# Relating Formulas to Taclets

Peter H. Schmitt

Karlsruhe Institute of Technology (KIT), Dept. of Informatics
Am Fasanengarten 5, 76131 Karlsruhe, Germany

This note relates the formulas - axioms, definitions, lemmas- used in the technical report [1] to the names of the taclets that implement them in the KeY system. This is particularly useful in finding the proof files for derived lemmas since the taclet name is part of the name of the proof file.

## 1 The Core Theory

1. $\forall x, y, z(x < y \land y < z \to x < z)$ — transitivity
   `olt_transAxiom`, `olt_trans`, `olt_transAut`
2. $\forall x(\neg x < x)$ — strict order
   `olt_irref\Axiom`, `olt_irref`
3. $\forall x, y(x < y \lor x \doteq y \lor y < x)$ — total order
   `olt_total_Axiom`
4. $\forall x(0 \leq x)$ — 0 is smallest element
   `oleq_zeroAxiom`, `olt_0Min`, `oleq_zero`
5. $0 < \omega \land \neg\exists x(\omega \doteq x + 1)$ — $\omega$ is a limit ordinal
   `omegaDef1`
6. $\forall y(0 < y \land \forall x(x < \omega \to x + 1 < y) - > \omega \leq y)$ — $\omega$ is the least limit ordinal
   `omegaDefLeastInf`
7. $\forall x(x < x + 1) \land \forall x, y(x < y \to x + 1 \leq y)$ — $x + 1$ is successor function
   `oSucc`, `oLeastSucc`
8. $\forall x(\forall y(y < x \to \phi(y)) \to \phi) \to \forall x\phi$ — transfinite induction scheme
   `oIndBasic`
9. $\forall x, y, z(\phi(x, y) \land \phi(x, z) \to y = z) \to$
   $\forall a\exists b\forall y(\exists x(\phi(x, y) \land x < a) \to y < b)$ — replacement axiom scheme
   `oReplacementScheme`
10. $\forall x, y(x \leq y \leftrightarrow x < y \lor x = y)$ — Def. of $\leq$
    `oleq_Def`, `oleq_replace`
11. $\forall x(lim(x) \leftrightarrow x \neq 0 \land \neg\exists y(x = y + 1))$ — Def. of limit ordinal
    `olimDef`

**Fig. 1.** The axioms of the Core Theory

## 2 Basic Lemmas From The Core Theory

12. $lim(\lambda) \leftrightarrow \lambda \neq 0 \wedge \forall ov(ov < \lambda \rightarrow (ov + 1) < \lambda)$ equivalent Def. of limit numbers
`olimDefEquiv`, `olimDefAdd`, `notLim1`, `notLim2`
13. $\phi(o_0) \wedge \forall x(\phi(x) \rightarrow \phi(x + 1)) \wedge$ variant of induction scheme
$\forall x(lim(x) \wedge \forall y(y < x \rightarrow \phi(y)) \rightarrow \phi(x)$
$\rightarrow \quad \forall x \phi(x)$
`oInd`
14. $\exists x \phi \rightarrow \exists x(\phi \wedge \forall y(y < x \rightarrow \neg\phi(y)))$ least number principle
`least_number_principle`
15. $\forall a \exists b \forall \lambda(\lambda < a \rightarrow t < b)$ special case of replacement scheme
`oSpecialReplacment`

**Fig. 2.** Basic Lemmas of the Core Theory

## 3 First Definitional Extensions

16. $0 + 1 = 1$ Def. of constant 1
`one_Def`, `oadd01`
17. $\forall x(\neg x < 0)$
`olt_zero`
18. $0 < 1$
`olt_01`
19. $0 \neq 1$
`oDiff01`
20. $\forall x(0 < x \rightarrow 1 \leq x)$
`olt_discret`
21. $\forall x(x < 1 \rightarrow x = 0)$
`olt_one`
22. $0 < \omega$
`omegaZero`
23. $1 < \omega$
`omegaOne`

**Fig. 3.** Definitional Extension for constant 1

24. $\forall x(x + 1 \neq 0)$
`o0notSuccQ`, `o0notSucc`
25. $x + 1 \doteq y + 1 \rightarrow x \doteq y$
`OSuccInjective`
26. $x < y \rightarrow x + 1 < y + 1$
`oltPlusOne`
27. $x \leq y \rightarrow x + 1 \leq y + 1$
`oleqPlusOne`

