```
ArchiveEntry "01: Sequent Proof Example (Simple)"
Description "6.2.3: Propositional Proof Rules".

Definitions /* function symbols cannot change their value */
  Real v;   /* parentheses for nullary function symbols are optional: Real v(); */
  Real b;
End.

Problem
  v^2<=10 & b>0 -> b>0 & (!(v>=0) | v^2<=10)
End.

Tactic "01: Sequent Proof Example (Simple): Propositional proof"
  /* A simple proof with tactics from the Propositional menu  */
  implyR(1); andR(1); <(
    andL(-1); id,
    orR(1); andL(-1); id
  )
End.

End.

Exercise "02: Sequent Proof Example (Simple)"
Description "Exercise 6.3: Same propositional structure as 01: Sequent Proof Example (Simple). What
does that mean for the proof?".

Definitions
  Real a;
  Real c;
  Real x;
  Real y;
  Real z;
End.

Problem
  x^5=y^2+5 & a^2>c^2 -> a^2>c^2 & (!(z<x^2) | x^5=y^2+5)
End.

End.

ArchiveEntry "03: Proofs with Dynamics in Sequent Calculus"
Description "6.2.5: Proofs with Dynamics".

Definitions /* function symbols cannot change their value */
  Real b;
  Real v;
End.

ProgramVariables /* program variables may change their value over time */
  Real a;
  Real c;
End.

Problem
  [a:=-b; c:=10;](v^2<=10 & -a>0 -> b>0 & (!(v>=0) | v^2<=c))
End.

Tactic "03: Proof with Dynamics in Sequent Calculus: Simple Proof"
  composeb(1); assignb(1); assignb(1); implyR(1); andR(1); <(
    andL(-1); edit("b()>0", -2=="--b()>0"); id,
    orR(1); andL(-1); id
  )
End.

Tactic "03: Proof with Dynamics in Sequent Calculus: Proof with Lemma"
  composeb(1); assignb(1); assignb(1); edit("v()^2<=10&b()>0->b()>0&(!v()>=0|v()^2<=10)", 1);
  /* refer to already proved entries by their name */
  useLemma("01: Sequent Proof Example (Simple)")
```

```
End.

End.

ArchiveEntry "04: Short Bouncing Ball: single hop"
Description "6.4: A Sequent Proof for the Single-Hop Bouncing Ball".

Definitions      /* function symbols cannot change their value */
  Real H;        /* initial height */
  Real g;        /* gravity */
  Real c;        /* damping coefficient */
  Bool A()               <-> 0<=x & x=H & v=0 & g>0 & 1>=c&c>=0; /* initial conditions */
  Bool B(Real x, Real v) <-> 0<=x & x<=H;                        /* safety condition */
  HP   dyn               ::= {x'=v,v'=-g & x>=0};
End.

ProgramVariables /* program variables may change their value over time */
  Real x, v;     /* height and velocity */
End.

Problem
  A() -> [dyn;]B(x,v)
End.

Tactic "04: Short Bouncing Ball: single hop: Proof"
  implyR(1); solve(1); allR(1); implyR(1); implyR(1); allL("t_", -3); implyL(-3); <(
    simplify(2); closeTrue,
    andR(1); <(
      edit("(-g())*(t_^2/2)+v*t_+x>=0", 1); id,
      QE
    )
  )
End.

End.

ArchiveEntry "05: Arithmetic (1)"
Description "6.5: Real Arithmetic".

Definitions /* function symbols cannot change their value */
  Real a;
  Real b;
  Real x;
  Real y;
End.

Problem
  a>0 & b>0 -> (y>=0 -> a*x^2 + b*y >= 0)
End.

Tactic "05: Arithmetic (1): Proof"
  QE
End.

End.

ArchiveEntry "06: Arithmetic (2)"
Description "6.5: Real Arithmetic".

Definitions /* function symbols cannot change their value */
  Real x;
End.

Problem
  x^2>0 -> x>0
End.

Tactic "06: Arithmetic (2): Counterexample"
  QE /* false: Tools->Counterexample */
```

```
End.

End.

Exercise "07: Exercise Arithmetic"
Description "6.5: Real Arithmetic".

Definitions /* function symbols cannot change their value */
  Real x;
End.

