BCS2213 – Formal methods

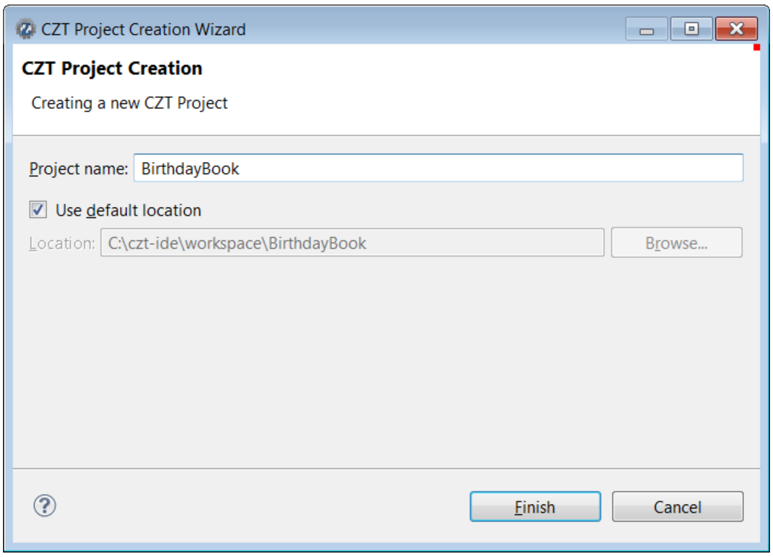
Teaching assignment 4. Z specification of BirthdayBook**.**

1. Run CZT-IDE.

   The Community Z Tools (CZT) is a set of tools for editing, typechecking and animating formal specifications written in Z specification language.

2. Creating a CZT project.

   Create a new CZT project by selecting **File > New > Project > CZT > CZT Project** in the menu.



Project name: BirthdayBook

☑ Use default location

Location: C:\czt-ide\workspace\BirthdayBook

- Enter the name of the new project.

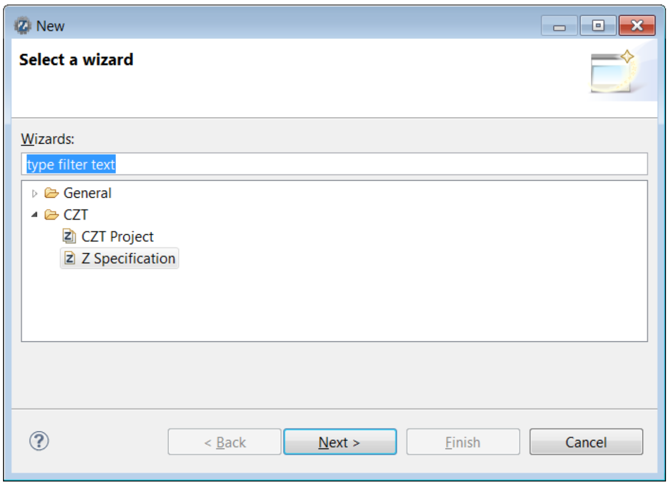
Note: you can create CZT project in your existing directories: just specify an existing directory in the Location field. The name does not have to match the directory.

- If the **Z perspective** is not active at the moment, the wizard will ask you whether you want to open the Z perspective. It is highly recommended that you open it because it will automatically open a set of Eclipse views useful for Z development.
- The new project will be created and selected in the workbench window.

3. Creating Z specification.

   The CZT editor plug-in provides a wizard to create a new Z specification. The wizard can be opened using the menu entry **File > New > Other… > CZT > Z Specification**.

