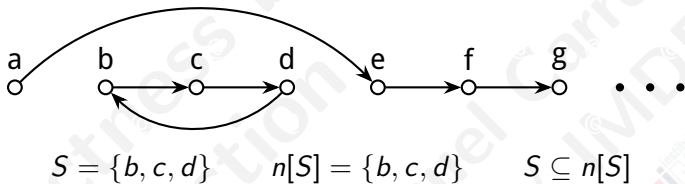


No cycles:

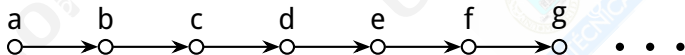


Characterizing (and avoiding) cycles

Cycles:

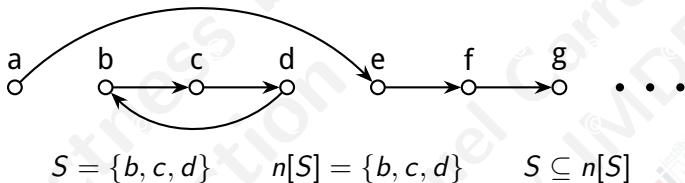


No cycles:

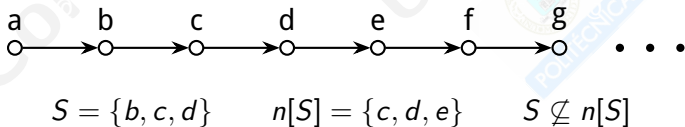


Characterizing (and avoiding) cycles

Cycles:



No cycles:



(for *almost* any $S \subseteq V$)

- If a list has a cycle, then there is a $S \subseteq V$ s.t. $S \subseteq n[S]$.
- On the other hand, it is always the case that $\emptyset \subseteq n[\emptyset]$.
- So we insist that this is the only case:

$$\text{axm_3} : \forall S \cdot S \subseteq V \wedge S \subseteq n[S] \Rightarrow S = \emptyset$$

- It can be used to prove properties in infinite lists.
- We will derive from it an axiom scheme of induction.

- Absence of cycles: $\forall S \cdot S \subseteq V \wedge S \subseteq n[S] \Rightarrow S = \emptyset$
- S can be written as $S = V \setminus T$, for some T
(For example, $T = V \setminus S$ would work)
- Then:

$$\forall S \cdot S = V \setminus T \wedge \boxed{S \subseteq V} \wedge S \subseteq n[S] \Rightarrow S = \emptyset$$

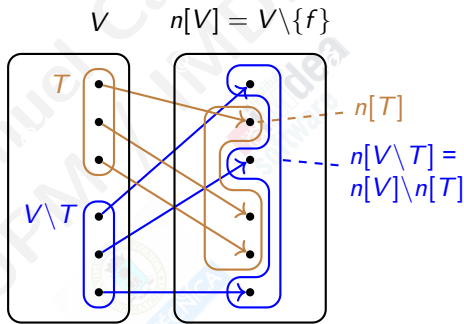
↑
Redundant

- Removing redundant subformula: $\forall S \cdot S = V \setminus T \wedge S \subseteq n[S] \Rightarrow S = \emptyset$
- Let us focus on $S = \emptyset$

From absence of cycles to induction

Let us simplify $\forall S \cdot S = V \setminus T \wedge S \subseteq n[S] \Rightarrow S = \emptyset$

- If $S = V \setminus T$, then
 $S = \emptyset \equiv V \setminus T = \emptyset \equiv V \subseteq T$
- The non-cycle condition then becomes
 $\forall S \cdot S = V \setminus T \wedge S \subseteq n[S] \Rightarrow V \subseteq T$
- Let us focus on $n[S]$
- Since $S = V \setminus T$, $n[S] = n[V \setminus T]$
- Since n is bijective, $n[V \setminus T]$ and $n[T]$ don't intersect (see figure on the right)
- Therefore, $n[V \setminus T] = n[V] \setminus n[T]$



- Since $S = V \setminus T$ and $n[V \setminus T] = n[V] \setminus n[T]$, $S \subseteq n[S]$ becomes $V \setminus T \subseteq n[V] \setminus n[T]$
- Let us simplify that condition
- By definition: $f \in V$ and $f \notin n[V \setminus T]$ (f is not in the range of n)
- Since $V \setminus T \subseteq n[V \setminus T]$, $f \notin V \setminus T$
(because $f \notin n[V \setminus T]$ and $V \setminus T$ is a subset of $n[V \setminus T]$)
- Therefore f must be *subtracted* from V by T , and then $f \in T$
- Also by definition, $n[V] = V \setminus \{f\}$.
- So we can rewrite $V \setminus T \subseteq n[V] \setminus n[T]$ as $V \setminus T \subseteq (V \setminus \{f\}) \setminus n[T]$

• Let us simplify $\underbrace{\overbrace{V}^a \setminus \overbrace{T}^b}_e \subseteq \underbrace{\overbrace{(V \setminus \{f\})}^c \setminus \overbrace{n[T]}^d}_f$.