**Fig. 4.** Lemmas for immediate successor

28. $x < y + 1 \rightarrow (x < y \vee x \doteq y)$
`olessPlusOne`
29. $x \leq y \wedge y \leq z \rightarrow x \leq z$
`oleq_trans`, `oleq_transAut`
30. $x \leq y \wedge y < z \rightarrow x < z$
`oltleq_trans`, `oltleq_transAut`, `oleqolt_transQ`
31. $x < y \wedge y \leq z \rightarrow x < z$
`oleqolt_trans`, `oleqolt_transAut`
32. $x < y \rightarrow \neg y < x$
`irrByolt`
33. $x \leq y \rightarrow \neg y < x$
`irrByoltleq`
34. $x < y \rightarrow \neg y \leq x$
`olt2oleq`
35. $x \leq y \wedge y \leq x \rightarrow x \doteq y$
`oleq_antisym`

**Fig. 5.** Lemmas on transitivity and related topics

# 4 Definitional Extensions for Maximum and Supremum

| | |
|---|---|
| 36. $\forall x, y(omax(x, y) \doteq (\text{if } \; x \leq y \; \text{ then } \; y \; \text{ else } \; x))$ | Def. of binary maximum<br>taclet: `omaxDef` |
| 37. $z < omax(x, y) \leftrightarrow (z < x \vee z < y)$ | taclet `omaxLess` |
| 38. $omax(x, y) < z \leftrightarrow (x < z \wedge y < z)$ | taclet `omaxGreater` |
| 39. $z \leq omax(x, y) \leftrightarrow (z \leq x \vee z \leq y)$ | taclet `omaxLeq` |
| 40. $omax(x, y) \leq z \leftrightarrow (x \leq z \wedge y \leq z)$ | taclet `omaxGeq` |
| 41. $omax(0, x) \doteq x$ | taclet `omax0Left` |
| 42. $omax(x, 0) \doteq x$ | taclet `omax0Right` |
| 43. $x \leq omax(x, y)$ | taclet `omaxLeft` |
| 44. $y \leq omax(x, y)$ | taclet `omaxRight` |
| 45. $(x < y \wedge y \doteq z) \rightarrow x < z$ | taclet `WRolteq` |
| 46. $omax(x, y) \doteq omax(y, x)$ | taclet `omaxSymQ` |
| 47. $x < y \rightarrow omax(x + 1, y) \doteq omax(x, y)$ | taclet `omaxPlusOnR` |
| 48. $x < y \rightarrow omax(y, x + 1) \doteq omax(x, y)$ | taclet `omaxPlusOnL` |
| 49. $omax(x, y + 1) \leq omax(x, y) + 1$ | taclet `omaxPlusOneQR` |
| 50. $omax(x + 1, y) \leq omax(x, y) + 1$ | taclet `omaxPlusOneQL` |

**Fig. 6.** Definitional Extensions for *omax*

51. $\forall x(x < t_0 \to t_1(x) \leq sup_{\lambda<t_0}(t_1(\lambda))) \quad \wedge$ Def. of supremum
$\forall y(\forall x(x < t_0 \to t_1(x) \leq y) \to sup_{\lambda<t_0}(t_1) \leq y))$
taclet: `osupDef`
52. $sup_{\lambda<0} t \doteq 0$ taclet `osup0`
53. $sup_{\lambda<1} t \doteq t[0]$ taclet `osup1`
54. $lim(x) \to sup_{\lambda<x}\lambda \doteq x$ taclet `oselfSup`
55. $sup_{\lambda<x+1}\lambda \doteq x$ taclet `oselfSupSuc`
56. $sup_{\lambda<x+1} t \doteq omax(sup_{\lambda<x} t, t[x])$ taclet `osupSucc`
57. $\forall\lambda(t_1 \doteq t_2) \to sup_{\lambda<x} t_1 \doteq sup_{\lambda<x} t_2$ taclet `osupEqualTerms`
58. $\forall x(x < z_1 \to \exists y(y < z_2 \wedge t_1[x] \leq t_2[y])) \wedge \forall y(y < z_2 \to \exists x(x < z_1 \wedge t_2[y] \leq t_1[x]))$
$\to sup_{\lambda<z_1} t_1 \doteq sup_{\lambda<z_2} t_2$
taclet `osupMutualCofinal`
59. $\forall\lambda(t_1 \leq t_2) \to sup_{\lambda<b}\ t_1 \leq sup_{\lambda<b}\ t_2$ taclet: `osupLocalLess`
60. $b_1 \leq b_2 \to sup_{\lambda<b_1} t \leq sup_{\lambda<b_2} t$ taclet: `osupShorter`
61. $sup_{\lambda<\omega}\lambda = \omega$ taclet: `enum:osupOmega`