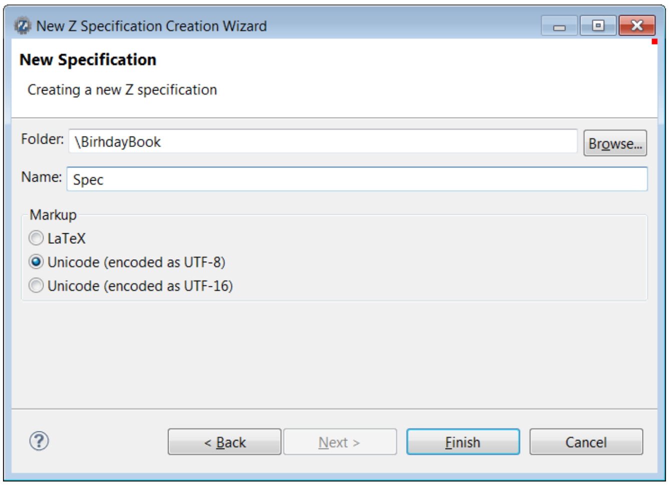
- In this wizard, enter the name of the new Z specification (please use Lab_5_<your ID> for the name) and folder (\BirthdayBook).
- As the Z specification can be written using LaTeX or Unicode markup, you need to specify a markup for the new specification. Choose Unicode (UTF 8) markup.
- The name of the new specification will be the name you specified above appended by the extension corresponding to the markup you selected.



Wizards:

