

# Übersicht: Taktiken

## Unterschiede der Taktiken

Isabelle bietet verschiedenste Taktiken für die Durchführung von Beweisen an. Diese unterscheiden sich jedoch deutlich in ihren Möglichkeiten und Einsatzbereichen. Unten stehende Tabelle soll dies verdeutlichen, vorher sollen jedoch noch die einzelnen Kategorien, in denen sie sich unterscheiden, verdeutlicht werden:

**Simplifikation:** Diese Kategorie gibt an, ob die jeweilige Taktik Termersetzungsregeln anwendet

**simp:** Termersetzungsregeln werden angewandt

**no simp:** Termersetzungsregeln werden nicht angewandt

**Deduktion:** Wendet die Taktik Deduktionsregeln an? Und wenn ja, welche?

**no deduction:** Keine Anwendung von Deduktionsregeln

**constructive:** Die Taktik verwendet nur Regeln, die der konstruktivistischen Logik entsprechen (also z.B. nicht die Regel `classical`)

**classical:** Die Taktik verwendet alle Deduktionsregeln

**safe:** Nur sichere Regeln werden angewandt (weniger als **classical**)

**obvious:** Nur "offensichtliche" Regeln werden angewandt (weniger als **safe**)

**Subgoal:** Diese Kategorie gibt an, wie mit dem aktuelle Subgoal verfahren wird.

**whole subgoal:** Die Taktik versucht, das Subgoal komplett zu lösen und bricht mit einem Fehler ab, wenn ihr das nicht gelingt

**split:** Die Taktik vereinfacht das Subgoal soweit es kann und kann dabei auch das Subgoal in weitere Subgoals aufteilen

**no split:** Die Taktik vereinfacht das Subgoal soweit es kann, teilt es jedoch nicht in weitere Subgoals auf

**Amount of Subgoals:** Wird die Taktik auf nur ein oder alle vorhandenen Subgoals angewandt?

**one:** Taktik wird nur auf das erste Subgoal angewandt

**all:** Taktik wird auf alle Subgoals angewandt

	simp	no simp	no deduction	constructive	classical	safe	obvious	whole subgoal	split	no split	one	all
simp	+		+							+	+	
simp_all	+		+							+		+
clarify		+					+			+	+	
clarsimp	+						+			+	+	
safe		+				+			+		+	
iprover		+		+				+			+	
fast		+			+			+			+	
fastsimp	+				+			+			+	
blast		+			+			+			+	
force	+				+			+			+	
auto	+				+				+			+

Bemerkungen: `blast` eignet sich gut für Quantoren, `force` testet auch Möglichkeiten, die `fast`, `blast` oder `auto` nicht testen, kann dafür aber deutlich länger brauchen bzw. schneller in Endlosschleifen hängen bleiben

## Regeln zu Taktiken hinzufügen

**Simplifikationsregeln:** Termersetzungsregeln können zu allen Regeln mit Simplifikation hinzugefügt werden, solange sie die benötigte Form haben (Gleichung). Dies geschieht bei `simp` und `simp_all` durch Anhängen von `add:`, also z.B. `apply(simp_all add:bla1 bla2)`. Bei allen anderen Regeln muss ein `simp:`, gefolgt vom Regelnamen, angehängt werden, also z.B. `apply(auto simp:bla)`. Es ist auch möglich, Regeln zu entfernen (weil die Simplifikation z.B. in einer Endlosschleife hängt oder ein nicht gewünschtes Resultat liefert). Dies geschieht durch `del:` und dem entsprechenden Lemmanamen. Vorsicht: in den Deduktions-taktiken muss auch noch `simp` vor das `del:` geschrieben werden. Ebenso ist es auch möglich, den Simplifier nur mit bestimmten Regeln aufzurufen mittels `only:` und den entsprechenden Regeln (auch hier bei den entsprechenden Taktiken das zusätzliche `simp` nicht vergessen). Beispiele: `apply(simp add:hip)`, `apply(simp_all only:hip hop)`, `apply(fastsimp simp del:foo bar)`, `apply(auto simp only:zoo)`

**Deduktionsregeln:** Alle Taktiken mit Deduktion können weitere Deduktionsregeln in ihre anzuwendenden Regeln aufnehmen, nur unterscheidet man hier zwischen Introduktions-, Eliminations- und Destruktionsregeln. Erstere werden durch `intro:` angehängt, zweitere durch `elim:` und letztere durch `dest:`. Kombinationen sind natürlich auch möglich, also z.B.

```
apply(fastsimp dest:foo intro:bar elim:zip zap).
```