Disjunctive Completion Is Not “Optimal”

Helmut Seidl
FB IV - Informatik, University of Trier, Germany,
seidl@uni-trier.de

Christian Fecht
Department of Computer Science, Universität des Saarlandes,
Saarbrücken, Germany, fecht@cs.uni-sb.de

Enhancing ordinary analysis of Prolog with abstract domain \(D\) by using disjunctive completion \(P(D)\) instead of \(D\) [1] may gain precision, but still is at most as precise as abstract \(OLDT\)-resolution with \(D\) [2, 3]. While \(OLDT\) with \(P(D)\) gains nothing over \(OLDT\) with \(D\), we give an example which proves that ordinary analysis with \(P(D)\) may lose information against \(OLDT\) with \(D\) and hence cannot be “optimal”. Consider program:

\[
\begin{align*}
s(X, Y, Z) & \leftarrow a(X, Y, Z), p(X, Y, Z). & a(X, Y, Z) & \leftarrow X = Y. \\
p(X, Y, Z) & \leftarrow . & a(X, Y, Z) & \leftarrow Y = Z.
\end{align*}
\]

for query \(s(X, Y, Z)\) and assume we would like to compute \emph{pair-sharing} information with pair-sharing domain \(PS\) of Søndergaard [4]. The possible sharing arriving before the call to \(p\) is given between \(X\) and \(Y\) or \(Y\) and \(Z\), respectively. \(OLDT\)-resolution for the call to \(p\) propagates these two values separately and therefore returns just these values also for the program point behind the call to \(p\). On the contrary, ordinary analysis using \(P(PS)\), cannot propagate the two values separately. Instead, it \emph{combines} all possible return values for \(p\) with all possible values \emph{before} the call. Therefore, it introduces a further possible sharing, namely between \(X\) and \(Z\) – and thus loses precision.

References