type filter text

- General
- CZT
  - CZT Project
  - Z Specification



New Specification

Creating a new Z specification

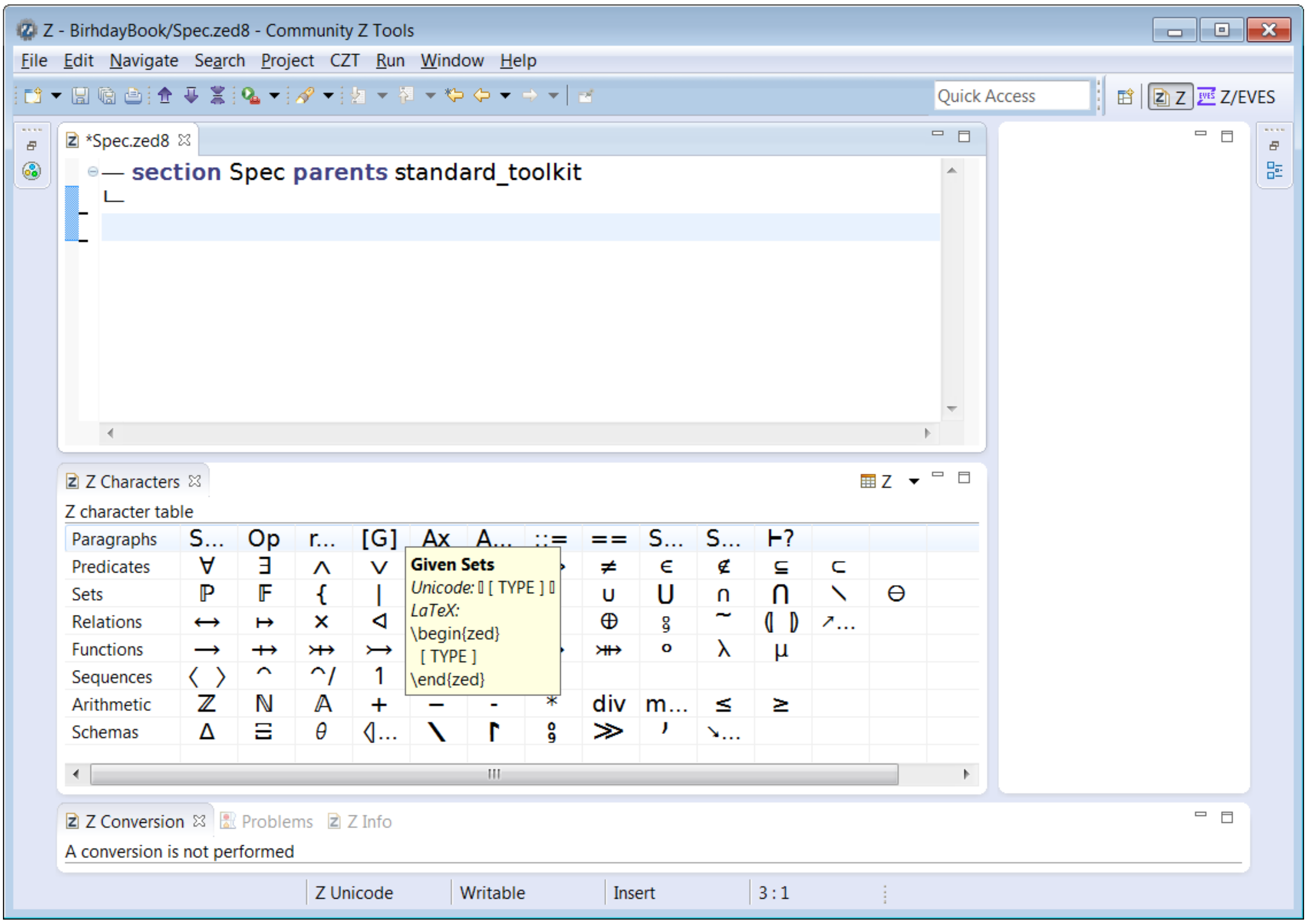
Folder: \BirhdayBook

Name: Spec

Markup

○ LaTeX
● Unicode (encoded as UTF-8)
○ Unicode (encoded as UTF-16)

- Then the new specification will be created with some initial contents and opened in the workbench window.



```
section Spec parents standard_toolkit
```

Z character table

Paragraphs	S…	Op	r…	[G]	Ax	A…	::=	==	S…	S…	⊢?		
Predicates	∀	∃	∧	∨				≠	∈	∉	⊆	⊂	
Sets	ℙ	𝔽	{	|				∪	⋃	∩	⋂	∖	⊖
Relations	↔	↦	×	◁				⊕	⨾	∼	⦇ ⦈	↗…	
Functions	→	⇸	⤀	↣				⤔	∘	λ	μ		
Sequences	⟨ ⟩	⁀	⁀/	1									
Arithmetic	ℤ	ℕ	𝔸	+	−	-	*	div	m…	≤	≥		
Schemas	Δ	Ξ	θ	⦉…	⧹	↾	⨾	≫	′	↘…			

