Working with Automated Reasoning Tools – Higher-Order Practicum –

Christoph Benzmüller and Geoff Sutcliffe

SS08, Block Course at Saarland University, Germany

Example – Function Composition

If function F is increasing and function G is decreasing,
then $G \circ F$ is decreasing

Example – Function Composition

```
thf(fun_composition,definition,(
     fun_composition := (^ [F:($i>$i),G:($i>$i),X:$i]: (G @ (F @ X)))
    )).


thf(fun_increasing,definition,(
    fun_increasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing,definition,(
    fun_decreasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ Y)) @ (F @ X)))))
    )).


thf(thm,theorem,(
     ![F:($i>$i),G:($i>$i),SM:($i>($i>$o))]:
       (( (((fun_increasing @ F) @ SM)
          & ((fun_decreasing @ G) @ SM))
        => ((fun_decreasing @ ((fun_composition @ F) @ G)) @ SM)))
   )).
```

Example – Function Composition

```
thf(fun_composition,definition,(
     fun_composition := (^ [F:($i>$i),G:($i>$i),X:$i]: (G @ (F @ X)))
    )).


thf(fun_increasing,definition,(
    fun_increasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing,definition,(
    fun_decreasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ Y)) @ (F @ X)))))
    )).


thf(thm,theorem,(
     ![F:($i>$i),G:($i>$i),SM:($i>($i>$o))]:
       (( (((fun_increasing @ F) @ SM)
          & ((fun_decreasing @ G) @ SM))
        => ((fun_decreasing @ ((fun_composition @ F) @ G)) @ SM)))
   )).
```

Example – Function Composition

```
thf(fun_composition,definition,(
     fun_composition := (^ [F:($i>$i),G:($i>$i),X:$i]: (G @ (F @ X)))
    )).


thf(fun_increasing,definition,(
    fun_increasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing,definition,(
    fun_decreasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ Y)) @ (F @ X)))))
    )).


thf(thm,theorem,(
     ![F:($i>$i),G:($i>$i),SM:($i>($i>$o))]:
       (( (((fun_increasing @ F) @ SM)
          & ((fun_decreasing @ G) @ SM))
        => ((fun_decreasing @ ((fun_composition @ F) @ G)) @ SM)))
   )).
```

Example – Function Composition

```
thf(fun_composition,definition,(
     fun_composition := (^ [F:($i>$i),G:($i>$i),X:$i]: (G @ (F @ X)))
    )).


thf(fun_increasing,definition,(
    fun_increasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing,definition,(
    fun_decreasing :=
      (^[F:($i>$i),SMALLER:($i>($i>$o))]:
         (![X:$i,Y:$i]: (((SMALLER @ X) @ Y)
                         =>
                         ((SMALLER @ (F @ Y)) @ (F @ X)))))
    )).


thf(thm,theorem,(
     ![F:($i>$i),G:($i>$i),SM:($i>($i>$o))]:
       (( (((fun_increasing @ F) @ SM)
          & ((fun_decreasing @ G) @ SM))
        => ((fun_decreasing @ ((fun_composition @ F) @ G)) @ SM)))
   )).
```

Example – Function Composition (b)

```
thf(a,type,(a:$tType)).
thf(b,type,(b:$tType)).
thf(c,type,(c:$tType)).


thf(fun_composition,definition,(
     fun_composition := (^ [F:(a>b),G:(b>c),X:a]: (G @ (F @ X)))
    )).


thf(fun_increasing_ab,definition,(
    fun_increasing_ab :=
      (^[F:(a>b),SMALLER1:(a>(a>$o)),SMALLER2:(b>(b>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing_bc,definition,(
    fun_decreasing_bc :=
      (^[F:(b>c),SMALLER1:(b>(b>$o)),SMALLER2:(c>(c>$o))]:
         (![X:b,Y:b]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).
```

Example – Function Composition (b)

```
thf(a,type,(a:$tType)).
thf(b,type,(b:$tType)).
thf(c,type,(c:$tType)).


thf(fun_composition,definition,(
     fun_composition := (^ [F:(a>b),G:(b>c),X:a]: (G @ (F @ X)))
    )).


thf(fun_increasing_ab,definition,(
    fun_increasing_ab :=
      (^[F:(a>b),SMALLER1:(a>(a>$o)),SMALLER2:(b>(b>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing_bc,definition,(
    fun_decreasing_bc :=
      (^[F:(b>c),SMALLER1:(b>(b>$o)),SMALLER2:(c>(c>$o))]:
         (![X:b,Y:b]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).
```