- We know that $f \in V$ and $f \in T$.
- f is not in set (f) , and then it should not be in (e) ; it is removed by (b) .
- Then we have to worry about how much is removed by (b) and (d) .
- If (d) removes “too much”, then (e) will be larger.
- I.e., if (d) contains an element that is not in (b) , then (e) will contain an element that is not in (f) .
- Therefore, (d) cannot contain elements that are not in (b) (or, any element in (d) must also be in (b))
- So the formula simplifies to $n[T] \subseteq T$.

Putting all together, the non-cycle condition becomes

$$\forall S \cdot S = V \setminus T \wedge f \in T \wedge n[T] \subseteq T \Rightarrow V \subseteq T$$

If we expand $n[T] \subseteq T$:

$$\text{thm_2} : \forall T \cdot f \in T \wedge (\forall x \cdot x \in T \Rightarrow n(x) \in T) \Rightarrow V \subseteq T$$

- T the set of elements with some property P : $T = \{x | P(x)\}$
- So the meaning of thm_2 is:
 - If the initial node f has property P ($f \in T$), and
 - For every element with property P ($x \in T$), the next one has this property ($n(x) \in T$), then
 - All elements have property P ($V \subseteq T$).

Using thm_2 to prove list properties

- We want to prove $P(x)$ for all $x \in V$.
- Elements for which P holds:
 $T = \{x | x \in V \wedge P(x)\}.$
- We want to prove that $T = V$.
- Since by construction $T \subseteq V$, it is enough to prove $V \subseteq T$.
- We do that by instantiating T in thm_2.

$$\begin{aligned} f &\in \{x | x \in V \wedge P(x)\} && \wedge \\ \forall x. x \in \{x | x \in V \wedge P(x)\} \Rightarrow n(x) \in \{x | x \in V \wedge P(x)\} && \Rightarrow \\ V &\subseteq \{x | x \in V \wedge P(x)\} \end{aligned}$$

- $f \in \{x | x \in V \wedge P(x)\} \equiv P(f).$
- Second part equivalent to
 $\forall x. x \in V \wedge P(x) \Rightarrow P(n(x)).$
- The RHS is equivalent to
 $\forall x. x \in V \Rightarrow P(x).$
- Instantiating thm_2 gives a scheme to prove by induction in infinite lists.

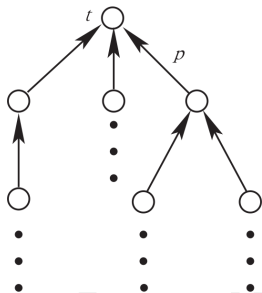
- Basically as infinite lists, but including a last (l) element and a different axiom 2:

axm_4 : $l \in V$

axm_5 : $\text{finite}(V)$

axm_2' : $n \in V \setminus \{l\} \mapsto V \setminus \{f\}$

induction : $\forall T \cdot T \subseteq V \wedge f \in T \wedge (\forall x \cdot x \in V \setminus \{l\} \wedge x \in T \Rightarrow n(x) \in T) \Rightarrow V \subseteq T$



- t is the root.
- p links node with parent (surjection).
- No cycles.

axm_1 : $t \in V$

axm_2 : $p \in V \setminus \{t\} \rightarrow V$

axm_3 : $\forall S \cdot S \subseteq p^{-1}[S] \Rightarrow S = \emptyset$

Induction rule:

$$\forall T \cdot t \in T \wedge p^{-1}[T] \subseteq T \Rightarrow V \subseteq T$$

Instantiation to prove properties:

$$\begin{aligned}
 \forall T \cdot \quad & T \subseteq V \wedge t \in T \wedge \\
 & (\forall x \cdot x \in V \setminus \{t\} \wedge p(x) \in T \Rightarrow x \in T) \\
 \Rightarrow & V \subseteq T
 \end{aligned}$$

Note: placement of p in implication is *opposite* w.r.r. f for lists – “direction” of arrows reversed!

- t is the root.
- p relates every node with its parent.
- L is the set of tree leaves.
- There should not be cycles.

$$\text{axm_1 : } t \in V$$

$$\text{axm_2 : } L \subseteq V$$

$$\text{axm_3 : } p \in V \setminus \{t\} \rightarrow V \setminus L$$

$$\text{axm_4 : } \forall S \cdot S \subseteq p^{-1}[S] \Rightarrow S = \emptyset$$

The induction scheme is as in infinite trees.