

BCS2213 – Formal methods

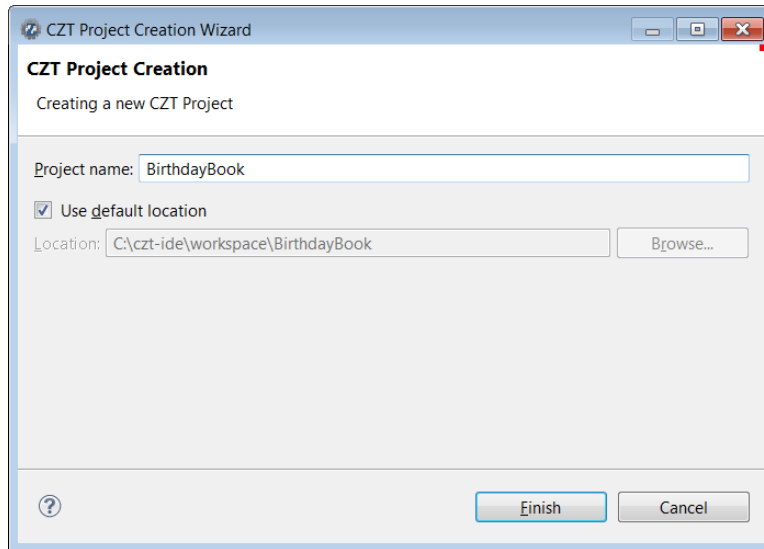
Teaching assignment 1. Z specification of BirthdayBook.

1. Run CZT-IDE.

The Community Z Tools (CZT) is a set of tools for editing and typechecking 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.



- Enter the name of the new project.

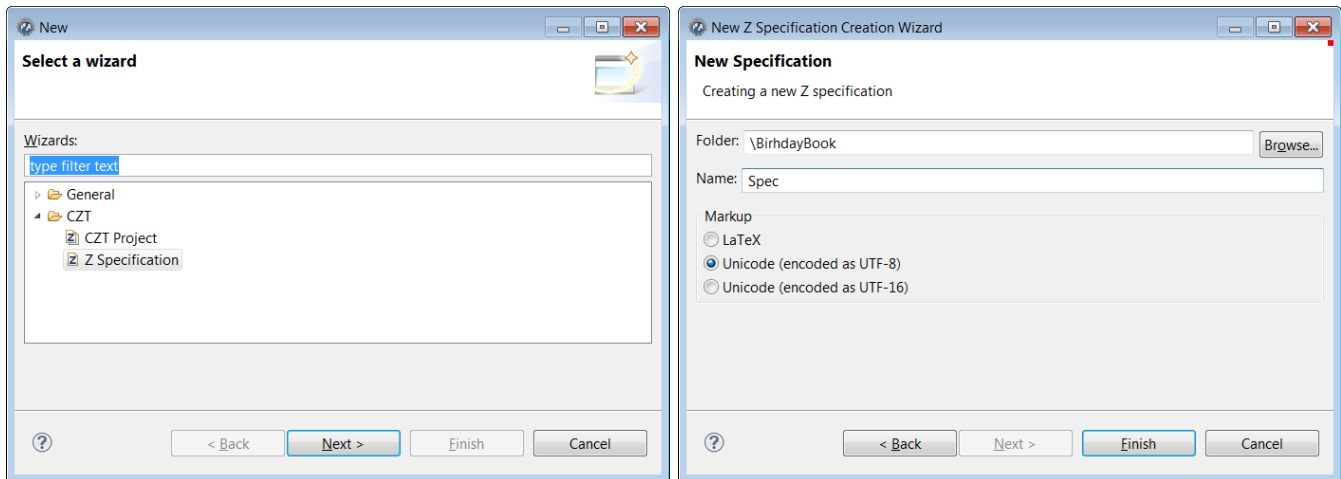
Note: you can create CZT project in your existing directories: just specify an existing directory in the Location field. The name does not have to match the directory.