Given Sets

Unicode: ▯ [TYPE] ▯

LaTeX:

```
\begin{zed}
 [ TYPE ]
\end{zed}
```

A conversion is not performed

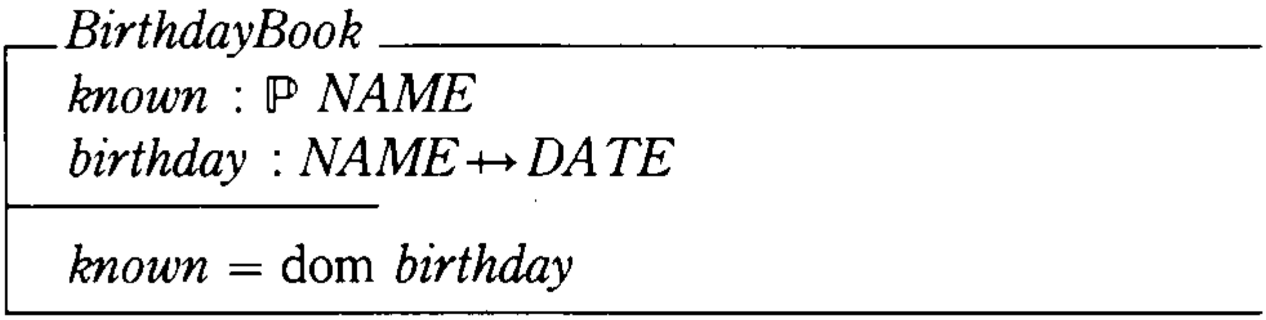
Z Unicode | Writable | Insert | 3 : 1

4. Develop Z specification of BirthdayBook for recording people's birthdays

5. First of all, define [NAME, DATE] as basic types of the specification.

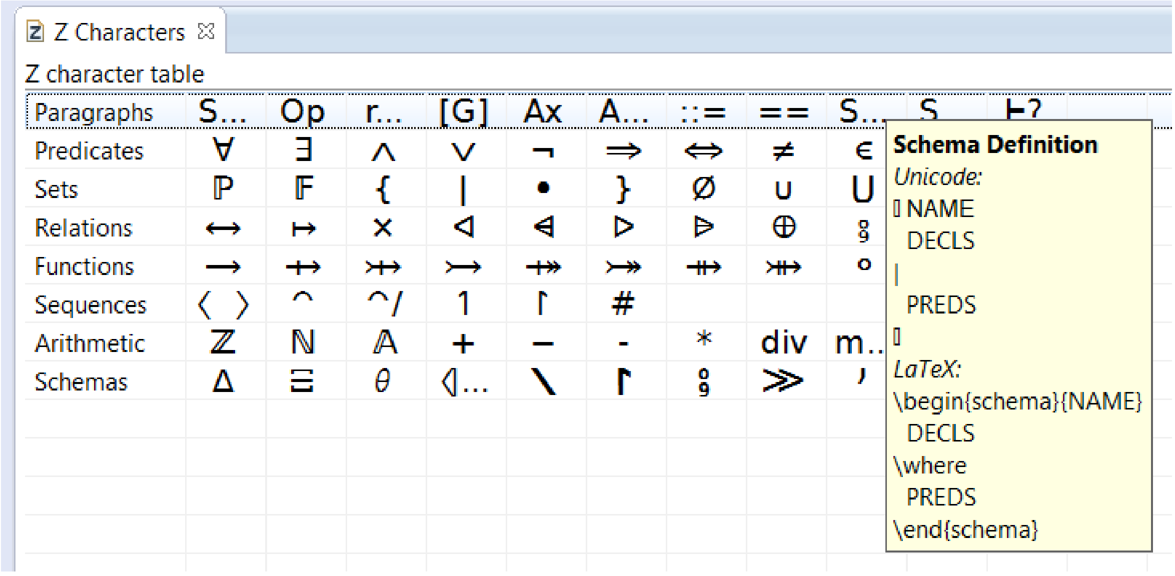
   For it, find in Z character table [G] cell (Given Sets) and press it (see the figure above).

6. Flowing given below Z specification, develop BirthdayBook schema.



```
┌─ BirthdayBook ──────────────────────
│ known : ℙ NAME
│ birthday : NAME ⇸ DATE
├──────────────
│ known = dom birthday
└─────────────────────────────────────
```

7. Find in Z character table Sch (Schema Definition) template and insert it into your specification.

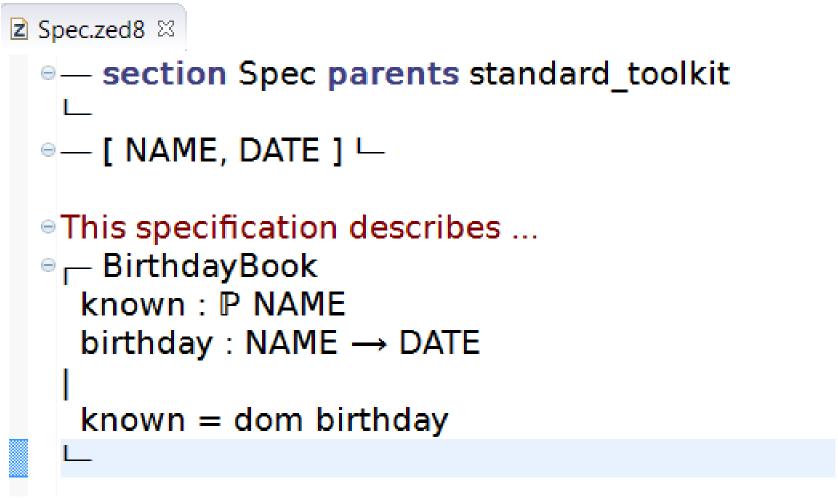


Z character table

Paragraphs	S...	Op	r...	[G]	Ax	A...	::=	==	S...	S	⊢?
Predicates	∀	∃	∧	∨	¬	⇒	⇔	≠	∈		
Sets	ℙ	𝔽	{	|	•	}	∅	∪	⋃		
Relations	↔	↦	×	◁	⩤	▷	⩥	⊕	⨾		
Functions	→	⇸	⤔	↣	⤀	⤖	⇻	⤕	∘		
Sequences	⟨ ⟩	⁀	⁀/	↿	↾	#					
Arithmetic	ℤ	ℕ	𝔸	+	−	-	*	div	m...		
Schemas	Δ	Ξ	θ	⦉...	\	↾	⨾	≫	′		

