Appendix A

# Isabelle/Isar quick reference

## A.1 Proof commands

### A.1.1 Primitives and basic syntax

| | |
|---|---|
| **fix** $\overline{x}$ | augment context by $\bigwedge \overline{x}\,.\,\square$ |
| **assume** $a$: $\overline{\varphi}$ | augment context by $\overline{\varphi} \Longrightarrow \square$ |
| **then** | indicate forward chaining of facts |
| **have** $a$: $\varphi$ | prove local result |
| **show** $a$: $\varphi$ | prove local result, refining some goal |
| **using** $\overline{a}$ | indicate use of additional facts |
| **unfolding** $\overline{a}$ | unfold definitional equations |
| **proof** $m_1$ ... **qed** $m_2$ | indicate proof structure and refinements |
| **{** ... **}** | declare explicit blocks |
| **next** | switch blocks |
| **note** $a = \overline{b}$ | reconsider facts |
| **let** $p = t$ | abbreviate terms by higher-order matching |

$$
\begin{array}{rcl}
\textit{theory-stmt} & = & \textbf{theorem}\ \textit{name}\text{: }\textit{prop}\ \textit{proof} \mid \textbf{definition}\ \ldots \mid \ldots \\
\textit{proof} & = & \textit{prfx}^*\ \textbf{proof}\ \textit{method}\ \textit{stmt}^*\ \textbf{qed}\ \textit{method} \\
 & \mid & \textit{prfx}^*\ \textbf{done} \\
\textit{prfx} & = & \textbf{apply}\ \textit{method} \\
 & \mid & \textbf{using}\ \textit{name}^+ \\
 & \mid & \textbf{unfolding}\ \textit{name}^+ \\
\textit{stmt} & = & \textbf{\{}\ \textit{stmt}^*\ \textbf{\}} \\
 & \mid & \textbf{next} \\
 & \mid & \textbf{note}\ \textit{name} = \textit{name}^+ \\
 & \mid & \textbf{let}\ \textit{term} = \textit{term} \\
 & \mid & \textbf{fix}\ \textit{var}^+ \\
 & \mid & \textbf{assume}\ \textit{name}\text{: }\textit{prop}^+ \\
 & \mid & \textbf{then}^?\ \textit{goal} \\
\textit{goal} & = & \textbf{have}\ \textit{name}\text{: }\textit{prop}\ \textit{proof} \\
 & \mid & \textbf{show}\ \textit{name}\text{: }\textit{prop}\ \textit{proof}
\end{array}
$$

### A.1.2 Abbreviations and synonyms

| | | |
|---|---|---|
| **by** $m_1$ $m_2$ | $\equiv$ | **proof** $m_1$ **qed** $m_2$ |
| **..** | $\equiv$ | **by** *rule* |
| **.** | $\equiv$ | **by** *this* |
| **hence** | $\equiv$ | **then have** |
| **thus** | $\equiv$ | **then show** |
| **from** $\overline{a}$ | $\equiv$ | **note** *this* $= \overline{a}$ **then** |
| **with** $\overline{a}$ | $\equiv$ | **from** $\overline{a}$ **and** *this* |
| **from** *this* | $\equiv$ | **then** |
| **from** *this* **have** | $\equiv$ | **hence** |
| **from** *this* **show** | $\equiv$ | **thus** |

### A.1.3 Derived elements

| | | |
|---|---|---|
| $\textbf{also}_0$ | $\approx$ | **note** *calculation* $=$ *this* |
| $\textbf{also}_{n+1}$ | $\approx$ | **note** *calculation* $=$ *trans* $[OF$ *calculation this*$]$ |
| **finally** | $\approx$ | **also from** *calculation* |
| **moreover** | $\approx$ | **note** *calculation* $=$ *calculation this* |
| **ultimately** | $\approx$ | **moreover from** *calculation* |
| **presume** $a$: $\overline{\varphi}$ | $\approx$ | **assume** $a$: $\overline{\varphi}$ |
| **def** $a$: $x \equiv t$ | $\approx$ | **fix** $x$ **assume** $a$: $x \equiv t$ |
| **obtain** $\overline{x}$ **where** $a$: $\overline{\varphi}$ | $\approx$ | ... **fix** $\overline{x}$ **assume** $a$: $\overline{\varphi}$ |
| **case** $c$ | $\approx$ | **fix** $\overline{x}$ **assume** $c$: $\overline{\varphi}$ |
| **sorry** | $\approx$ | **by** *cheating* |