- 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.

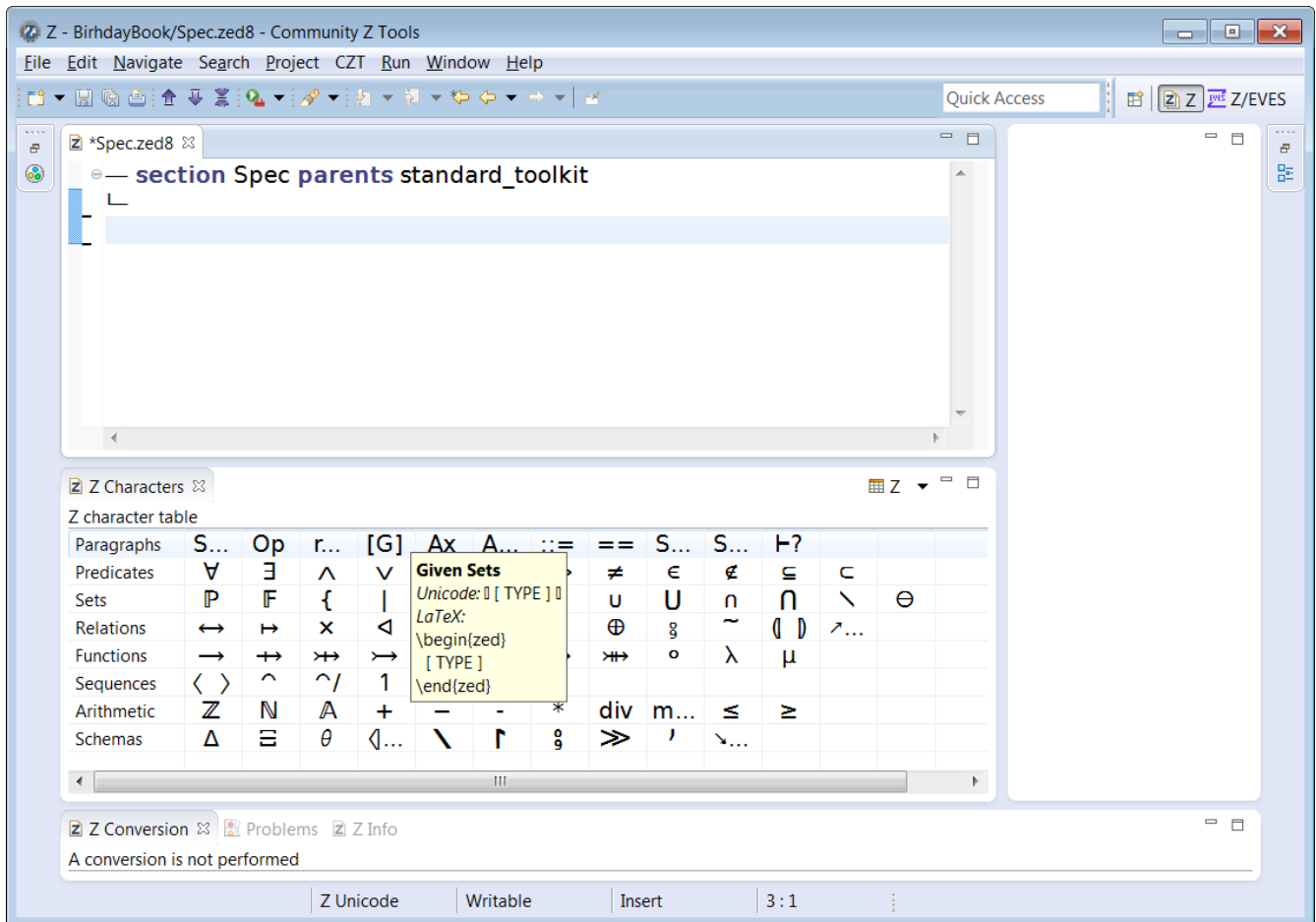
3. Creating Z specification.