Schema Definition
Unicode:
▯ NAME
  DECLS
|
  PREDS
▯
LaTeX:
\begin{schema}{NAME}
  DECLS
\where
  PREDS
\end{schema}

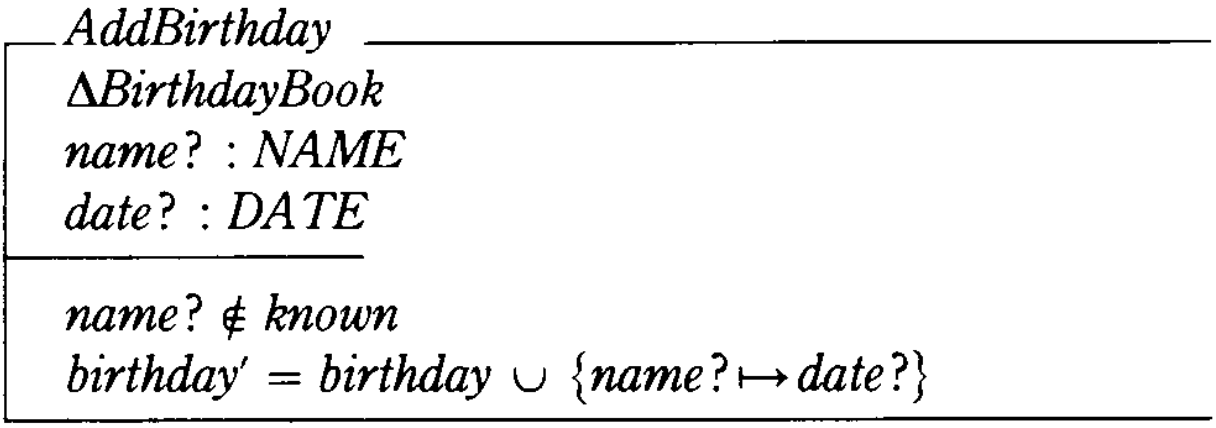
8. As result, at this stage your specification will looks like



```
— section Spec parents standard_toolkit
└
— [ NAME, DATE ] └

This specification describes ...
┌ BirthdayBook
 known : ℙ NAME
 birthday : NAME → DATE
|
 known = dom birthday
└
```

