

## 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.

Add :: Operator not shown (but see page 7).  
addBlock :: NonLocal n => Block n C C -> LabelMap (Block n C C) -> LabelMap (Block n C C) not shown (but see page 4).  
analyzeAndRewriteFwdBody defined on page 5.  
arfGraph defined on page 9.  
Assign defined in Figure 1 on page 3.  
b1 let- or  $\lambda$ -bound on page 4.  
b2 let- or  $\lambda$ -bound on page 4.  
BCat defined in Figure 2 on page 4.  
BFirst defined in Figure 2 on page 4.  
Binop :: Operator -> Expr -> Expr -> Expr not shown (but see page 7).  
BLast defined in Figure 2 on page 4.  
Block defined in Figure 2 on page 4.  
block defined on page 10.  
blockmap let- or  $\lambda$ -bound on page 10.  
blocks let- or  $\lambda$ -bound on page 10.  
BMiddle defined in Figure 2 on page 4.  
BNode defined on page 13.  
body defined on page 10.  
Bot defined on page 6.  
Branch defined in Figure 1 on page 3.  
bs let- or  $\lambda$ -bound on page 4.  
bs1 let- or  $\lambda$ -bound on page 4.  
bs2 let- or  $\lambda$ -bound on page 4.  
Bwd defined on page 10.  
C defined in Figure 2 on page 4.  
cat defined on page 10.  
ChangeFlag defined in Figure 4 on page 5.  
Checkpoint defined on page 8.  
checkpoint defined on page 8.  
CkpointMonad defined on page 8.  
CondBranch defined in Figure 1 on page 3.  
ConstFact defined in Figure 5 on page 8.  
constFactAdd defined in Figure 5 on page 8.  
constLattice defined in Figure 5 on page 8.  
constProp defined in Figure 5 on page 8.  
constPropPass defined in Figure 5 on page 8.  
cp let- or  $\lambda$ -bound in Figure 5 on page 8.  
DataflowLattice defined in Figure 4 on page 5.  
DBlock defined on page 9.  
delFromFactBase :: FactBase f -> [(Label, f)] -> FactBase f not shown (but see page 14).  
DG defined on page 9.  
dgSplice defined on page 10.  
Direction defined on page 10.  
do\_block let- or  $\lambda$ -bound on page 10.  
elemFactBase :: Label -> FactBase f -> Bool not shown (but see page 14).  
elemLabelSet :: Label -> LabelSet -> Bool not shown (but see page 14).  
emptyLabelSet :: LabelSet not shown (but see page 14).  
entries let- or  $\lambda$ -bound on page 10.

entryFact let- or  $\lambda$ -bound on page 10.  
entryLabel defined in Figure 2 on page 4.  
ex let- or  $\lambda$ -bound in Figure 2 on page 4.  
Expr defined on page 3.  
extendFactBase :: FactBase f -> Label -> f -> FactBase f not shown (but see page 14).  
extendJoinDomain defined on page 6.  
extendLabelSet :: LabelSet -> Label -> LabelSet not shown (but see page 14).  
Fact defined in Figure 4 on page 5.  
FactBase defined in Figure 4 on page 5.  
factBaseLabels :: FactBase f -> [Label] not shown (but see page 14).  
factBaseList :: FactBase f -> [(Label, f)] not shown (but see page 14).  
fact\_bot defined in Figure 4 on page 5.  
fact\_join defined in Figure 4 on page 5.  
forwardBlockList defined on page 10.  
fp\_lattice defined in Figure 4 on page 5.  
fp\_rewrite defined in Figure 4 on page 5.  
fp\_transfer defined in Figure 4 on page 5.  
ft let- or  $\lambda$ -bound in Figure 5 on page 8.  
Fuel defined in Figure 4 on page 5.  
FuelMonad defined in Figure 4 on page 5.  
Fwd defined on page 10.  
FwdPass defined in Figure 4 on page 5.  
FwdRewrite defined in Figure 4 on page 5.  
FwdTransfer defined in Figure 4 on page 5.  
getFact defined on page 10.  
getFuel defined in Figure 4 on page 5.  
GMany defined in Figure 2 on page 4.  
GNil defined in Figure 2 on page 4.  
Graph defined in Figure 2 on page 4.  
graph defined on page 9.  
Graph' defined on page 9.  
gSplice defined on page 4.  
GUnit defined in Figure 2 on page 4.  
init\_fbase let- or  $\lambda$ -bound on page 10.  
iterFwdRw defined on page 7.  
Just0 defined in Figure 2 on page 4.  
Label defined in Figure 2 on page 4.  
LabelMap (a type) not shown (but see page 14).  
LabelSet (a type) not shown (but see page 14).  
lattice let- or  $\lambda$ -bound on page 10.  
lookup let- or  $\lambda$ -bound in Figure 5 on page 8.  
lookupFact :: FactBase f -> Label -> Maybe f not shown (but see page 14).  
mapEE defined on page 8.  
mapEN defined on page 8.  
mapUnion :: LabelMap a -> LabelMap a -> LabelMap a not shown (but see page 4).  
mapVE defined on page 8.  
mapVN defined on page 8.  
MaybeC (a type of kind \* -> \* -> \*) not shown (but see page 3).  
MaybeChange defined on page 8.  
Maybe0 defined in Figure 2 on page 4.  
mkFactBase :: [(Label, f)] -> FactBase f not shown (but see page 5).  
mkFRewrite defined in Figure 4 on page 5.  
mkFTransfer defined in Figure 4 on page 5.  
mkFTransfer3 defined on page 7.  
NewFact defined in Figure 4 on page 5.  
NoChange defined in Figure 4 on page 5.  
Node defined in Figure 1 on page 3.  
node let- or  $\lambda$ -bound in Figure 5 on page 8.

nodeToG defined on page 8.  
 NonLocal defined in Figure 2 on page 4.  
 normalizeGraph defined on page 9.  
 Nothing0 defined in Figure 2 on page 4.  
 0 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  $\lambda$ -bound in Figure 5 on page 8.  
 simp let- or  $\lambda$ -bound in Figure 5 on page 8.  
 simplify defined in Figure 5 on page 8.  
 s\_node let- or  $\lambda$ -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), f1 (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), t1 (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.