Example – Function Composition (b)

```
thf(a,type,(a:$tType)).
thf(b,type,(b:$tType)).
thf(c,type,(c:$tType)).


thf(fun_composition,definition,(
     fun_composition := (^ [F:(a>b),G:(b>c),X:a]: (G @ (F @ X)))
    )).


thf(fun_increasing_ab,definition,(
    fun_increasing_ab :=
      (^[F:(a>b),SMALLER1:(a>(a>$o)),SMALLER2:(b>(b>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                        =>
                        ((SMALLER2 @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing_bc,definition,(
    fun_decreasing_bc :=
      (^[F:(b>c),SMALLER1:(b>(b>$o)),SMALLER2:(c>(c>$o))]:
         (![X:b,Y:b]: (((SMALLER1 @ X) @ Y)
                        =>
                        ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).
```

Example – Function Composition (b)

```
thf(a,type,(a:$tType)).
thf(b,type,(b:$tType)).
thf(c,type,(c:$tType)).


thf(fun_composition,definition,(
     fun_composition := (^ [F:(a>b),G:(b>c),X:a]: (G @ (F @ X)))
    )).


thf(fun_increasing_ab,definition,(
    fun_increasing_ab :=
      (^[F:(a>b),SMALLER1:(a>(a>$o)),SMALLER2:(b>(b>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ X)) @ (F @ Y)))))
    )).


thf(fun_decreasing_bc,definition,(
    fun_decreasing_bc :=
      (^[F:(b>c),SMALLER1:(b>(b>$o)),SMALLER2:(c>(c>$o))]:
         (![X:b,Y:b]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).
```

Example – Function Composition (b)

```
thf(fun_decreasing_ac,definition,(
    fun_decreasing_ac :=
      (^[F:(a>c),SMALLER1:(a>(a>$o)),SMALLER2:(c>(c>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                          =>
                          ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).
```

```
thf(thm,theorem,(
     ![F:(a>b),G:(b>c),
         L1:(a>(a>$o)),L2:(b>(b>$o)),L3:(c>(c>$o))]:
       (( (((fun_increasing_ab @ F) @ L1) @ L2)
          & (((fun_decreasing_bc @ G) @ L2) @ L3))
        => (((fun_decreasing_ac @ ((fun_composition @ F) @ G)) @ L1) @ L3))
   )).
```

Example – Function Composition (b)

```
thf(fun_decreasing_ac,definition,(
    fun_decreasing_ac :=
      (^[F:(a>c),SMALLER1:(a>(a>$o)),SMALLER2:(c>(c>$o))]:
         (![X:a,Y:a]: (((SMALLER1 @ X) @ Y)
                       =>
                       ((SMALLER2 @ (F @ Y)) @ (F @ X)))))
    )).


thf(thm,theorem,(
     ![F:(a>b),G:(b>c),
         L1:(a>(a>$o)),L2:(b>(b>$o)),L3:(c>(c>$o))]:
       (( (((fun_increasing_ab @ F) @ L1) @ L2)
          & (((fun_decreasing_bc @ G) @ L2) @ L3))
        => (((fun_decreasing_ac @ ((fun_composition @ F) @ G)) @ L1) @ L3))
   )).
```

Example – Knights and Knaves

A very special island is inhabited only by knights and knaves.
Knights always tell the truth, and knaves always lie.
You meet two inhabitants: Zoey and Mel.
Zoey tells you that Mel is a knave.
Mel says, 'Neither Zoey nor I are knaves.'
Can you determine who is a knight and who is a knave?

Example – Knights and Knaves (variant lucas)

```
%----------------------------------------------------------------------
%----A very special island is inhabited only by knights and knaves.
thf(kk_6_1,axiom,(
    ! [X: $i] :
     ( ( is_a @ X @ knight )
        <~> ( is_a @ X @ knave ) ) )).


%----Knights always tell the truth.
thf(kk_6_2,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knight )
      => ( ! [A: $o] :
             ( says @ X @ A )
        => A ) ) )).


%-----Knaves always lie.
thf(kk_6_3,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knave )
      => ( ! [A: $o] : ( says @ X @ A )
        => ~ A ) ) )).
```

Example – Knights and Knaves (variant lucas)

```
%----------------------------------------------------------------------
%----A very special island is inhabited only by knights and knaves.
thf(kk_6_1,axiom,(
    ! [X: $i] :
     ( ( is_a @ X @ knight )
        <~> ( is_a @ X @ knave ) ) )).


%----Knights always tell the truth.
thf(kk_6_2,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knight )
      => ( ! [A: $o] :
             ( says @ X @ A )
        => A ) ) )).


%-----Knaves always lie.
thf(kk_6_3,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knave )
      => ( ! [A: $o] : ( says @ X @ A )
        => ~ A ) ) )).
```

