

Modeling Sparse Matrices

```
assert transValid {  
  all e: ELL, c: CSR, m: Matrix |  
    repInv[e] and alpha[e, m] and ellcsr[e, c]  $\Rightarrow$   
    repInv[c] and alpha[c, m]  
}  
check transValid  
for 4 Int, 7 seq, 1 Matrix, 1 ELL, 1 CSR, 2 Value
```

Fig. 1. Verifying that the ELL to CSR translation is valid

I. MODELING SPARSE MATRIX FORMAT TRANSLATIONS

Why might one want to translate sparse matrix formats?

For a translation to be considered valid, the final state must preserve the representation invariant of the target format as well as represent the same abstract matrix as the original format. Intermediate states, however, do not need to preserve the representation invariant. The Alloy assertion used to model this relationship is shown in Fig. 1, which states that if the ELL matrix, e , is a valid representation of the abstract matrix, m , and the translation is applied, giving the CSR matrix, c , then it follows that c is a valid CSR matrix that represents the abstract matrix m .

The translation from ELL to CSR in Alloy is performed by quantifying the state of the algorithm for every step in the innermost loop. TODO: create an image that has a sequence of circles. Each circle has the state of the innermost loop, i.e., the values of i , k , and kp . Then talk about how the values of i , k , and kp are used to relate the structure of the two states using the relationships defined in the algorithm.