Sets
Overview

- Set notation
- Inductively defined sets
Set notation
Type \( \text{'a set} \): sets over type \( \text{'a} \)
Type 'a set: sets over type 'a

• {}, {e_1, . . . , e_n}, {x. P x}
Sets

Type 'a set: sets over type 'a

- {},
- \{e_1, \ldots, e_n\},
- \{x. P x\}
- e \in A, \quad A \subseteq B
Sets

Type `a set`: sets over type `a`

- `{}`, `{e_1, ..., e_n}`, `{x. P x}`
- `e ∈ A, A ⊆ B`
- `A ∪ B, A ∩ B, A - B, - A`
Sets

Type 'a set: sets over type 'a

- \{\}, \{e_1, \ldots, e_n\}, \{x. \ P \ x\}
- e \in A, \ A \subseteq B
- A \cup B, \ A \cap B, \ A - B, \ - A
- \bigcup_{x \in A} B \ x, \ \bigcap_{x \in A} B \ x
Sets

Type 'a set: sets over type 'a

- \{\}, \{e_1, \ldots, e_n\}, \{x. P x\}
- e \in A, \ A \subseteq B
- A \cup B, \ A \cap B, \ A - B, \ - A
- \bigcup_{x \in A} B x, \ \bigcap_{x \in A} B x
- \{i..j\}
Sets

Type 'a set: sets over type 'a

- \{\}, \{e_1, \ldots, e_n\}, \{x. P \ x\}
- e \in A, \ A \subseteq B
- A \cup B, \ A \cap B, \ A - B, \ -A
- \bigcup_{x \in A} B \ x, \ \bigcap_{x \in A} B \ x
- \{i..j\}
- \text{insert} :: 'a \Rightarrow 'a \text{ set} \Rightarrow 'a \text{ set}
Sets

Type ’a set: sets over type ’a

- \{\}, \{e_1,\ldots,e_n\}, \{x. P x\}
- e \in A, A \subseteq B
- A \cup B, A \cap B, A - B, -A
- \bigcup_{x \in A} B x, \bigcap_{x \in A} B x
- \{i..j\}
- \text{insert :: } ’a \Rightarrow ’a set \Rightarrow ’a set
- f ‘ A \equiv \{y. \exists x \in A. y = f x\}
Sets

Type ’a set: sets over type ’a

• \{\}, \{e_1, \ldots, e_n\}, \{x. P x\}
• e \in A, A \subseteq B
• A \cup B, A \cap B, A - B, - A
• \bigcup_{x \in A} B x, \bigcap_{x \in A} B x
• \{i..j\}
• \text{insert} :: ’a \Rightarrow ’a set \Rightarrow ’a set
• f ‘ A \equiv \{y. \exists x \in A. y = f x\}
• \ldots
Proofs about sets

Natural deduction proofs:

• \textit{equalityI: } \[A \subseteq B; B \subseteq A \implies A = B\]
Proofs about sets

Natural deduction proofs:

- **equalityI**: \([A \subseteq B; B \subseteq A] \implies A = B\)
- **subsetI**: \((\forall x. x \in A \implies x \in B) \implies A \subseteq B\)
Proofs about sets

Natural deduction proofs:

• equalityI: \([A \subseteq B; B \subseteq A] \implies A = B\)
• subsetI: \((\forall x. x \in A \implies x \in B) \implies A \subseteq B\)
• ... (see Tutorial)
Demo: proofs about sets
Bounded quantifiers

• $\forall x \in A. \ P x$
Bounded quantifiers

• \( \forall x \in A. \ P x \equiv \ \forall x. \ x \in A \rightarrow P x \)
Bounded quantifiers

- $\forall x \in A. \ P x \equiv \forall x. \ x \in A \rightarrow P x$
- $\exists x \in A. \ P x$
Bounded quantifiers

- $\forall x \in A. P x \equiv \forall x. x \in A \rightarrow P x$
- $\exists x \in A. P x \equiv \exists x. x \in A \land P x$
**Bounded quantifiers**

- $\forall x \in A. \ P x \equiv \forall x. \ x \in A \rightarrow P x$
- $\exists x \in A. \ P x \equiv \exists x. \ x \in A \land P x$
- $\text{ballI}: (\forall x. \ x \in A \rightarrow P x) \rightarrow \forall x \in A. \ P x$
- $\text{bspec}: [\forall x \in A. \ P x; \ x \in A] \rightarrow P x$
Bounded quantifiers

- $\forall x \in A. \ P x \equiv \forall x. \ x \in A \rightarrow P x$
- $\exists x \in A. \ P x \equiv \exists x. \ x \in A \land P x$
- $\text{ballI}: (\forall x. \ x \in A \rightarrow P x) \rightarrow \forall x \in A. \ P x$
- $\text{bspec}: [\forall x \in A. \ P x; x \in A] \rightarrow P x$
- $\text{bexI}: [P x; x \in A] \rightarrow \exists x \in A. \ P x$
- $\text{bexE}: [\exists x \in A. \ P x; \forall x. \ [x \in A; P x] \rightarrow Q] \rightarrow Q$
Inductively defined sets
Example: even numbers

