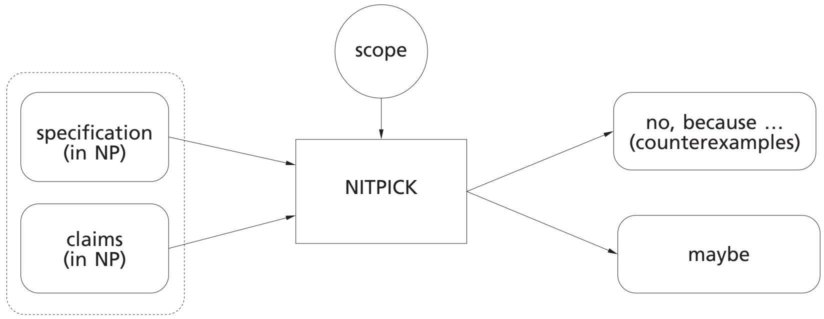

Figure 1: How Nitpick is used to check specifications

k	*#labelled*	*#unlabelled*
1	2	2
2	16	7
3	512	36
4	65536	317
5	3.4e7	5624
6	6.9e10	251610
7	5.6e14	33642660

Table 1: The number of $k \times k$ binary relations
with isomorphs included (first column)
and excluded (second column)

syntax
$s ::= d\,f \mid d\,s$
$d ::= v\colon \langle t, t\rangle$
$f ::= (e = e) \mid \neg f \mid f \wedge f$
$e ::= v \mid \varnothing \mid Id \mid Un \mid e\,;e \mid e \cup e \mid e \cap e \mid e \setminus e \mid e^{\sim} \mid e^{+}$
(s *in Spec*, d *in Decl*, f *in Formula*, e *in Term*, v *in Var*, t *in Type*)

typing rules
$e_1\colon \langle s, t\rangle, e_2\colon \langle t, u\rangle \vdash e_1\,;e_2\colon \langle s, u\rangle$
$e_1\colon \langle s, t\rangle, e_2\colon \langle s, t\rangle \vdash e_1 \circ e_2\colon \langle s, t\rangle$ ($\circ$ *is* $\cup, \cap, \setminus$)
$e\colon \langle s, t\rangle \vdash e^{\sim}\colon \langle t, s\rangle$
$e\colon \langle t, t\rangle \vdash e^{+}\colon \langle t, t\rangle$
$Id\colon \langle t, t\rangle, Un\colon \langle s, t\rangle, \varnothing\colon \langle s, t\rangle$

semantics of terms
$\#[\![\#]\!]\colon Interpretation \times Term \rightarrow \mathcal{P}\,(U \times U)$
$I\,[\![v]\!] = A[\![v]\!]$
$I\,[\![\varnothing]\!] = \{\}$
$I\,[\![Id\colon \langle s, s\rangle]\!] = \{(x, x) \mid x \in \tau[\![s]\!]\}$
$I\,[\![Un\colon \langle s, t\rangle]\!] = \tau[\![s]\!] \times \tau[\![t]\!]$

each operator o has a meaning O
$I\,[\![o(e_1, ..., e_n)]\!] = O\,(I\,[\![e_1]\!], ..., I\,[\![e_n]\!])$

meanings of operators
$p\,;q = \{(x, y) \mid \exists z.\ (x, z) \in p \wedge (z, y) \in q\}$
$p \cup q = \{(x, y) \mid (x, y) \in p \vee (x, y) \in q\}$
$p \cap q = \{(x, y) \mid (x, y) \in p \wedge (x, y) \in q\}$
$p \setminus q = \{(x, y) \mid (x, y) \in p \wedge (x, y) \notin q\}$
$p^{\sim} = \{(y, x) \mid (x, y) \in p\}$
$p^{+} = \bigcap \{q \mid p \subseteq q \wedge q\,;q \subseteq q\} = p \cup (p\,;p) \cup ...$

semantics of formulae
$\#[\![\#]\!]\colon Interpretation \times Formula \rightarrow Bool$
$I\,[\![s = t]\!] = (I\,[\![s]\!] = I\,[\![t]\!])$
$I\,[\![f \wedge g]\!] = I\,[\![f]\!] \wedge I\,[\![g]\!]$
$I\,[\![\neg f]\!] = \neg\, I\,[\![f]\!]$