**Fig. 7.** Definitional Extensions for *sup*

62. $\forall x, y, z(x \leq y \wedge y \leq z \to x \leq z)$ taclets. `oleq_trans, oleq_transAut`
63. $\forall x, y, z(x \leq y \wedge y < z \to x < z)$ taclets. `oltleq_trans, oltleq_transAut`
64. $\forall x, y, z(x < y \wedge y \leq z \to x < z)$ taclets: `oleqolt_trans, oleqolt_transAut`
65. $\forall x, y, z(z < (max(x, y) \leftrightarrow (z < x \vee z < y)))$ taclet: `omaxLess`
66. $\forall x, y, z(max(x, y) < z \leftrightarrow (x < z \wedge y < z))$ taclet: `omaxGreater`
67. $\forall x, y(max(x, y) \doteq max(y, x))$ taclet: `omaxSymQ`

**Fig. 8.** Derivable taclets on $\leq$ and *max*

## 5 Embedding Natural Numbers

68. $onat(0) \doteq 0$ taclet: `onatZeroDef`
69. $0 \leq n \to onat(n+1) \doteq onat(n) + 1$ taclet: `onatSuccDef`
70. $onat(1) \doteq 1$ taclet: `onatOne`
71. $onat(2) \doteq (0+1) + 1$ taclet: `onatTwo`
72. $onat(3) \doteq onat(2) + 1$ taclet: `onatThree`
73. $onat(4) \doteq onat(3) + 1$ taclet: `onatFour`
74. $onat(5) \doteq onat(4) + 1$ taclet: `onatFive`
75. $onat(6) \doteq onat(5) + 1$ taclet: `onatSix`
76. $onat(7) \doteq onat(6) + 1$ taclet: `onatSeven`
77. $onat(8) \doteq onat(7) + 1$ taclet: `onatEight`
78. $onat(9) \doteq onat(8) + 1$ taclet: `onatNine`
79. $(0 \leq n \land 0 \leq m) \to onat(n+m) \doteq onat(n) + onat(m)$ taclet: `onatoadd`
80. $(0 \leq n \land 0 \leq m \land onat(n) \doteq onat(m)) \to n \doteq m$ taclet: `onatInj`
81. $(0 \leq n \land 0 \leq m) \to (onat(n) < onat(m) \leftrightarrow n < m)$ taclet: `onatolt, onatoltAut`
82. $0 \leq n \to onat(n) < \omega$ taclet: `onatLessOmega`

**Fig. 9.** Definition of and lemmas for the injection *onat*

## 6 Ordinal Arithmetic

83. $x + 0 \doteq x$ taclet: `oadd_Def0Right`
84. $x + (y+1) \doteq (x+y) + 1$ taclet: `oadd_DefSucc`
85. $lim(y) \to x + y \doteq sup_{\lambda<y}(x+\lambda)$ taclet: `oadd_DefLim`
86. $x * 0 \doteq 0$ taclet: `otimes_Def0Right`
87. $x * (y+1) \doteq x * y + x$ taclet: `otimes_DefSucc`
88. $lim(y) \to x * y \doteq sup_{\lambda<y}(x * \lambda)$ taclet: `otimes_DefLim, otimes_DefLimQ`
89. $x^0 \doteq 1$ taclet: `oexp_Def0Right`
90. $x^{y+1} \doteq x^y * x$ taclet: `oexp_DefSucc`
91. $(lim(y) \land 0 < x) \to x^y \doteq sup_{\lambda<y} x^\lambda$ taclet: `oexp_DefLim`
92. $lim(y) \to 0^y \doteq 0$ taclet: `oexp_DefLim0`

