A. Index of defined identifiers

This appendix lists every nontrivial identifier used in the body of the paper. For each identifier, we list the page on which that identifier is defined or discussed—or when appropriate, the figure (with line number where possible). For those few identifiers not defined or discussed in text, we give the type signature and the page on which the identifier is first referred to.

Some identifiers used in the text are defined in the Haskell Prelude; for those readers less familiar with Haskell (possible even at the Haskell Symposium!), these identifiers are listed in Appendix C.

nodeToG defined on page 8.
NonLocal defined in Figure 2 on page 4.
normalizeGraph defined on page 9.
NothingO defined in Figure 2 on page 4.
O defined in Figure 2 on page 4.
OldFact defined in Figure 4 on page 5.
pairFwd defined on page 7.
PElem defined on page 6.
restart defined on page 8.
setFuel defined in Figure 4 on page 5.
s_exp let- or λ-bound in Figure 5 on page 8.
simp let- or λ-bound in Figure 5 on page 8.
simplify defined in Figure 5 on page 8.
s_node let- or λ-bound in Figure 5 on page 8.
SomeChange defined in Figure 4 on page 5.
stdMapJoin :: Ord k => JoinFun v -> JoinFun (Map.Map k v) not shown (but see page 7).
Store defined in Figure 1 on page 3.
successors defined in Figure 2 on page 4.
thenFwdRw defined on page 7.
Top defined on page 6.
Var defined on page 3.
varHasLit defined in Figure 5 on page 8.
WithBot defined on page 6.
WithTop defined on page 6.
WithTopAndBot defined on page 6.

B. Undefined identifiers

cnl (p4), Cond (Fig 5, p8), deepFwdRw (p7), f' (p7), fb (p10), fixpoint (p10), fl (Fig 5, p8), frewrite (p10), ft1 (p10), ft2 (p10), ftransfer (p10), fwdEntryFact (p10), fwdEntryLabel (p10), FwdRew (p5), Lit (Fig 5, p8), mapXX (p8), new (Fig 5, p8), noFwdRw (p5), old (Fig 5, p8), pass (p9), pass' (p10), prod (p8), rew (p10), rw (p5), shallowFwdRw (p12), ShapeLifter (p9), singletonDG (p10), Some (p11), tl (Fig 5, p8), toDg (p9), withFuel (p10).

C. Identifiers defined in Haskell Prelude or a standard library

!, $, &, &&, *, +, ++, -, ., /, =<<, ==, >, >=, >>, >>=, Bool, concatMap, const, curry, Data.Map, drop, False, flip, fmap, foldl, foldr, fst, head, id, Int, Integer, Just, last, liftM, map, Map.empty, Map.insert, Map.lookup, Map.Map, mapM_, Maybe, Monad, not, Nothing, otherwise, return, snd, String, tail, take, True, uncurry, undefined .