

Small step Semantik

Beispiel zur λ -Abstraktion und zu bedingten Ausdrücken: Absolutbetrag

```
let val abs =  $\lambda x:\text{int. if } x \geq 0 \text{ then } x \text{ else } \sim x$ 
in (abs  $\sim 4$ , abs 3)
end
```

steht für

```
let val abs =  $\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x$ 
in (abs  $\sim 4$ , abs 3)
end
```

- $((\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) \sim 4, (\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) 3)$
mit Regel (LET-EXEC)
- $(\text{if } \geq (\sim 4, 0) \text{ then } \sim 4 \text{ else } \sim \sim 4, (\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) 3)$
mit Regel (BETA-V)
- $(\text{if } \text{false} \text{ then } \sim 4 \text{ else } \sim \sim 4, (\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) 3)$
mit Regel (OP)
- $(\sim \sim 4, (\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) 3)$
mit Regel (COND-FALSE)
- $(4, (\lambda x:\text{int. if } \geq (x, 0) \text{ then } x \text{ else } \sim x) 3)$
mit Regel (OP)
- $(4, \text{if } \geq (3, 0) \text{ then } 3 \text{ else } \sim 3)$
mit Regel (BETA-V)
- $(4, \text{if } \text{true} \text{ then } 3 \text{ else } \sim 3)$
mit Regel (OP)
- $(4, 3)$
mit Regel (COND-TRUE)