Formal Definition			Type	Example / Meaning	%
Turn (			action	*used to change body/view orientation*	0.41
t	direction = [		list[1]		0.67
		Left	value	*turn left*	
		Right	value	*turn right*	
		Front	value	*turn front == go straight (inherited)*	
		Back	value	*turn back*	
	]				
t	face =				0.32
		Face()	action	*turn to face the red carpet*	
wt	location =				0.14
		Travel()	action	*at the stone floor, turn right*	
	view =		enum		0.06
		True (omit for False)	value	*turn only the view to verify something*	
t/wt	precond =		action		0.04
tt		Face()	action	*with your back to the wall, turn right*	
wt		Travel()	action	*when the hall ends, turn right*	
tw	postcond =				0.01
		Travel()	action	*go right onto the pink hallway*	
)					
Travel (			action	*used to move the agent in the environment*	0.64
w	until =				0.62
		Verify()	action	*go to the yellow-tiled hall*	
w	distance = [		list[1]		0.42
		Distance()	description	*go forward two segments*	
	]				
tw	face =				0.24
		Face()	action	*go down the grassy hallway*	
w	past =				0.05
		Verify()	action	*go forward, passing the chair*	
w	follow =				0.04
		Follow()	sub-task	*follow the yellow path*	
ww	location =				0.01
		Travel()	action	*at the lamp, go straight*	
)					
Distance (			description	*used to represent distances*	
	count =				0.88
		[0-9]+	integer	*walk forward twice*	
	distUnit =				0.08
		Verify()	action	*move one block*	
)					

Verify (			action	*used to compare descriptions and observations*	0.97
v	desc = [		list[N]		0.92
		Thing()	description	*when you come to a red brick path*	
	]				
w	goal =				0.06
		DeclareGoal()	sub-task	*that is your destination*	
w	dist =				0.01
		Distance()	description	*one segment before the chair*	
)					
Face (			action	*used to orient the agent wrt the environment*	0.34
t	faced =				1.00
		Verify()	action	*go towards the easel*	
t	direction = [		list[1]		0.28
		Left	value	*go straight, towards the chair*	
		Right	value	*face the hallway on your right*	
		Front	value	*face the hallway on your left*	
		Back	value	*face the hallway behind you*	
	]				
)					
Find (			action	*used to give non-localized instructions*	0.01
	until =				1.00
		Verify()	action	*at the intersection between two pink hallways*	
)					
Follow (			sub-task	*used to trace a path through the environment*	
	until =				0.63
		Verify()	action	*follow the brown floored hallway to the chair*	
)					
DeclareGoal (			sub-task	*used to confirm the destination*	
	cond =				0.94
		Travel()	action	*the office 254 is down this hallway*	
	goal = [		list[1]		0.88
		String	text	*the destination identifier, i.e. 'office 254'*	
	]				
)					
Thing (			description	*used to represent entities in the environment*	
	type =		enum		1.00
		TypeObj	value	*the chair*	
		TypePath	value	*the pink floor*	
		TypeStruct	value	*this hallway*	
		TypeRegion	value	*the flowered floored area*	

	value =			enum		1.00
		LinearStructure				
		P	Path	value	*the black hallway*	
		S	End	value	*the end of the blue carpet*	
		O	Wall	value	*the wall on your left*	
		Object				
		O	GenChair	value	*the chair*	
		O	Hatrack	value	*the hatrack*	
		O	Lamp	value	*the lamp*	
		O	Easel	value	*the easel*	
		O	Sofa	value	*the bench*	
		O	Barstool	value	*the stool*	
		O	Furniture	value	*between the two pieces of furniture*	
		AreaStructure				
		S	Intersection	value	*the intersection containing the easel*	
		S	Corner	value	*move to the corner*	
		S	DeadEnd	value	*make the first right into a dead end*	
		S	Block	value	*turn right and move one block*	
		Picture				
		O	Butterfly	value	*towards the butterflies on the wall*	
		O	Eiffel	value	*the towers on both sides of the walls*	
		O	Fish	value	*the yellow halls with fish on the walls*	
		O	Pic	value	*generic picture*	
		Texture				
		O	Rose	value		
		O	Wood	value		
		O	Grass	value		
		O	Cement	value		
		O	BlueTile	value		
		O	Brick	value		
		O	Stone	value		
		O	Honeycomb	value		
		O	Gray	value	*alias for [Cement, Stone]*	
		O	Greenish	value	*alias for [Grass, Honeycomb]*	
		O	Brown	value	*alias for [Brick, Wood]*	
		O	Dark	value	*alias for [Stone, BlueTile, Wood, Brick]*	
		O	Flooring	value	*generic flooring pattern*	
	dist =					0.82
		Immediate		value		
		Near		value		
		Far		value		
	side = [			list[1]		0.67
		Left		value	*the chair on your left*	
		Right		value	*the chair on your right*	

		Front	value	*the chair in front of you*	
		Back	value	*the chair behind you*	
		At	value	*go past the lamp (no side)*	
		Sides	value	*there should be butterfly images on the walls*	
	]				
	Appear = [		list[N]		0.33
		Rose	value		
		Wood	value		
		Grass	value		
		Cement	value		
		BlueTile	value		
		Brick	value		
		Stone	value		
		Honeycomb	value		
		Gray	value	*alias for [Cement, Stone]*	
		Greenish	value	*alias for [Grass, Honeycomb]*	
		Brown	value	*alias for [Brick, Wood]*	
		Dark	value	*alias for [Stone, BlueTile, Wood, Brick]*	
		Flooring	value	*generic flooring pattern*	
	]				
	Part = [		list[N]		0.14
		Thing()	description	*the intersection with a bare concrete hall*	
	]				
	Detail = [		list[N]		0.07
		Thing()	description	*the intersection containing the hatrack*	
	]				
	Structural = [		list[1]		0.02
		T_Int	value	*you will reach a 'T' intersection*	
		Long	value	*the longer end of the hallway*	
		Short	value	*the shorter end of the hall*	
	]				
	Order_adj = [		list[1]		0.02
		[0-9]+	integer	*move to the second alley*	
	]				
	On = [		list[N]		0.02
		Thing()	description	*the picture on the wall*	
	]				
	Count = [		list[1]		0.02
		[0-9]+	integer	*there will be two chairs*	
	]				
	Past = [		list[N]		0.01
		Thing()	description	*the lamp past the chair*	
	]				
)					