**Fig. 10.** Definition of ordinal arithmetic operations

93. $y \neq 0 \to x < x + y$ taclet: `oaddStrictMonotone`
94. $x \leq x + y$ taclet: `oaddMonotone`
95. $y \leq x + y$ taclet: `oaddLeftMonotone`
96. $x + y \doteq 0 \to (x \doteq 0 \land y \doteq 0)$ taclet: `zerosum`
97. $x < y \to z + x < z + x$ taclet: `oltAddLessLeft`
98. $x \leq y \to z + x \leq z + x$ taclet: `oleqAddLessLeft`
99. $x \leq y \to x + z \leq y + z$ taclet: `oleqAddLessRight, oleqAddLessRightQ`
100. $(x < y \land u < w) \to x + u < y + w$ taclet: `oadd2olt`
101. $(x \leq y \land u \leq w) \to x + u \leq y + w$ taclet: `oadd2oleq`
102. $max(z + x, z + y) \doteq z + max(x, y)$ taclet: `omaxAddL`
103. $max(x + z, y + z) \doteq max(x.y) + z$ taclet: `omaxAddR`

**Fig. 11.** Lemmas on addition and order

104. $lim(y) \to lim(x + y)$ taclet: `olimAddolim`
105. $lim(x) \to \omega \leq x$ taclet: `omegaLeastLim1, omegaLeastLim2`
106. $(lim(x) \land x \leq \omega) \to x \doteq \omega$ taclet: `omegaLeastLim3`
107. $lim(x) \to 0 < x$ taclet: `limitZero`
108. $lim(x) \to 1 < x$ taclet: `limitOne`
109. $z + x \doteq z + y \to x \doteq y$ taclet: `oaddRightInjective`
110. $(lim(y) \land x < y) \to (x + 1) < y$ taclet: `olimDedekind`
111. $x < \omega \land y < \omega) \to (x + y) < \omega$ taclet: `oaddLessOmega, oaddLessOmegaAxiom`
112. $0 + x \doteq x$ taclet: `oadd0Left`
113. $x < \omega \to x + \omega \doteq \omega$ taclet: `oaddLeftomega`
114. $(x < \omega \land \omega \leq y) \to x + y \doteq y$ taclet: `oaddLeftAbsorb`
115. $\omega \leq x \to \exists y, n(lim(y) \land n < \omega \land x \doteq y + n)$ taclet: `repLimPlusNat`
116. $x \leq y \to \exists z(x + z \doteq y)$ taclet: `ordDiff`
117. $x + 1 \doteq y + 1 \to x \doteq y$ taclet: `oAddOneInj`
118. $x + y < x + z \to y < z$ taclet: `oAddOltPreserv`
119. $b \neq 0 \to sup_{\lambda<b}(x + y) = x + sup_{\lambda<b}y$ if $\lambda$ not free in $x$ taclet: `osupAddStaticTerm`
120. $x + (y + z) \doteq (x + y) + z$ taclet: `oaddAssoc`
121. $y < \omega \to 1 + y \doteq y + 1$ taclet: `oaddFiniteComOn`
122. $(x < \omega \land y < \omega) \to x + y \doteq y + x$ taclet: `oaddFiniteCom`