9. Specify Dynamic Aspects of the schema by adding AddBirthday operation.



AddBirthday

$\Delta BirthdayBook$

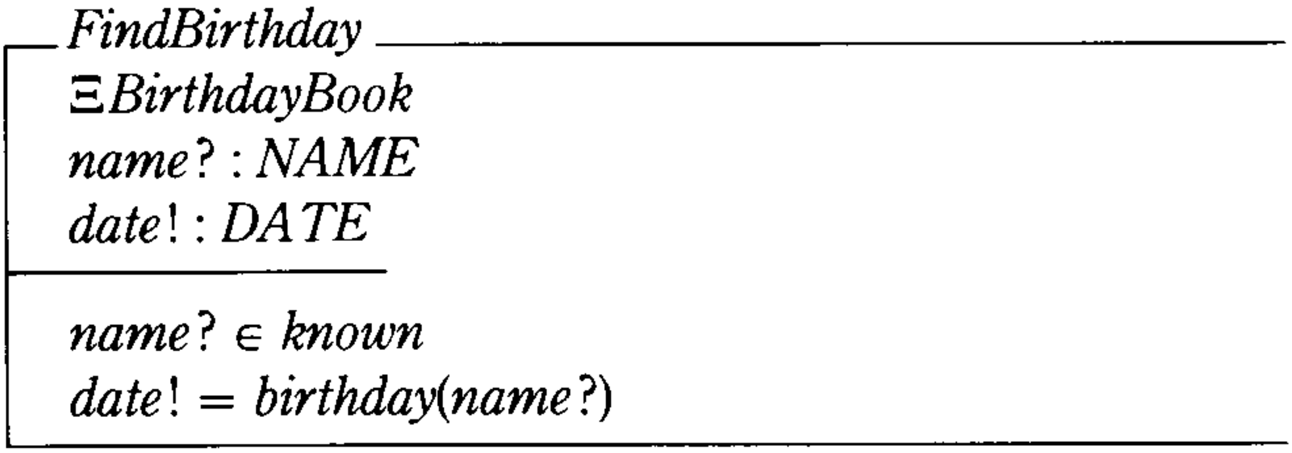
$name? : NAME$

$date? : DATE$

$name? \notin known$

$birthday' = birthday \cup \{name? \mapsto date?\}$

10. Add FindBirthday operation.



FindBirthday

$\Xi BirthdayBook$

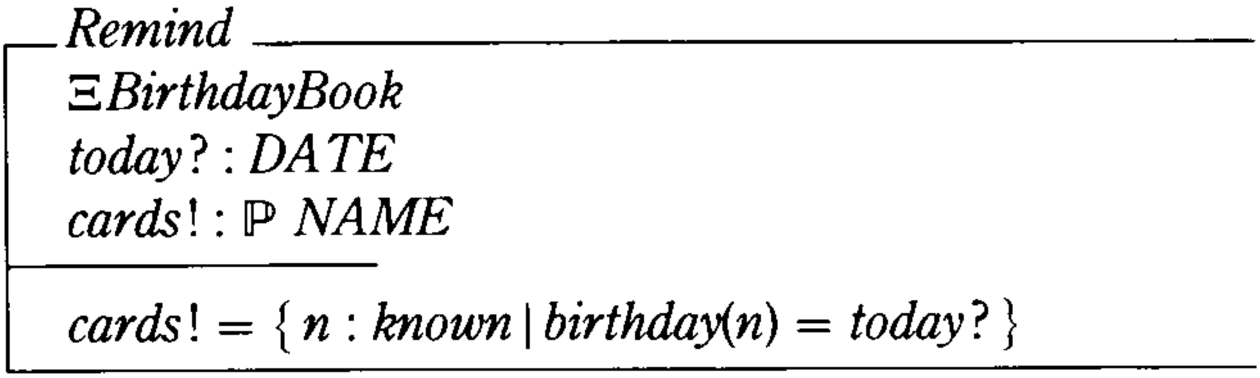
$name? : NAME$

$date! : DATE$

$name? \in known$

$date! = birthday(name?)$

11. Add *Remind* operation.



Remind

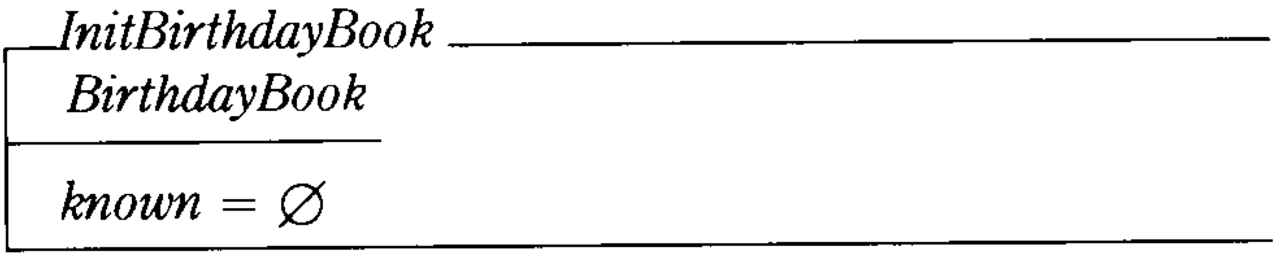
$\Xi BirthdayBook$

$today? : DATE$

$cards! : \mathbb{P}\ NAME$

$cards! = \{ n : known \mid birthday(n) = today? \}$

12. Describe the *initial state* of the system.



InitBirthdayBook

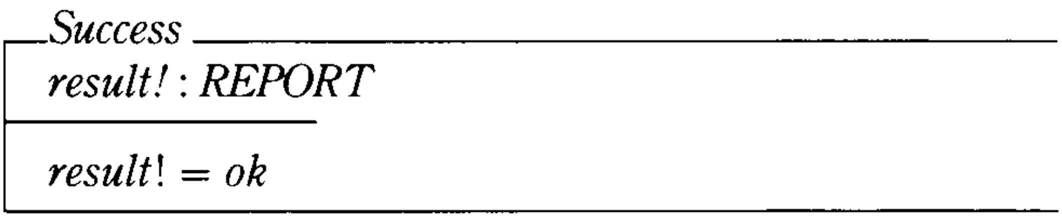
$BirthdayBook$

$known = \varnothing$

13. Strengthening *AddBirthday*

