



02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54–62		
Foil 5.1		
Anne Haxthausen, IMM/DTU	Spring 2002	

## Sets

### Contents

- what is a set? 2
- set type expressions 3
- set operators 3
- set value expressions 6
- examples of modeling using sets 7


02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54–62		
Foil 5.2		
Anne Haxthausen, IMM/DTU	Spring 2002	

## What is a set?

A set is: an unordered collection of distinct values of the same type.

Examples:

```
{1,3,5}
{"John","Peter","Ann"}
```

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54–62		
Foil 5.3		
Anne Haxthausen, IMM/DTU	Spring 2002	

## Set Type Expressions

- `type_expr-set` denotes the type consisting of all finite sets

$$\{v_1, \dots, v_n\}$$

where  $n \geq 0$ ,  $v_i : \text{type\_expr}$

- `type_expr-infset` denotes the type consisting of all sets

$$\{v_1, \dots, v_n\},$$


$$\{v_1, \dots, v_n, \dots\}$$

where  $n \geq 0$ ,  $v_i : \text{type\_expr}$

### Operators

$\cup, \cap, \setminus, \in, \notin, \subset, \subseteq, \supset, \supseteq$ , **card**

**card**  $\{n \mid n : \text{Nat}\} \equiv \text{chaos}$

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54–62		
Foil 5.4		
Anne Haxthausen, IMM/DTU	Spring 2002	

## Set Type Expressions

### Examples

#### **Bool-set**


denotes the type consisting of the sets:

```
{ }
{true}
{false}
{true,false}
```

#### **Nat-set**

denotes the type consisting of the sets:

```
{ }
{0}
{1}
...
{0,1}
...
{1,2,3}
...
```

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54-62		
Foil 5.5		
Anne Haxthausen, IMM/DTU	Spring 2002	


### Examples

**Bool-infset = Bool-set**

**Nat-infset**

denotes the type consisting of the sets:

$\{\}$   
 $\{0\}$   
 $\{1\}$   
 $\dots$   
 $\{0,1\}$   
 $\dots$   
 $\{1,2,3\}$   
 $\dots$   
 $\{0,1,2,3,4,\dots\}$   
 $\{2,3,5,7,\dots\}$   
 $\dots$

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54-62		
Foil 5.6		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Value Expressions

Enumerated:

$\{1,2\}$   
 $\{1,2,1\}$

$\{\text{value\_expr}_1, \dots, \text{value\_expr}_n\}$

Ranged:


$\{3 .. 7\} = \{3,4,5,6,7\}$   
 $\{3 .. 3\} = \{3\}$   
 $\{3 .. 2\} = \{\}$

$\{\text{value\_expr}_1 .. \text{value\_expr}_2\}$

Comprehended:

$\{2*n \mid n : \mathbf{Nat} \cdot n \leq 3\}$

$\{\text{value\_expr}_1 \mid \text{typing}_1, \dots, \text{typing}_n \cdot \text{value\_expr}_2\}$

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54-62		
Foil 5.7		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Example I

**scheme RESOURCE\_MANAGER =**  
**class**

**type**

Resource,  
Pool = Resource-**set**


**value**

obtain : Pool  $\rightarrow$  Pool  $\times$  Resource  
 obtain(p) **as** (p1,r1) **post** r1  $\in$  p  $\wedge$  p1 = p \ {r1}  
**pre** p  $\neq$  {},

release : Resource  $\times$  Pool  $\rightarrow$  Pool  
 release(r,p)  $\equiv$  p  $\cup$  {r}

**pre** r  $\notin$  p

**end**

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets; ch.8, pp.54-62		
Foil 5.8		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Example II

**scheme EQUIVALENCE\_RELATION =**  
**class**

**type**

Elem,  
Class = Elem-**infset**,  
Relation = Class-**infset**

**value**

is\_wf\_Relation : Relation  $\rightarrow$  **Bool**

is\_wf\_Relation(r)  $\equiv$

$\{\} \notin r \wedge$

$(\forall e : \text{Elem} \cdot \exists c : \text{Class} \cdot c \in r \wedge e \in c) \wedge$

$(\forall c1, c2 : \text{Class} \cdot$

$c1 \in r \wedge c2 \in r \wedge c1 \neq c2 \Rightarrow c1 \cap c2 = \{\}),$

initial : Relation =  $\{\{e\} \mid e : \text{Elem}\},$

make\_equivalent : Elem  $\times$  Elem  $\times$  Relation  $\rightarrow$  Relation

make\_equivalent(e1,e2,r)  $\equiv$

$\{c \mid c : \text{Class} \cdot c \in r \wedge \{e1,e2\} \cap c = \{\} \} \cup$

$\{c1 \cup c2 \mid c1, c2 : \text{Class} \cdot$


$c1 \in r \wedge c2 \in r \wedge e1 \in c1 \wedge e2 \in c2\},$

are\_equivalent : Elem  $\times$  Elem  $\times$  Relation  $\rightarrow$  **Bool**

are\_equivalent(e1,e2,r)  $\equiv$

$(\exists c : \text{Class} \cdot c \in r \wedge e1 \in c \wedge e2 \in c)$


**end**

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets: ch.8, pp.54-62		
Foil 5.9		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Example II — continued

Are the following properties true for EQUIVALENCE\_RELATION?

- (1)  $\text{is\_wf\_Relation}(\text{initial})$ ,
- (2)  $\forall e_1, e_2 : \text{Elem}, r : \text{Relation} \cdot$   
 $\text{is\_wf\_Relation}(r) \Rightarrow$   
 $\text{is\_wf\_Relation}(\text{make\_equivalent}(e_1, e_2, r))$

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets: ch.8, pp.54-62		
Foil 5.10		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Example III

**scheme** SET\_DATABASE =

**class**

**type**

Record = Key  $\times$  Data,

Database = { rs : Record-set  $\cdot$  is\_wf\_Database(rs) },

Key, Data

**value**

is\_wf\_Database : Record-set  $\rightarrow$  **Bool**

is\_wf\_Database(rs)  $\equiv$

$(\forall k : \text{Key}, d1, d2 : \text{Data} \cdot$

$((k, d1) \in rs \wedge (k, d2) \in rs) \Rightarrow d1 = d2)$ ,

empty : Database = { },

insert : Key  $\times$  Data  $\times$  Database  $\rightarrow$  Database

insert(k, d, db)  $\equiv$  remove(k, db)  $\cup$  { (k, d) },

remove : Key  $\times$  Database  $\rightarrow$  Database

remove(k, db)  $\equiv$  db  $\setminus$  { (k, d) | d : Data  $\cdot$  true },

defined : Key  $\times$  Database  $\rightarrow$  **Bool**


defined(k, db)  $\equiv$   $(\exists d : \text{Data} \cdot (k, d) \in db)$ ,

lookup : Key  $\times$  Database  $\rightarrow$  Data

lookup(k, db) **as** d **post** (k, d)  $\in$  db

**pre** defined(k, db)

**end**

02262: Formal Aspects of Software Engineering I		 Technical University of Denmark Informatics and Mathematical Modelling Computer Science and Technology
Sets: ch.8, pp.54-62		
Foil 5.11		
Anne Haxthausen, IMM/DTU	Spring 2002	

### Set Example III — continued

Are the following properties true for SET\_DATABASE?

is\_wf\_Database(empty),

$\forall k : \text{Key}, d : \text{Data}, db : \text{Database} \cdot \text{is\_wf\_Database}(\text{insert}(k, d, db))$ ,

$\forall k : \text{Key}, db : \text{Database} \cdot \text{is\_wf\_Database}(\text{remove}(k, db))$