Figure 2: Syntax, typing and semantics of a simple relational language

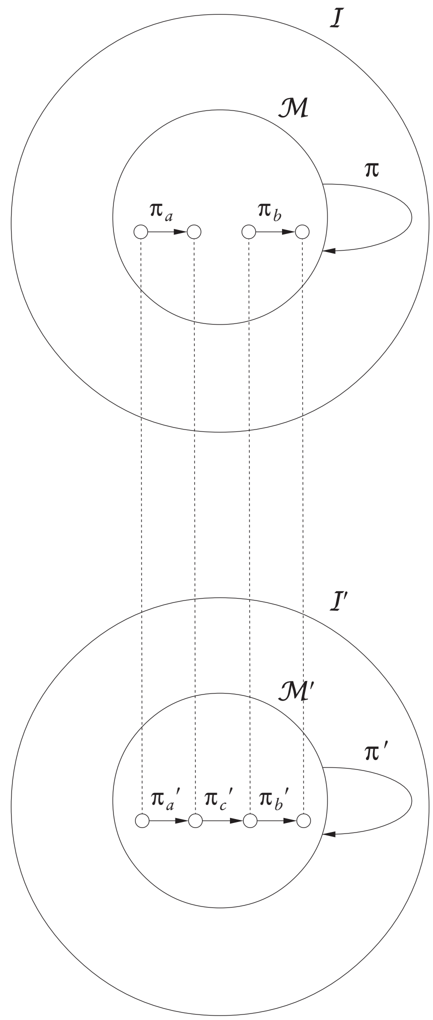

Figure 3: Interpretations, models and orbits

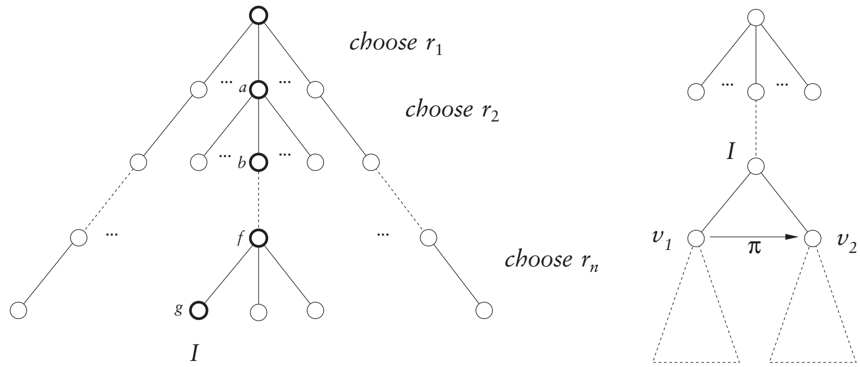

Figure 4 (left): Part of a full enumeration tree
Figure 5 (right): Isomorph elimination at an internal node

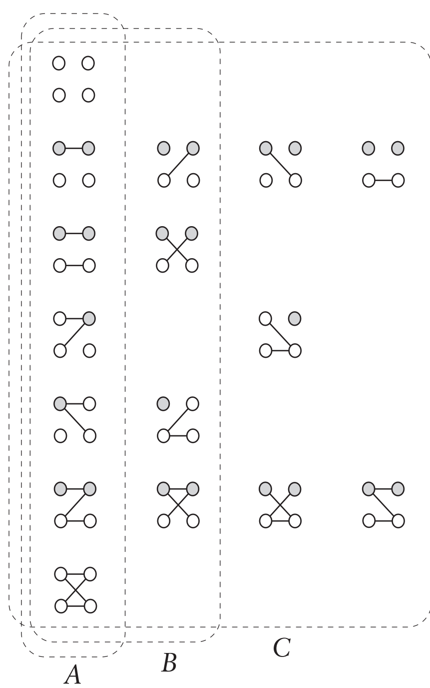

Figure 6: The 2× 2 relations

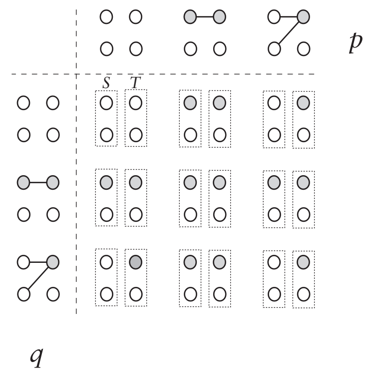

Figure 7: An example of how type colourings are obtained from relation colourings

```
fun enumerate (A, eq, i)
    if i > n
        if eval(F, A) then display (A)
        return
    s := left[i]
    t := right[i]
    forall (x, eql, eqr) in gen (tau[s], tau[t], eq[s], eq[t])
        A′ := (var [i], x) :: A
        eq [s] := eql
        eq [t] := eqr
        enumerate (A, eq, i + 1)
```

Figure 8: Basic Algorithm

```
fun enumerateR (A, leq, req, i)
    if i > n
        if eval(F, A) then display (A)
        return
    forall j in termvars [i]
        v := evalterm (term [j], A)
        leq [j], req [j] := colour (v)
    eql := coarsest
    eqr := coarsest
    forall j in relvars [i]
        if left [i] = left [j]
            eql := join (eql, leq [j])
        if right [i] = right [j]
            eqr := join (eqr, req [j])
    forall x in genR (tau [left[i]], tau [right [i]], leq, req)
        leq [i], req [i] := colour (x)
        A := (var [i], x) :: A
        enumerateR (A, leq, req, i + 1)
```