### A.1.4 Diagnostic commands

| | |
|---|---|
| **pr** | print current state |
| **thm** $\overline{a}$ | print theorems |
| **term** $t$ | print term |
| **prop** $\varphi$ | print meta-level proposition |
| **typ** $\tau$ | print meta-level type |

## A.2 Proof methods

**Single steps (forward-chaining facts)**

| | |
|---|---|
| *assumption* | apply some assumption |
| *this* | apply current facts |
| *rule* $\overline{a}$ | apply some rule |
| *rule* | apply standard rule (default for **proof**) |
| *contradiction* | apply ¬ elimination rule (any order) |
| *cases t* | case analysis (provides cases) |
| *induct* $\overline{x}$ | proof by induction (provides cases) |

**Repeated steps (inserting facts)**

| | |
|---|---|
| – | no rules |
| *intro* $\overline{a}$ | introduction rules |
| *intro_classes* | class introduction rules |
| *elim* $\overline{a}$ | elimination rules |
| *unfold* $\overline{a}$ | definitions |

**Automated proof tools (inserting facts, or even prems!)**

| | |
|---|---|
| *rules* | intuitionistic proof search |
| *blast*, *fast* | Classical Reasoner |
| *simp*, *simp_all* | Simplifier (+ Splitter) |
| *auto*, *force* | Simplifier + Classical Reasoner |
| *arith* | Arithmetic procedure |

## A.3 Attributes

**Operations**

| | |
|---|---|
| *OF* $\overline{a}$ | rule resolved with facts (skipping "_") |
| *of* $\overline{t}$ | rule instantiated with terms (skipping "_") |
| *where* $\overline{x} = \overline{t}$ | rule instantiated with terms, by variable name |
| *symmetric* | resolution with symmetry rule |
| *THEN b* | resolution with another rule |
| *rule_format* | result put into standard rule format |
| *elim_format* | destruct rule turned into elimination rule format |

**Declarations**

| | |
|---|---|
| *simp* | Simplifier rule |
| *intro*, *elim*, *dest* | Pure or Classical Reasoner rule |
| *iff* | Simplifier + Classical Reasoner rule |
| *split* | case split rule |
| *trans* | transitivity rule |
| *sym* | symmetry rule |

## A.4 Rule declarations and methods

| | *rule* | *rules* | *blast* etc. | *simp* etc. | *auto* etc. |
|---|---|---|---|---|---|
| *elim*! *intro*! (Pure) | × | × | | | |
| *elim intro* (Pure) | × | × | | | |
| *elim*! *intro*! | × | | × | | × |
| *elim intro* | × | | × | | × |
| *iff* | × | | × | × | × |
| *iff*? | × | | | | |
| *elim*? *intro*? | × | | | | |
| *simp* | | | | × | × |
| *cong* | | | | × | × |
| *split* | | | | × | × |

## A.5 Emulating tactic scripts

### A.5.1 Commands

| | |
|---|---|
| **apply** $m$ | apply proof method at initial position |
| **apply_end** $(m)$ | apply proof method near terminal position |
| **done** | complete proof |
| **defer** $n$ | move subgoal to end |
| **prefer** $n$ | move subgoal to beginning |
| **back** | backtrack last command |

### A.5.2 Methods

| | |
|---|---|
| *rule_tac insts* | resolution (with instantiation) |
| *erule_tac insts* | elim-resolution (with instantiation) |
| *drule_tac insts* | destruct-resolution (with instantiation) |
| *frule_tac insts* | forward-resolution (with instantiation) |
| *cut_tac insts* | insert facts (with instantiation) |
| *thin_tac* $\varphi$ | delete assumptions |
| *subgoal_tac* $\varphi$ | new claims |
| *rename_tac* $\overline{x}$ | rename suffix of goal parameters |
| *rotate_tac n* | rotate assumptions of goal |
| *tactic text* | arbitrary ML tactic |
| *case_tac t* | exhaustion (datatypes) |
| *induct_tac* $\overline{x}$ | induction (datatypes) |
| *ind_cases t* | exhaustion + simplification (inductive sets) |