Informally:
Example: even numbers

Informally:

- 0 is even
Example: even numbers

Informally:

- 0 is even
- If \( n \) is even, so is \( n + 2 \)
Example: even numbers

Informally:

• 0 is even
• If $n$ is even, so is $n + 2$
• These are the only even numbers
Example: even numbers

Informally:

• 0 is even
• If $n$ is even, so is $n + 2$
• These are the only even numbers

In Isabelle/HOL:

```isar
inductive_set Ev :: nat set  — The set of all even numbers
```
Example: even numbers

Informally:

• 0 is even
• If \( n \) is even, so is \( n + 2 \)
• These are the only even numbers

In Isabelle/HOL:

```isabelle
inductive_set Ev :: nat set  
where
  0 ∈ Ev | 
  n ∈ Ev ⟹ n + 2 ∈ Ev
```

— The set of all even numbers
Format of inductive definitions

\textbf{inductive\_set} \quad S :: \tau \ \textit{set}
Format of inductive definitions

inductive_set S :: τ set
where
\[ a_1 \in S; \ldots ; a_n \in S; A_1; \ldots ; A_k \] \Rightarrow a \in S /

\vdots
Format of inductive definitions

inductive_set S :: \( \tau \) set
where
\[
[ a_1 \in S; \ldots ; a_n \in S; A_1; \ldots ; A_k ] \implies a \in S
\]
where \( A_1; \ldots ; A_k \) are side conditions not involving \( S \).
Proving properties of even numbers

Easy: \( 4 \in \text{Ev} \)

\[ 0 \in \text{Ev} \implies 2 \in \text{Ev} \implies 4 \in \text{Ev} \]
Proving properties of even numbers

Easy: $4 \in \text{Ev}$

$$0 \in \text{Ev} \implies 2 \in \text{Ev} \implies 4 \in \text{Ev}$$

Trickier: $m \in \text{Ev} \implies m+m \in \text{Ev}$
Proving properties of even numbers

Easy: $4 \in Ev$

\[ 0 \in Ev \implies 2 \in Ev \implies 4 \in Ev \]

Trickier: $m \in Ev \implies m+m \in Ev$

Idea: induction on the length of the derivation of $m \in Ev$
Proving properties of even numbers

Easy: \( 4 \in Ev \)

\[
0 \in Ev \implies 2 \in Ev \implies 4 \in Ev
\]

Trickier: \( m \in Ev \implies m+m \in Ev \)

Idea: induction on the length of the derivation of \( m \in Ev \)
Better: induction on the \textit{structure} of the derivation
Proving properties of even numbers

Easy: \(4 \in Ev\)
\[
0 \in Ev \implies 2 \in Ev \implies 4 \in Ev
\]

Trickier: \(m \in Ev \implies m+m \in Ev\)

Idea: induction on the length of the derivation of \(m \in Ev\)

Better: induction on the structure of the derivation

Two cases: \(m \in Ev\) is proved by
- rule \(0 \in Ev\)
Proving properties of even numbers

Easy: \(4 \in Ev\)

\[
0 \in Ev \implies 2 \in Ev \implies 4 \in Ev
\]

Trickier: \(m \in Ev \implies m+m \in Ev\)

Idea: induction on the length of the derivation of \(m \in Ev\)

Better: induction on the structure of the derivation

Two cases: \(m \in Ev\) is proved by

- rule \(0 \in Ev\)
  \[
  \implies m = 0 \implies 0+0 \in Ev
  \]
Proving properties of even numbers

Easy: $4 \in Ev$

\[ 0 \in Ev \implies 2 \in Ev \implies 4 \in Ev \]

Trickier: $m \in Ev \implies m+m \in Ev$

Idea: induction on the length of the derivation of $m \in Ev$

Better: induction on the structure of the derivation

Two cases: $m \in Ev$ is proved by

- rule $0 \in Ev$
  \[ \implies m = 0 \implies 0+0 \in Ev \]

- rule $n \in Ev \implies n+2 \in Ev$
Proving properties of even numbers

Easy: \( 4 \in Ev \)

\[
0 \in Ev \implies 2 \in Ev \implies 4 \in Ev
\]

Trickier: \( m \in Ev \implies m+m \in Ev \)

Idea: induction on the length of the derivation of \( m \in Ev \)

Better: induction on the structure of the derivation

Two cases: \( m \in Ev \) is proved by

- rule \( 0 \in Ev \)
  \[
  \implies m = 0 \implies 0+0 \in Ev
  \]

- rule \( n \in Ev \implies n+2 \in Ev \)
  \[
  \implies m = n+2 \text{ and } n+n \in Ev \text{ (ind. hyp.!)}
  \]
Proving properties of even numbers