**Fig. 12.** Lemmas on addition

123. $x * 1 \doteq x$ taclet: `otimesOneRight`
124. $1 * x \doteq x$ taclet: `otimesOneLeft`
125. $0 * x \doteq 0$ taclet: `otimesZeroLeft`
126. $(0 < z \wedge x < y) \rightarrow z * x < z * y$ taclet: `otimesMonotone, otimesMonotoneQ`
127. $x \leq y \rightarrow z * x \leq z * y$ taclet: `otimesWeakMonotoneQ`
128. $z * x < z * y \rightarrow (0 < z \wedge x < y)$ taclet: `otimesMonotoneRev`
129. $(0 < z \wedge z * x \doteq z * y) \rightarrow x \doteq y$ taclet: `otimesLeftInjective`
130. $x \leq y \rightarrow x * z \leq y * z$ taclet: `otimesLeftMonotone`
131. $0 \neq x \rightarrow y \leq x * y$ taclet: `otimesRightMonotoneQ`
132. $x * y \doteq 0 \rightarrow (x \doteq 0 \vee y \doteq 0)$ taclet: `otimesZero`
133. $x * y \doteq 1 \rightarrow (x \doteq 1 \wedge y \doteq 1)$ taclet: `otimesOne`
134. $(x < \omega \wedge y < \omega) \rightarrow x * y < \omega$ taclet: `otimesFiniteAxiom, otimesFinite`
135. $(x \neq 0 \wedge x < \omega) \rightarrow x * \omega \doteq \omega$ taclet: `otimesNomega, otimesNomegaQ`
136. $max(z * x, z * y) \doteq z * max(x, y)$ taclet: `omaxTimesL`
137. $max(x * z, y * z) \doteq max(x, y) * z$ taclet: `omaxTimesR`
138. $sup_{\lambda<b} x * y \doteq x * sup_{\lambda<b} y$ provided $\lambda$ is not free in $x$.
taclet: `osupTimesStaticTerm`
139. $x * (y + z) \doteq x * y + x * z$ taclet: `odistributive, odistributiveQ`
140. $(x < \omega \wedge y < \omega \wedge z < \omega) \rightarrow (x + y) * z \doteq x * z + y * z$ taclet: `odistributiveFinite`
141. $x * (y * z) \doteq (x * y) * z$ taclet: `otimesAssoc, otimesAssocQ`
142. $(x < \omega \wedge y < \omega) \rightarrow x * y \doteq y * x$ taclet: `otimesFiniteCom`
143. $(x < \omega \wedge y < \omega \wedge \omega * x < \omega * y) \rightarrow x < y$ taclet: `oltomegatimes`
144. $(x_1 < \omega \wedge x_2 < \omega \wedge y_1 < \omega \wedge y_2 < \omega \wedge$
$\omega * x_1 + y_1 < \omega * x_2 + y_2) \rightarrow$
$\omega * x_1 < \omega * x_2 \vee (\omega * x_1 \doteq \omega * x_2 \wedge y_1 < y_2)$
taclet: `oltlexicographic`
145. $(0 \leq n_1 \wedge 0 \leq n_2 \wedge 0 \leq m_1 \wedge 0 \leq m_2 \wedge$
$\omega * onat(n_1) + onat(m_1) < \omega * onat(n_2) + onat(m_2) \rightarrow$
$n_1 < n_2 \vee (n_1 \doteq n_2 \wedge m_1 < m_2)$
taclet: `oltlexicographicInt`
146. $(1 < x \wedge 1 < y) \rightarrow (x + y) \leq x * y$ taclet: `oleqAddTimes`
147. $(0 < x \wedge lim(y)) \rightarrow lim(x * y)$ taclet: `olimtimes1, olimtimes1Q`
148. $(0 < y \wedge lim(x)) \rightarrow lim(x * y)$ taclet: `olimtimes2, olimtimes2Q`
149. $(x \neq 0 \wedge y < \omega) \rightarrow (x + y) * \omega \doteq x * \omega$ taclet: `Klaua26c1a`
150. $(x \neq 0 \wedge y < \omega \wedge lim(z)) \rightarrow (x + y) * z \doteq x * z$ taclet: `Klaua26c1`
151. $(x < \omega \wedge lim(z)) \rightarrow ((x \doteq 0 \wedge x * z \doteq 0) \vee (x \neq 0 \wedge x * z \doteq z))$ taclet: `otimesNlimit`

**Fig. 13.** Lemmas on multiplication

152. $x^1 \doteq x$ taclet: `oexpOne`

**Fig. 14.** Lemmas on exponentiation

153. $y \neq 0 \to \exists z (y * z \leq x \wedge x < y * (z + 1))$ taclet: `oleastMultiple`
154. $y \neq 0 \to \exists d, r (x \doteq y * d + r \wedge r < y)$ taclet: `odivQ`
155. $(y \neq 0 \wedge r_1 < y \wedge r_2 < y \wedge y * d_1 + r_1 \doteq y * d_2 + r_2) \to (d_1 \doteq d_2 \wedge r_1 \doteq r_2)$ taclet: `odivUnique`
156. $lim(y) \to \exists x (y \doteq \omega * x)$ taclet: `odivLim`

