

# Program Specialisation

## 1. Conjunctive Partial Deduction

Consider the program:

$\text{sumandproduct}(L,S,P) \leftarrow \text{sum}(L,0,S), \text{product}(L,1,P).$

$\text{sum}([],Acc,Acc) \leftarrow.$

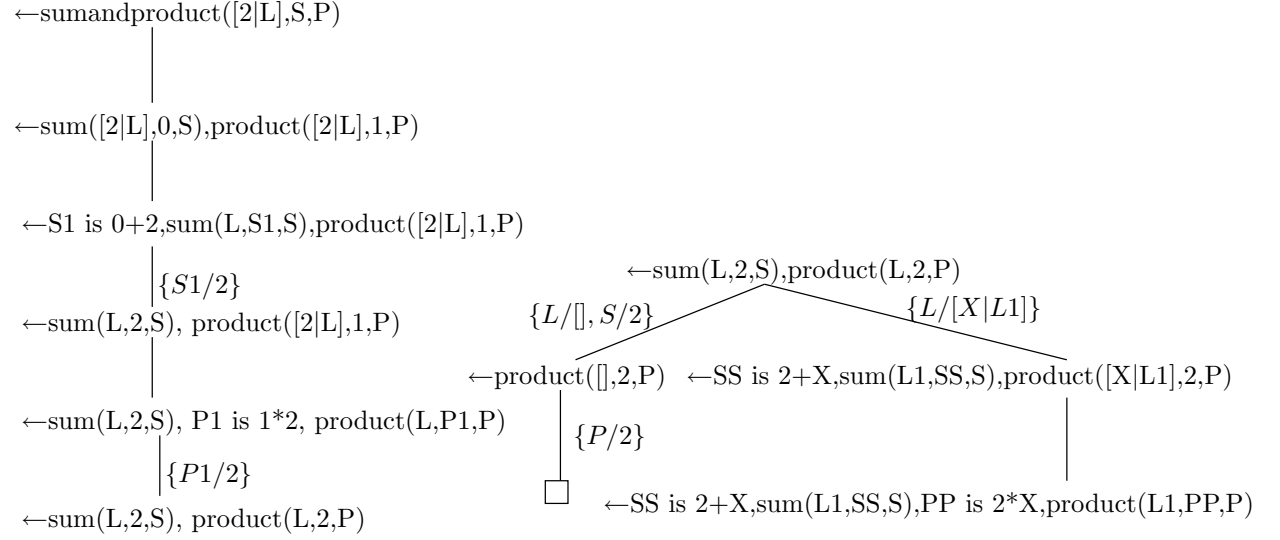
$\text{sum}([X|L],PS,S) \leftarrow PS1 \text{ is } PS+X, \text{sum}(L,PS1,S).$

$\text{product}([],Acc,Acc) \leftarrow.$

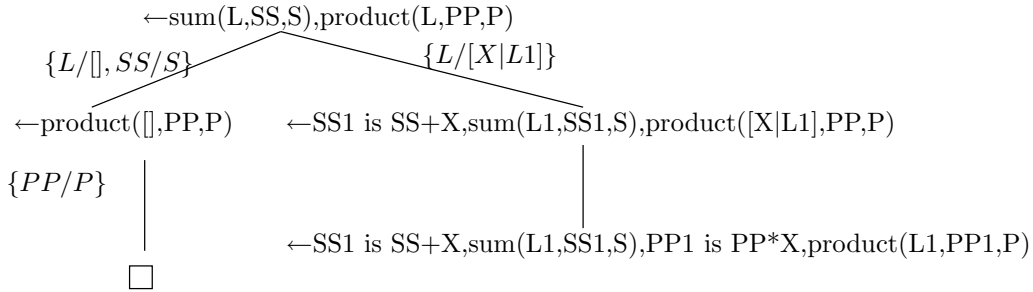
$\text{product}([X|L],PP,P) \leftarrow PP1 \text{ is } PP*X, \text{product}(L,PP1,P).$

Use conjunctive partial deduction to specialise this program for the query  $\leftarrow \text{sumandproduct}([2|L],S,P).$

First, *sum* and *product* are unfolded until their input (first arg) is consumed. Then, a tree is build for the conjunctive goal  $\leftarrow \text{sum}(L,2,S), \text{product}(L,2,P).$



$\text{sum}(L1,SS,S), \text{product}(L1,PP,P)$  is not an instance of the root, hence another tree has to be build:



The root of the third tree is more general than that of the second, it gives the same level of specialisation (same shape), hence one can discard the second tree.

Define  $\text{sum\_product}(L,SS,PP,S,P) \leftarrow \text{sum}(L,SS,S), \text{product}(L,PP,P).$ , extend the last tree with the root  $\leftarrow \text{sum\_product}(L,SS,PP,S,P)$  and use the definition to fold the leaves. Note that  $\text{sum}(L,2,S), \text{product}(L,2,P)$  is folded into  $\text{sum\_product}(L,2,2,S,P).$

$\text{sumandproduct}([2|L],S,P) \leftarrow \text{sum\_product}(L,2,2,S,P).$

$\text{sum\_product}([],S,P,S,P).$

$\text{sum\_product}([X|L],PS,PP,S,P) \leftarrow PS1=PS+X, PP1=PP*X, \text{sum\_product}(L,PS1,PP1,S,P).$