Easy: \( 4 \in \text{Ev} \)

\[
0 \in \text{Ev} \implies 2 \in \text{Ev} \implies 4 \in \text{Ev}
\]

Trickier: \( m \in \text{Ev} \implies m+m \in \text{Ev} \)

Idea: induction on the length of the derivation of \( m \in \text{Ev} \)

Better: induction on the structure of the derivation

Two cases: \( m \in \text{Ev} \) is proved by

- rule \( 0 \in \text{Ev} \)
  \[
  \implies m = 0 \implies 0+0 \in \text{Ev}
  \]

- rule \( n \in \text{Ev} \implies n+2 \in \text{Ev} \)
  \[
  \implies m = n+2 \text{ and } n+n \in \text{Ev} \text{ (ind. hyp.!)}
  \implies m+m = (n+2)+(n+2) = ((n+n)+2)+2 \in \text{Ev}
  \]
Rule induction for Ev

To prove

\[ n \in Ev \implies P n \]

by *rule induction* on \( n \in Ev \) we must prove
Rule induction for $Ev$

To prove

\[ n \in Ev \implies P\ n \]

by rule induction on $n \in Ev$ we must prove

- $P\ 0$
Rule induction for Ev

To prove

\[ n \in Ev \implies P \; n \]

by rule induction on \( n \in Ev \) we must prove

- \( P \; 0 \)
- \( P \; n \implies P(n+2) \)
**Rule induction for Ev**

To prove

\[ n \in Ev \implies P n \]

by *rule induction* on \( n \in Ev \) we must prove

- \( P 0 \)
- \( P n \implies P(n+2) \)

**Rule** \( \text{Ev} \text{.induct} \):

\[
\left[ n \in Ev; P 0; \bigwedge n. P n \implies P(n+2) \right] \implies P n
\]
Rule induction for $Ev$

To prove

$$n \in Ev \implies P\ n$$

by rule induction on $n \in Ev$ we must prove

- $P\ 0$
- $P\ n \implies P(n+2)$

Rule $Ev\ .\ induct$:

\[
\left[ \begin{array}{c}
 n \in Ev; P\ 0; \bigwedge n. P\ n \implies P(n+2)
\end{array} \right] \implies P\ n
\]

An elimination rule
Set $S$ is defined inductively.
Rule induction in general

Set $S$ is defined inductively. To prove

$x \in S \implies P(x)$

by rule induction on $x \in S$
**Rule induction in general**

Set $S$ is defined inductively. To prove

$$x \in S \implies P(x)$$

by *rule induction* on $x \in S$ we must prove for every rule

$$[ a_1 \in S; \ldots ; a_n \in S ] \implies a \in S$$

that $P$ is preserved:
Set $S$ is defined inductively.
To prove

$$x \in S \implies P \ x$$

by rule induction on $x \in S$
we must prove for every rule

$$[ a_1 \in S; \ldots ; a_n \in S ] \implies a \in S$$

that $P$ is preserved:

$$[ P \ a_1; \ldots ; P \ a_n ] \implies P \ a$$
Rule induction in general

Set $S$ is defined inductively. To prove

$$x \in S \implies P(x)$$

by rule induction on $x \in S$ we must prove for every rule

$$[a_1 \in S; \ldots ; a_n \in S] \implies a \in S$$

that $P$ is preserved:

$$[P(a_1); \ldots ; P(a_n)] \implies P(a)$$

In Isabelle/HOL:

apply(erule $S.induct$)
Demo: inductively defined sets
Inductive predicates

\( \tau \text{ set} \leadsto \tau \Rightarrow \text{bool} \)
Inductive predicates

\[ \tau \text{ set } \rightsquigarrow \tau \Rightarrow \text{bool} \]

Example:

inductive \( Ev :: \text{nat} \Rightarrow \text{bool} \)

where

\[ Ev \ 0 \ | \]
\[ Ev \ n \ \Rightarrow \ Ev \ (n + 2) \]
**Inductive predicates**

\[ \tau \text{ set} \iff \tau \Rightarrow \text{bool} \]

Example:

```plaintext
inductive Ev :: nat \Rightarrow \text{bool}
where
    Ev 0 | Ev n \Rightarrow Ev (n + 2)
```

Comparison:

- **predicate** simpler syntax
- **set** direct usage of \( \cup \) etc
**Inductive predicates**

\[ \tau \text{ set} \leadsto \tau \Rightarrow \text{bool} \]

Example:

- **inductive** \(Ev :: \text{nat} \Rightarrow \text{bool} \)

where

\[
\begin{align*}
Ev \ 0 & \mid \\
Ev \ n & \Rightarrow Ev \ (n + 2)
\end{align*}
\]

Comparison:

- **predicate** simpler syntax
- **set** direct usage of \(\cup\) etc

Inductive predicates can be of type \(\tau_1 \Rightarrow \ldots \Rightarrow \tau_n \Rightarrow \text{bool}\)