**Fig. 15.** Lemmas on decomposition

## 7 Well-ordering Pairs of Ordinals

157. $(v_1, v_2) \ll (v_3, v_4) \leftrightarrow$ Def. of $\ll$
$max(v_1, v_2) < max(v_3, v_4) \vee$
$max(v_1, v_2) = max(v_3, v_4) \wedge v_2 < v_4 \vee$
$max(v_1, v_2) = max(v_3, v_4) \wedge v_2 = v_4 \wedge v_1 < v_3$
taclets: `oltp_DefAxiom, oltp_Def`

158. $(0, 0) \ll (1, 0)$ taclet `oltpLess10`

159. $(0, 0) \ll (0, 1)$ taclet `oltpLess011`

160. $(1, 0) \ll (0, 1)$ taclet `oltpLess012`

161. $(0, x) \ll (0, 1) \to x \doteq 0$ taclet `oltpLess013`

162. $(x, y) \ll (1, 0) \to (x \doteq 0 \wedge y \doteq 0)$ taclet `oltpOne`

163. $(x, y) \ll (0, 1) \to ((x \doteq 0 \wedge y \doteq 0) \vee (x \doteq 1 \wedge y \doteq 0))$ taclet `oltpTwo`

164. $\forall v_1 \forall v_2 (\neg (v_1, v_2) \ll (v_1, v_2))$ taclets: `oltp_irr`

165. $\forall v_1 \forall v_2 \forall v_3 \forall v_4 \forall v_5 \forall v_5 ($
$(v_1, v_2) \ll (v_3, v_4) \wedge (v_3, v_4) \ll (v_5, v_6) \to (v_1, v_2) \ll (v_5, v_6))$
taclets: `oltp_transQ, oltp_trans`

166. $\forall v_1 \forall v_2 \forall v_3 \forall v_4 ((v_1, v_2) \ll (v_3, v_4) \vee (v_3, v_) \ll (v_1, v_2) \vee (v_1 \doteq v_3 \wedge v_2 \doteq v_4))$
taclet: `oltp_totalAxiom`

167. $(t_1, t_2) \ll (t_3, t_2) \leftrightarrow t_1 < t_3$ taclet: `oltp_same2`

168. $(t_1, t_2) \ll (t_1, t_3) \leftrightarrow t_2 < t_3$ taclet: `oltp_same1`

169. $\forall v_1 \forall v_2 ((v_1, v_2) \ll (v_1 + 1, v_2)$ taclet: `oltp_addOneL`

170. $\forall v_1 \forall v_2 ((v_1, v_2) \ll (v_1, v_2 + 1)$ taclet: `oltp_addOneR`

171. $\exists v_1 \exists v_2 \phi(v_1, v_2) \to \exists v_1 \exists v_2 (\phi(v_1, v_2) \wedge \forall w_1 \forall w_2; (w_1, w_2) \ll (v_1, v_2) \to \neg \phi(w_1, w_2))$
taclet: `oltpLeastPair2, oltpLeastPair`

172. $\forall v_1, v_2 (\forall v_3, v_4; ((v_3, v_4) \ll (v_1, v_2) \to \phi(v_3, v_4)) \to \phi(v_1, v_2)) \quad \to \forall v_1 \forall v_2 \phi(v_1, v_2)$
taclet: `oltpInduction`

**Fig. 16.** Definition and fundamental consequences of $\ll$

173. $\forall v_1, v_2, w_1, w_2(succp(v_1, v_2, w_1, w_2) \leftrightarrow (v_1, v_2) \ll (w_1, w_2) \wedge$
$\forall v_3, v_4((v_1, v_2) \ll (v_3, v_4) \rightarrow$
$(w_1, w_2) = (v_3, v_4) \vee (w_1, w_2) \ll (v_3, v_4)))$
taclet: `succp_Def`

174. $\forall v_1, v_2, w_1, w_2(succp(v_1, v_2, w_1, w_2) \rightarrow \forall v_3, v_4((v_3, v_4) \ll (w_2, w_3) \rightarrow$
$(v_1, v_2) = (v_3, v_4) \vee (v_3, v_4) \ll (w_1, w_2)$
taclet: `succpConseq`

175. $\forall v_1, v_2, v_3(succp(v_1, v_1, v_1 + 1, 0))$ taclets: `oltpSuccEq2`, `oltpSuccEq`

