This newsletter extends Newsletter 122 by recording a few peephole optimisations applicable to sets used as iterators.

i. The construction

\[ (1) \quad (\forall x \in s-t) \text{ block;} \]

can be improved to the equivalent

\[ (1') \quad t_o = t; \ (\forall x \in s \mid s \notin t_o) \text{ block;} \]

Note that execution of (1') certainly involves no more work than execution of (1), and in addition avoids the formation of the set s-t together with the space allocation and garbage collection overhead which might be implied.

ii. Similarly,

\[ (2) \quad (\forall x \in s \ast t) \text{ block;} \]

can be improved to

\[ (2') \quad t_o = t; \ (\forall x \in s \mid x \notin t_o) \text{ block;} \]

or still better to

\[ (2'') \quad s_o = s; \ t_o = t; \]

\[ \text{if } (\# s_o) \not< t_o \text{ then } <s_o, t_o> = <t_o, s_o>; \]

\[ (\forall x \in s_o \mid x \notin t_o) \text{ block;} \]
iii. Again using the same idea,

\[(3) \quad (\forall x \in s + t) \text{ block;}
\]

can be improved to

\[(3') \quad s_o = s; \quad t_o = t;
(\forall x \in s_o) \text{ block;}
(\forall x \in t_o \mid x \notin s_o) \text{ block;}
\]