Proof support for Z via HOL

Roger Bishop Jones
ICL Defence Systems
This document consists of the overheads for a presentation at IED project 1563 quarterly meeting on May 4 1990.
DEGREES of RECKLESSNESS
in SEEKING
PROOF SUPPORT for Z

1 RECORD SPECIFICATION in BARE
HOL

2 USE PRE/POST PROCESSORS with
HOL

3 IMPLEMENT Z-like PARSER/PRETTY
PRINTER for HOL

4 ADD some CONSPICUOUSLY CONSERVATIVE EXTENSIONS to HOL

5 CHANGE to LOGIC more SUITABLE
FOR Z

(CHANGE of PARADIGM
NOT CONSIDERED)
OBJECTIVE:

Proof support for Z

Achieve high assurance requirements

PROBLEM:

Complexity of Z

Lack of proof rules for Z

SOLUTION:

SEMANTIC EMBEDDING OF Z IN HOL
SEMANTIC EMBEDDING

CAN BE ACHIEVED USING ONLY CONSERVATIVE EXTENSIONS

CAN BE IMPLEMENTED WITHOUT KNOWLEDGE OF Z PROOF RULES

CONSTITUTES FORMAL SEMANTICS FOR Z

IMPLEMENTATION SUPPORTS DERIVATION OF PROOF RULES FOR Z

Z PROOF SYSTEM COMES WITH (ALMOST) SAME ASSURANCE OF CONSISTENCY AS HOL
HOW TO DO IT

1. Implement parser/type-checker for Z yielding abstract syntax decorated with types

2. Specify (in Z) and implement a map from the abstract syntax of Z to abstract syntax of HOL

3. Write pretty printer for Z (in HOL)

4. Specify and implement ‘definitions’ for Z constructs

5. Derive proof rules for Z

6. Develop libraries (of theorems, rules, tactics..)
HOL TYPES AND TERMS

TYPE ::= 

  mk_vartype << string >>

  | mk_type << string × seq TYPE >>

FTERM ::= 

  mk_var << string × TYPE >>

  | mk_const << string × TYPE >>

  | mk_comb << FTERM × FTERM >>

  | mk_abs << string × TYPE × FTERM >>
Z TYPES

ZTYPE ::= givenT <<IDENT>>
    | varT <<IDENT>>
    | powerT <<ZTYPE>>
    | tupleT <<seq ZTYPE>>
    | schemaT <<IDENT ZTYPE>>

GTYPE == seq IDENT \times ZTYPE