176. $\forall v_1, v_2(v_2 + 1 < v_1 \rightarrow succp(v_1, v_2, v_1, v_2 + 1)$
taclets: `oltpSuccMaxFirst`, `oltpSuccMaxFirstA`

177. $\forall v_1, v_2; (v_2 + 1 = v_1 \rightarrow succp(v_1, v_2, 0, v_2 + 1))$
taclets: `oltpSuccMaxFirst2`, `oltpSuccMaxFirst2A`

178. $\forall v_1, v_2(v_1 < v_2 \rightarrow succp(v_1, v_2, v_1 + 1, v_2))$
taclets: `oltpSuccMaxSecond`, `oltpSuccMaxSecond2`

179. $succp(v1, v1, w1, w2) \rightarrow w1 \doteq v1 + 1 \wedge w2 \doteq 0$
taclets: `succpConseqEqQ, succpConseqEq`

180. $(v2 + 1 < v1 \wedge succp(v1, v2, w1, w2) \rightarrow w1 \doteq v1 \wedge w2 \doteq v2 + 1$
taclets: `succpConseqLessRP, succpConseqLessRPQ`

181. $succp(v2 + 1, v2, w1, w2) \rightarrow w1 \doteq 0 \wedge w2 \doteq v2 + 1$
taclets: `succpConseqLessR, succpConseqLessRQ`

182. $(v1 < v2 \wedge succp(v1, v2, w1, w2)) \rightarrow w1 \doteq v1 + 1 \wedge w2 \doteq v2$
taclets: `succpConseqGreater, succpConseqGreaterQ`

183. $\forall v_1, v_2 \exists w_1, w_2(succp(v_1, v_2, w_1, w_2))$ taclets: `succpExists`

184. $succp(v_1, v_2, w_1, w_2) \rightarrow max(w_1, w_2) \leq max(v_1, v_2) + 1$ taclets: `succpOmax`

**Fig. 17.** Successor pairs in the well-ordering $\ll$

185. $limp(t_1, t_2) \leftrightarrow \forall x, y((x, y) \ll (t_1, t_2) \rightarrow \exists u, w((x, y) \ll (u, w) \wedge (u, w) \ll (t_1, t_2)))$
taclet `enum:limp_Def`

186. $limp(x, y) \leftrightarrow \neg\exists u, w(succp(u, w, x, y))$ taclet `limp_DefAlt`

187. $limp(x_1, x_2) \wedge (y_1, y_2) \ll (x_1, x_2) \wedge succp(y_1, y_2, z_1, z_2) \rightarrow (z_1, z_2)) \ll (x_1, x_2)$
taclet `limp_SuccLess`

188. $limp(0, 0)$ taclet: `oltpLimZeroZero`

189. $succp(x, y, u, w) \rightarrow \neg limp(u, w)$ taclet: `limpSuccFalse`

190. $\forall v_1, v_2(lim(v_2) \wedge v_2 \leq v_1 \rightarrow limp(v_1, v_2))$ taclet: `oltpLimR`

191. $\forall v_1, v_2(lim(v_1) \wedge v_1 \leq v_2 \rightarrow limp(v_1, v_2 + 1))$ taclet: `oltpLimL`

192. $\forall v(lim(v) \rightarrow limp(0, v))$ taclet: `:oltpLimZeroR`

193. $\forall v(lim(v) \rightarrow limp(v, 0))$ taclet: `:oltpLimZeroL`

194. $\forall v_1, v_2(lim(v_1) \wedge lim(v_2) \rightarrow limp(v_1, v_2))$ taclet: `oltpLimLim`

195. $\forall v_1, v_2(limp(v_1, v_2) \rightarrow (v_1 = 0 \wedge v_2 = 0) \vee (v_1 = 0 \wedge lim(v_2)) \vee$
$(lim(v_1) \wedge v_2 = 0) \vee (lim(v_1) \wedge lim(v_2)) \vee$
$(v_2 \leq v_1 \wedge lim(v_2)) \vee (lim(v_1) \wedge \exists v_3(v_2 = v_3 + 1 \wedge v_1 < v_2)$
taclet: `limpConseq`