Example – Knights and Knaves (variant lucas)

```
%----------------------------------------------------------------------
%----A very special island is inhabited only by knights and knaves.
thf(kk_6_1,axiom,(
    ! [X: $i] :
     ( ( is_a @ X @ knight )
        <~> ( is_a @ X @ knave ) ) )).


%----Knights always tell the truth.
thf(kk_6_2,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knight )
      => ( ! [A: $o] :
             ( says @ X @ A )
        => A ) ) )).


%-----Knaves always lie.
thf(kk_6_3,axiom,(
     ! [X: $i] :
       ( ( is_a @ X @ knave )
      => ( ! [A: $o] : ( says @ X @ A )
        => ~ A ) ) )).
```

Example – Knights and Knaves (variant lucas)

```
%----Zoey tells you that Mel is a knave.
thf(kk_6_5,axiom,
    ( says @ zoey @ ( is_a @ mel @ knave ) )).


%----Mel says, 'Neither Zoey nor I are knaves.'
thf(kk_6_6,axiom,
    ( says @ mel
    @ ~ ( ( is_a @ zoey @ knave )
        | ( is_a @ mel @ knave ) ) )).


%----Can you determine who is a knight and who is a knave?
thf(query,theorem,(
    ? [Y: $i,Z: $i] :
      ( ( is_a @ mel @ Y )
      & ( is_a @ zoey @ Z ) ) )).
%----------------------------------------------------------------------
```

Example – Knights and Knaves (variant lucas)

```
%----Zoey tells you that Mel is a knave.
thf(kk_6_5,axiom,
    ( says @ zoey @ ( is_a @ mel @ knave ) )).


%----Mel says, 'Neither Zoey nor I are knaves.'
thf(kk_6_6,axiom,
    ( says @ mel
    @ ~ ( ( is_a @ zoey @ knave )
        | ( is_a @ mel @ knave ) ) )).


%----Can you determine who is a knight and who is a knave?
thf(query,theorem,(
    ? [Y: $i,Z: $i] :
      ( ( is_a @ mel @ Y )
      & ( is_a @ zoey @ Z ) ) )).
%----------------------------------------------------------------------
```

Example – Knights and Knaves (variant lucas)

```
%----Zoey tells you that Mel is a knave.
thf(kk_6_5,axiom,
    ( says @ zoey @ ( is_a @ mel @ knave ) )).


%----Mel says, 'Neither Zoey nor I are knaves.'
thf(kk_6_6,axiom,
    ( says @ mel
    @ ~ ( ( is_a @ zoey @ knave )
        | ( is_a @ mel @ knave ) ) )).


%----Can you determine who is a knight and who is a knave?
thf(query,theorem,(
    ? [Y: $i,Z: $i] :
      ( ( is_a @ mel @ Y )
      & ( is_a @ zoey @ Z ) ) )).
%--------------------------------------------------------------------
```

Example – Knights and Knaves (variant chris)

```
%----------------------------------------------------------------------
%----A very special island is inhabited only by knights and knaves.
thf(kk_6_1,axiom,(
    ! [X: $i] :
      ( ( is_a @ X @ islander )
     => ( ( is_a @ X @ knight )
        | ( is_a @ X @ knave ) ) ) )).

%----Knights always tell the truth.
thf(kk_6_2,axiom,(
    ! [X: $i] :
      ( ( is_a @ X @ knight )
     => ( ! [A: $o] :
            ( says @ X @ A )
       => A ) ) )).

%-----Knaves always lie.
thf(kk_6_3,axiom,(
    ! [X: $i] :
      ( ( is_a @ X @ knave )
     => ( ! [A: $o] : ( says @ X @ A )
       => ~ A ) ) )).
```

Example – Knights and Knaves (variant chris)

```
%----You meet two inhabitants: Zoey and Mel.
thf(kk_6_4,axiom,
    ( ( is_a @ zoey @ islander )
    & ( is_a @ mel @ islander ) )).

%----Zoey tells you that Mel is a knave.
thf(kk_6_5,axiom,
    ( says @ zoey @ ( is_a @ mel @ knave ) )).

%----Mel says, 'Neither Zoey nor I are knaves.'
thf(kk_6_6,axiom,
    ( says @ mel
    @ ~ ( ( is_a @ zoey @ knave )
        | ( is_a @ mel @ knave ) ) )).

%----Can you determine who is a knight and who is a knave?
thf(query,theorem,(
    ? [Y: $i,Z: $i] :
      ( ( is_a @ Y @ knight )
      & ( is_a @ Z @ knave) ) )).
%----------------------------------------------------------------------
```