Define type *REPORT* ::= *ok* | *already_known* | *not_known* (use freetype definition symbol)

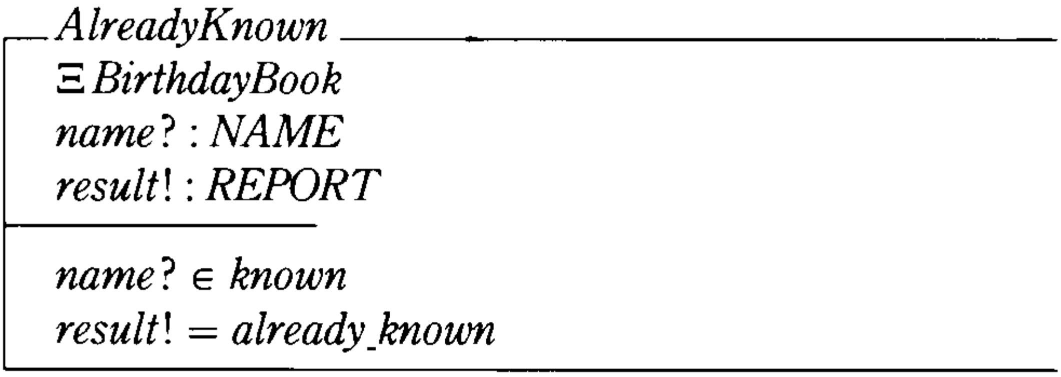
Define *Success* and *AlreadyKnown* schemas



Success

$result! : REPORT$

$result! = ok$



AlreadyKnown

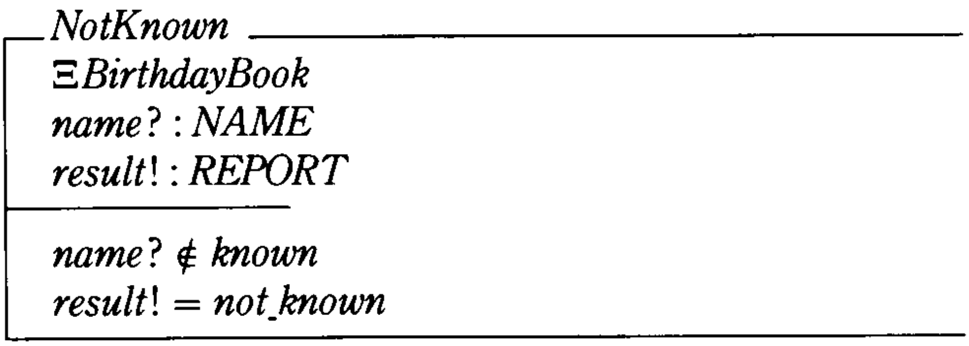
$\Xi BirthdayBook$
$name? : NAME$
$result! : REPORT$

$name? \in known$
$result! = already\_known$

14. Combine schemas *Success* and *AlreadyKnown*

$$RAddBirthday \mathrel{\widehat=} (AddBirthday \land Success) \lor AlreadyKnown$$

15. Define the schema *NotKnown*



NotKnown

$\Xi BirthdayBook$
$name? : NAME$
$result! : REPORT$

$name? \notin known$
$result! = not\_known$

16. Combine the schemas

$$RRemind \mathrel{\widehat=} Remind \land Success$$

$$RFindBirthday \mathrel{\widehat=} (FindBirthday \land Success) \lor NotKnown$$

17. Please upload your labsheet with commented properties and explanations into Moodle.