Figure 9: Refined Algorithm

$[\mathit{Phone}]$

$\mathit{Switch} = [\mathit{conns}\colon \mathit{Phone} \leftrightarrow \mathit{Phone}]$

$\mathit{Call}\ (\mathit{from}, \mathit{to}\colon \mathit{Phone}) =$
$\quad [\mathit{Switch} \mid \mathit{to} \notin \mathit{ran\ conns}\ \wedge\ \mathit{conns}' =\ \mathit{conns} \cup \{(\mathit{from}, \mathit{to})\}]$

$\mathit{OneCaller} = [\mathit{Switch} \mid \mathit{inj\ conns}]$

$\mathit{OneRole} = [\mathit{Switch} \mid \mathit{dom\ conns} \cap \mathit{ran\ conns} = \emptyset]$

$\mathit{Claim}1\ (\mathit{from}, \mathit{to}\colon \mathit{Phone}) :: \mathit{Call}\ (\mathit{from}, \mathit{to}) \wedge \mathit{OneCaller} \Rightarrow \mathit{OneCaller}'$

$\mathit{Claim}2\ (\mathit{from}, \mathit{to}\colon \mathit{Phone}) :: \mathit{Call}\ (\mathit{from}, \mathit{to}) \wedge \mathit{OneRole} \Rightarrow \mathit{OneRole}'$

Figure 10: Example specification

Figure 11a: An instance of the schema *OneRole.*
The graph represents the relation *Conns.*

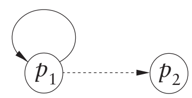

Figure 11b: An instance of the schema *Call*: *p1* is connected to itself, and calls *p2.*
The dotted arc is the new connection; the relation *Conns* is represented by the solid arc, and *Conns'* by both arcs. The variables *from* and *to* have the values *p1* and *p2* respectively.

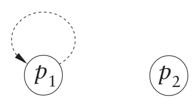

Figure 11c: A counterexample of *Claim2: p1* calls itself.

$[Phone, Number]$

$Switch = [$
$\quad Called: Phone \leftrightarrow Number$
$\quad Net: Number \rightarrow Phone$
$\quad Conns: Phone \leftrightarrow Phone$
$\quad |$
$\quad Conns = \ Called \ ; Net]$

$Call\ (from: Phone; to: Number) = [$
$\quad Switch$
$\quad |$
$\quad to \notin ran\ Called$
$\quad Called' = Called \cup \{(from, to)\}$
$\quad Net' = Net]$

$OneCaller = [Switch \mid inj\ conns]$

$OneRole = [Switch \mid dom\ conns \cap ran\ conns = \emptyset]$

$Claim1\ (from, to: Phone) :: Call\ (from, to) \wedge OneCaller \Rightarrow OneCaller'$

$Claim2\ (from, to: Phone) :: Call\ (from, to) \wedge OneRole \Rightarrow OneRole'$

Figure 12: An elaborated version of the example of Figure 10

scope	*# cases*	*Claim1*	*Claim2*
2	64	16 (4.0)	32 (2.0)
3	4608	168 (27)	692 (7)
4	1048576	2816 (372)	22944 (46)
5	8.39 E8	91986 (9.1e4)	1.43e6 (590)

Table 2: Performance for the example of Figure 10. The reductions are due entirely to isomorph elimination. A scope of *k* means that the enumeration was restricted to cases involving *k* phones or fewer ($\Sigma[\![Phone]\!] = k$). The second column gives the number of cases in the unreduced space. The others give, for the two claims checked, the number of cases, and the reduction factor (in parentheses), for an enumeration of the same space using isomorph elimination. Term symmetry has no effect for this example.

scope	*# cases*	*Claim1*	*Claim2*
2	576	80 (7.2)	144 (4)
		72 (8)	118 (4.9)
3	294,912	2,320 (127)	8,832 (33.4)
		2,248 (131)	8,014 (36.8)
4	6.55e8	130,864 (5e3)	1.29e6 (508)
		130,528 (5e3)	1.23e6 (533)
5	6.52e12	1.69e7 (3.9e5)	5.16e8 (1.3e4)
		1.68e7 (3.9e5)	5.01e8(1.3e4)

Table 3: Reductions for the elaborated example of Figure 12. A scope of *k* means that the enumeration was restricted to cases involving at most *k* phones and *k* numbers ($\Sigma[\![Phone]\!] = \Sigma[\![Number]\!] = k$). The first line for each scope gives the results with term symmetry switched off, the second line with term symmetry on.