

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Liste*₁ *Element*

Liste → *Element*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Liste*₁ *Element*

Liste → *Element*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste*₁

Liste → *Element*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste*₁

Liste → *Element*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste'*

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste'*

Liste' →

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste'*

Liste' → *Element* *Liste'*₁

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Verbund → verbund ID ist *Liste* ende ID;

Liste → *Element* *Liste'*

Liste' → *Element* *Liste'*₁

Liste' → ε

Element → ID : *Typ* ;

Typ → zahl

Typ → bool

Liste → *Element* *Liste'*

Liste' → *Element* *Liste'*₁

Liste' → ε

Liste → *Element* *Liste'*
{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*;

Liste' → *Element* *Liste'*₁

Liste' → ε

Liste → *Element* *Liste'*
{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$

Liste' → *Element* *Liste'_1*

Liste' → ε

Liste → *Element* *Liste'*
{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0

Liste' → *Element* *Liste'_1*

Liste' → ε

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
 Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
 Element.start = 0
 Liste'.start = $f(\textit{Element.groesse}, 3)$ }

Liste' → *Element* *Liste'_1*
Liste' → ε

Liste → *Element Liste'*
{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

Liste' → *Element Liste'_1*
{ *Liste'.gepackt* = *Element.groesse* + *Liste'_1.gepackt*
Liste'.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'_1.ungepackt})$
Element.start = *Liste'.start*
Liste'_1.start = $f(\textit{Element.groesse}, \textit{Liste'.start})$ }

Liste' → ε
{ *Liste'.ungepackt* = 0
Liste'.gepackt = 0 }

Liste → *Element Liste'*

{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

Liste → *Element Liste'*

{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```
procedure Liste  
is
```

```
begin
```

```
    Element;
```

```
    Liste';
```

```
end Liste;
```

Liste → *Element Liste'*

{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = *f(Element.groesse, Liste'.ungepackt)*
Element.start = 0
Liste'.start = *f(Element.groesse, 3)* }

```
procedure Liste
is
    gepackt , ungepackt : Integer;

begin

    Element;

    Liste';

end Liste;
```

Liste → *Element Liste'*

{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```
procedure Liste
is
    gepackt, ungepackt : Integer;
    t1, t2, t3 : Integer;
begin

    Element;

    Liste';

end Liste;
```

Liste → *Element Liste'*
{ *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```
procedure Liste
is
    gepackt, ungepackt : Integer;
    t1, t2, t3 : Integer;
begin

    Element (start=>0, groesse=>t1);

    Liste';

end Liste;
```

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```

procedure Liste
is
  gepackt, ungepackt : Integer;
  t1, t2, t3 : Integer;
begin

  Element (start=>0, groesse=>t1);

  Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);

end Liste;
  
```

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```

procedure Liste
is
  gepackt, ungepackt : Integer;
  t1, t2, t3 : Integer;
begin

  Element (start=>0, groesse=>t1);

  Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);
  gepackt := t1 + t3;

end Liste;
  
```

Liste → *Element Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = $f(\textit{Element.groesse}, \textit{Liste'.ungepackt})$
Element.start = 0
Liste'.start = $f(\textit{Element.groesse}, 3)$ }

```

procedure Liste
is
  gepackt, ungepackt : Integer;
  t1, t2, t3 : Integer;
begin

  Element (start=>0, groesse=>t1);

  Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);
  gepackt := t1 + t3;
  ungepackt := f(t1, t2);
end Liste;
  
```

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = *f*(*Element.groesse*, *Liste'.ungepackt*)
Element.start = 0
Liste'.start = *f*(*Element.groesse*, 3) }

```

procedure Liste (gepackt, ungepackt : out Integer)
is
    t1, t2, t3 : Integer;
begin
    Element (start=>0, groesse=>t1);

    Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);
    gepackt := t1 + t3;
    ungepackt := f(t1, t2);
end Liste;
  
```

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = *f*(*Element.groesse*, *Liste'.ungepackt*)
Element.start = 0
Liste'.start = *f*(*Element.groesse*, 3) }

```

procedure Liste (gepackt, ungepackt : out Integer)
is
  t1, t2, t3 : Integer;
begin
  -- Element (start : in Integer, groesse : out Integer);
  Element (start=>0, groesse=>t1);

  Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);
  gepackt := t1 + t3;
  ungepackt := f(t1, t2);
end Liste;
  
```

Liste → *Element* *Liste'*
 { *Liste.gepackt* = *Element.groesse* + *Liste'.gepackt*
Liste.ungepackt = *f*(*Element.groesse*, *Liste'.ungepackt*)
Element.start = 0
Liste'.start = *f*(*Element.groesse*, 3) }

```

procedure Liste (gepackt, ungepackt : out Integer)
is
  t1, t2, t3 : Integer;
begin
  -- Element (start : in Integer, groesse : out Integer);
  Element (start=>0, groesse=>t1);
  -- Liste' (start           : in Integer,
  --         ungepackt, gepackt : out Integer);
  Liste' (start=>f(t1,3), ungepackt=>t2, gepackt=>t3);
  gepackt  := t1 + t3;
  ungepackt := f(t1, t2);
end Liste;
  
```