**Fig. 18.** Limit pairs in the well-ordering $\ll$

## 8 Coding Pairs of Ordinals

196. $\forall v_1, v_2( \phi(v_1, v_2) \to (\forall w_1, w_2(succp(v_1, v_2, w_1, w_2) \to \phi(w_1, w_2)))$
$\wedge$
$limp(v_1, v_2) \wedge \forall w_1, w_2((w_1, w_2) \ll (v_1, v_2) \to \phi(w_1, w_2)) \to \phi(v_1, v_2))$
$\to \quad \forall v_1, v_2 \phi(v_1, v_2)$
taclet: `oltpInd2`
197. $encode(0, 0)) \doteq 0$ taclet: `encodeZero`
198. $succp(v_1, v_2, w_1, w_2) \to encode(w_1, w_2) \doteq encode(v_1, v_2) + 1$ taclet: `encodeSucc`
199. $limp(v_1, v_2) \to (\forall w_1, w_2((w_1, w_2) \ll (v_1, v_2) \to encode(w_1, w_2) < encode(v_1, v_2))$
$\wedge$
$\forall x(\forall w_1, w_2((w_1, w_2) \ll (v_1, v_2) \to encode(w_1, w_2) < x)$
$\to encode(v_1, v_2) \leq x)$ *taclet* : `encodeLim`
200. $encode(1, 0) = 1 \wedge encode(0, 1) = 2 \ \wedge$
$encode(1, 1) = 3 \wedge encode(2, 0) = 4$
taclets: `encodeOne, encodeTwo, encodeThree, encodeFour`
201. $encode(x, y) \doteq 0 \to (x \doteq 0 \wedge y \doteq 0)$ taclets: `encodeZeroV`
202. $(v_1, v_2) \ll (w_1, w_2) \to encode(v_1, v_2) < encode(w_1, w_2)$ taclets: `encodeMonotone`
203. $((v_1, v_2) \ll (w_1, w_2) \vee (v_1, v_2) = (w_1, w_2)) \to encode(v_1, v_2) \leq encode(w_1, w_2)$
taclets: `encodeweakMonotone`
204. $encode(v_1, v_2) = encode(w_1, w_2) \to (v_1, v_2) = (w_1, w_2)$ taclets: `enum:encodeInj`
205. $\forall v_1, v_2(max(v_1, v_2) \leq encode(v_1, v_2))$ taclets: `encodeWeakIncreasing`
206. $\forall x, y(a + x \leq encode(a, x))$ taclet: `oaddEncode`
207. $\forall v_1, v_2, w_1, w_2; (encode(v_1, v_2) < encode(w_1, w_2) \to (v_1, v_2) \ll (w_1, w_2))$
taclet: `encodeoltpLess`
208. $\forall w \exists v_1, v_2(encode(v_1, v_2) = w)$ taclets: `encodeSurjective`

**Fig. 19.** Encoding pairs of ordinals

209. $\forall w(encode(decode1(w), decode2(w)) = w)$ taclets: `decodeDef`
210. $\forall v_1, v_2(decode1(encode(v_1, v_2)) = v_1)$ taclets: `decode1Id`
211. $\forall v_1, v_2(decode2(encode(v_1, v_2)) = v_2)$ taclets: `decode2Id`

**Fig. 20.** Decoding for pairs of ordinals

## 9 The Bounded $\mu$-Operator

212. $(\exists y(y < b \wedge i(y) \doteq 0) \rightarrow \quad i(omu_{x<b} i(x)) \doteq 0 \wedge \forall z(z < b) \rightarrow i(z) \neq 0)))$
$\wedge$
$(\neg \exists y(y < b \wedge i(y) \doteq 0) \rightarrow omu_{x<b} i(x) \doteq 0)$
taclet: `omuDef`
213. $omu_{x<0} i(x) \doteq 0$ taclet: `omuZero`
214. $(i(c) \doteq 0 \wedge c < b) \rightarrow i(omu_{x<b} i(x)) \doteq 0 \wedge i(omu_{x<b} i(x)) \leq c$ taclet: `omuTak1`
215. $omu_{x<b} i(x) \doteq 0 \vee (i(omu_{x<b} i(x)) \doteq 0 \wedge omu_{x<b} i(x) < b)$ taclet: `omuTak2`

**Fig. 21.** The bounded $\mu$-operator

## References


1. P. H. Schmitt. Takeuti's first-order theory of ordinals revisited. Technical Report 2, Department of Informatics, Karlsruhe Institute of Technology, 2018.