```
da-outlinks_wf  THM

⊢  ∀da:k:Knd fp-> Type, i:Id. da-outlinks(da;i) ∈ (IdLnk×Id×Type) List
| by Auto THEN Unfold `da-outlinks` 0
|    THEN
|    Inst
|    Thm* ∀T:Type, P:(T→𝔹), T':Type, f:({x:T| P(x) }→T'), L:T List.
|         mapfilter(f;P;L) ∈ T' List
|    {mapfilter_wf}
|    [lexpr{i'};{k:Knd| k ∈ dom(da) }]
|    THEN
|    BackThru -1
|    THEN
|    Reduce 0
|
.....subterm .....
1. da : k:Knd fp-> Type
2. i : Id
3. ∀P:({k:Knd| k ∈ dom(da) }→𝔹), T':Type, f:({x:{k:Knd| k ∈ dom(da) }
                                                | P(x) }
                                              →T'),
    L:{k:Knd| k ∈ dom(da) } List. mapfilter(f;P;L) ∈ T' List
4. k : {x:{k:Knd| k ∈ dom(da) }| has-src(i;x) }
⊢  k ∈ {k:Knd| k ∈ dom(da) & isrcv(k) }
  by DVar `k' THEN DVar `k'
     THEN
     RWO
     Thm* ∀i:Id, k:Knd. has-src(i;k) ⟺ isrcv(k) & source(lnk(k)) = i
     {assert_DASH_has_DASH_src}
     -1
     THEN
     ExRepD
END
```