/* Which subformula is not valid? */
Problem
    x^2>=0
  & (x>0 -> x^3>0)
  & (x>0 -> \exists y x^5*y^2>0)
  & (\forall a \forall b \exists x a*x+b=0)
End.

End.

ArchiveEntry "08: Quantifier Elimination After Universal Closure"
Description "6.5.1: Real Quantifier Elimination".

ProgramVariables /* program variables may change their value over time */
  Real x;
End.

Problem
  \forall d (d>=-x -> [x:=0; ++ x:=x+d;]x>=0)
End.

Tactic "08: Quantifier Elimination After Universal Closure: Proof"
  allR(1); choiceb(1.1); assignb(1.1.0); assignb(1.1.1); edit("\forall x \forall d (d>=-x-
>0>=0&x+d>=0)", 1); QE
End.

Tactic "08: Quantifier Elimination After Universal Closure: Proof in context of quantifier"
  chaseAt(1.0.1) ; edit("\forall d \forall x (d>=-x->0>=0&x+d>=0)", 1) ; QE
End.

End.

Exercise "09: Exercise Quantifier Elimination After Universal Closure"
Description "6.5.1: Real Quantifier Elimination".

ProgramVariables /* program variables may change their value over time */
  Real x;
End.

Problem
  x>=0 -> \exists d (d>=0 & [x:=0; ++ x:=x+d;]x>=0)
End.

End.


ArchiveEntry "10: Instantiating Real-Arithmetic Quantifiers"
Description "6.5.2: Instantiating Real-Arithmetic Quantifiers".

Definitions /* function symbols cannot change their value */
  Real t_0; /* names with optional index */
  Real x_0;
End.

ProgramVariables /* program variables may change their value over time */
  Real t;
  Real x;
End.
```

```
Problem
  x_0=x & t_0=t -> [{x'=-2, t'=1 & x>=0}](t-t_0 = (x_0-x)/2)
End.

Tactic "10: Instantiating Real-Arithmetic Quantifiers: Proof"
  implyR(1) ; solve(1) ; allR(1) ; implyR(1) ; implyR(1) ; allL("t_", -3) ; implyL(-3) ; <(
    simplify(2) ; closeTrue,
    QE
  )
End.

End.

ArchiveEntry "11: Substituting Equations into Formulas"
Description "6.5.5: Substituting Equations into Formulas".

Definitions
  Real x;
  Real e;
  Bool p(Real x);
End.

Problem
  x=e & p(e) -> p(x)
End.

Tactic "11: Substituting Equations into Formulas (Succedent)"
  implyR(1) ; andL(-1) ; allL2R(-1) ; id
End.

Tactic "11: Substituting Equations into Formulas (Antecedent)"
  implyR(1) ; andL(-1) ; allR2L(-1) ; id
End.

End.

ArchiveEntry "12: Abbreviating Terms to Reduce Complexity"
Description "6.5.6: Abbreviating Terms to Reduce Complexity".

Definitions /* function symbols cannot change their value */
  Real a;
  Real d;
  Real t;
  Real v;
  Real x;
End.

Problem
  a>=0 & t>=0 & 0 <= a/2*t^2 + v*t + x & a/2*t^2 + v*t + x <= d & d<=8 -> a/2*t^2 + v*t + x <= 8
End.

Tactic "12: Abbreviating Terms to Reduce Complexity: Proof"
  implyR(1); andL('L)*; edit("abbrv((a()/2*t()^2+v()*t()+x(),z))<=8", 1);
hideL(-6=="z=a()/2*t()^2+v()*t()+x()");
  hideL(-1=="a()>=0"); hideL(-1=="t()>=0"); hideL(-1=="0<=z"); QE
End.

End.

ArchiveEntry "13: Cutting to Transform Questions"
Description "6.5.7: Creatively Cutting Real Arithmetic to Transform Questions".

Definitions /* function symbols cannot change their value */
  Real x;
  Real y;
  Bool p(Real x); /* an uninterpreted predicate with argument x */
End.
```

```
Problem
  (x-y)^2<=0 & p(y) -> p(x)
End.

Tactic "13: Cutting to Transform Questions: Proof"
  implyR(1); andL(-1); cut("x()=y()"); <(
    hideL(-1=="(x()-y())^2<=0"); allL2R(-2); id,
    QE
  )
End.

End.
```