BLOCKS WORLD, DESCRIPTION

Blocks World is a simple exercise in generative specification. A specification is generative if there is enough information for a computer to implement the described world. The specification below has *not* been implemented, it is a first pass at knowledge engineering the Blocks World domain.

Domain:	{a, b, c, d}	are Blocks.
	{Table}	is the Table.
Variables:	x, y, z,...	represent single, unique objects from the domain.
Relations:	x On y	Block x is directly resting on Block y.
Terms:	X, Y, Z	are collections of On relations that specify a (true) configuration of Blocks and Table.

Example:

```
    [a]     is expressed as     {(a On b)
[b] [c]                          (b On T)
-------                          (c On T)}
```

The curly brackets indicate a term.

Constraints on Terms:

Every block is On something:	(x On y) or (x = T)
The Table is On nothing:	not(T On x)
Asymmetric:	if (x On y) then not(y On x)

Not default:
not(x On y) means (x On y) cannot exist in term set

Get-rid-of-Table: make a binary relation into a unary predicate

(x On T) = (x OnT)

Functions: Move(x Onto y) maps terms onto terms

preconditions:	(y = T) or not(u On y)	y has empty top
	not(v On x)	x has empty top
action:	Remove(x On z)	Pickup x
	Add(x On y)	Putdown x
postconditions:	(z = T) or not(w On z)	z has empty top