Relational Logic

Alloy[86,87] is a first-order logic with relations and transitive closure. *Relation* is a fancy word for table; *tuple* is a fancy word for row. An intuition for transitive closure is 'where can we get to from here?'

```
sig Tree { root : Node }
sig Node {
    left, right : lone Node,
    value : one Int,
}
```

[86] Daniel Jackson. *Software Abstractions: Logic, Language, and Analysis*. The MIT Press, Cambridge, Mass., April 2006. ISBN 978-0-262-10114-1

[87] http://alloy.mit.edu

Figure 7: An Alloy model of a binary tree. An intuition, good to a first-order approximation, is to read this like class declarations in an object-oriented language. Going beyond this intuition, root, left, right, and value are really binary relations; see the example below.

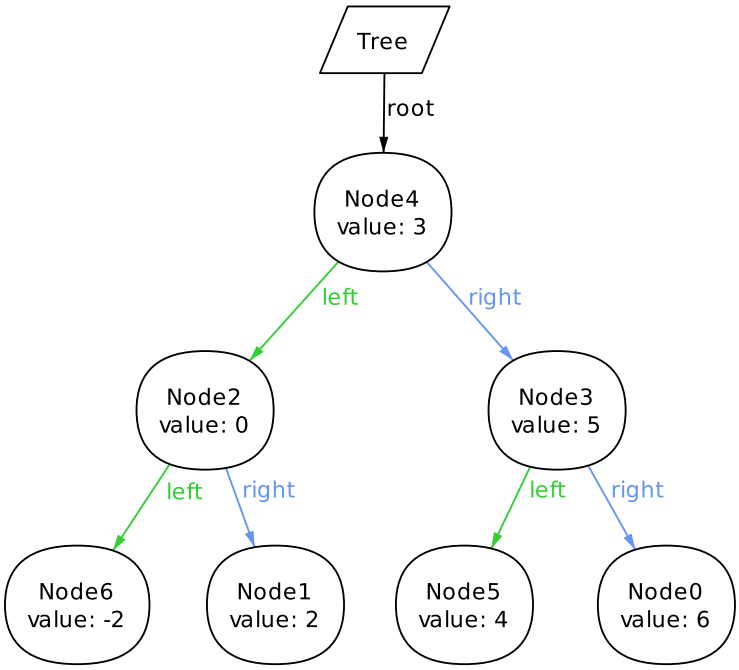

Figure 8: An instance of a binary tree. If we write the relations from this figure out in tabular form they look like this:

Tree
Treeo

root	
Treeo	Node4

Node
Nodeo
Node1
Node2
Node3
Node4
Node5
Node6

value	
Nodeo	6
Node1	2
Node2	0
Node3	5
Node4	3
Node5	4
Node6	-2

left	
Node4	Node2
Node2	Node6
Node3	Node5

right	
Node4	Node3
Node2	Node1
Node3	Nodeo

Int is the set of all integers, and iden is the identity relation (a binary relation that maps every atom to itself).

Int
-8
-7
⋮
6
7

iden	
Treeo	Treeo
Nodeo	Nodeo
Node1	Node1
Node2	Node2
Node3	Node3
Node4	Node4
Node5	Node5
Node6	Node6
-8	-8
-7	-7
⋮	⋮
6	6
7	7

union

left + right	
Node4	Node2
Node2	Node6
Node3	Node5
Node4	Node3
Node2	Node1
Node3	Nodeo

transitive closure

^left	
Node4	Node2
Node2	Node6
Node3	Node5
Node4	Node6

transitive closure

^right	
Node4	Node3
Node2	Node1
Node3	Nodeo
Node4	Nodeo

join

Treeo.root
Node4

root.^(left + right)	
Treeo	Nodeo
Treeo	Node1
Treeo	Node2
Treeo	Node3
Treeo	Node5
Treeo	Node6

Comprehension questions:

1. Why isn't Node4 included in this relation?
2. How could the expression be changed so that Node4 would be included?
3. What does the expression Treeo.root.^(left + right) evaluate to?

reflexive transitive closure: *r ≡ ^r + iden