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_1_<your ID> for the name) and folder (\BirthdayBook).
- As the Z specification can be written using LaTeX or Unicode, 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 appended by the extension corresponding to selected markup.



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



4. Develop Z specification of BirthdayBook for recording people's birthdays.
5. First of all, define [NAME, DATE] as the 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.

| |
|---------------------------------------------------------|
| <i>BirthdayBook</i> |
| <i>known</i> : $\mathbb{P} \text{ NAME}$ |
| <i>birthday</i> : $\text{NAME} \rightarrow \text{DATE}$ |
| <i>known</i> = dom <i>birthday</i> |

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

| Paragraphs | S... | Op | r... | [G] | Ax | A... | ::= | = | S... | S | ⊢? |
|------------|-------------------|--------------|--------------|-----------------|-----------------|------------------|-------------------|--------------|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----|
| Predicates | \forall | \exists | \wedge | \vee | \neg | \Rightarrow | \Leftrightarrow | \neq | \in | Schema Definition Unicode: \mathbb{P} NAME DECLS PREDS \mathbb{P} LaTeX: $\backslash\text{begin}\{\text{schema}\}\{\text{NAME}\}$ DECLS $\backslash\text{where}$ PREDS $\backslash\text{end}\{\text{schema}\}$ | |
| Sets | \mathbb{P} | \mathbb{F} | $\{$ | $ $ | \bullet | $\}$ | \emptyset | \cup | \mathbb{U} | | |
| Relations | \leftrightarrow | \mapsto | \times | \triangleleft | \triangleleft | \triangleright | \triangleright | \oplus | \otimes | | |
| Functions | \rightarrow | \mapsto | \mapsto | \mapsto | \mapsto | \mapsto | \mapsto | \mapsto | \mapsto | | |
| Sequences | $\langle \rangle$ | \wedge | $\wedge/$ | 1 | \uparrow | $\#$ | | | | | |
| Arithmetic | \mathbb{Z} | \mathbb{N} | \mathbb{A} | $+$ | $-$ | $-$ | $*$ | div | m.. | | |
| Schemas | Δ | Ξ | θ | $\langle \dots$ | \backslash | \uparrow | \otimes | \gg | , | | |

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

```

Spec.zed8
├─ 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 AddBirthday operation.

AddBirthday $\Delta \text{BirthdayBook}$ *name?* : *NAME**date?* : *DATE**name?* \notin *known**birthday'* = *birthday* \cup {*name?* \mapsto *date?*}

10. Add FindBirthday operation.

FindBirthday $\Xi \text{ BirthdayBook}$ *name?* : *NAME**date!* : *DATE**name?* \in *known**date!* = *birthday*(*name?*)

11. Add Remind operation.

Remind $\Xi \text{ BirthdayBook}$ *today?* : *DATE**cards!* : \mathbb{P} *NAME**cards!* = { *n* : *known* | *birthday*(*n*) = *today?* }12. Describe the *initial state* of the system.*InitBirthdayBook**BirthdayBook**known* = \emptyset

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

$\exists BirthdayBook$

$name? : NAME$

$result! : REPORT$

$name? \in known$

$result! = already_known$

14. Combine schemas *Success* and *AlreadyKnown*

$RAddBirthday \triangleq$

$(AddBirthday \wedge Success) \vee AlreadyKnown$

15. Define the schema *NotKnown*

NotKnown

$\exists BirthdayBook$

$name? : NAME$

$result! : REPORT$

$name? \notin known$

$result! = not_known$

16. Combine the schemas

$RRemind \triangleq Remind \wedge Success$

$RFindBirthday \triangleq (FindBirthday \wedge Success) \vee NotKnown$

17. Please upload your labsheet with commented properties and explanations into Kalam. Only file Lab_1_<your ID>.zed8 is needed (please also use Lab_1_<your ID> as the subject for e-mail).