```
<"FSM.lib";
// ring var: just all bit-level inputs (PS and PI) followed by S and T
ring rr = (2,X), (n23,n43,n42,n18,n40,n39,n38,n8,n31,n13,n36,n35,n34,n30,n33,v6_4,n28,n27,
n26,n25,n24,n23_1,n22,n21,n20,n19,n18_1,n17,n16,v0,v1,v2,v3,v4,v5,P,S,T), lp;
minpoly = X^4+X+1;

ideal A_in = v2,v3,v4,v5;
poly def_S = v2+ v3*X+ v4*X^2+ v5*X^3+S;
ideal X_in = v0,v1;
poly def_X = v0 + v1*X^5+P;
poly  red_S  =  S^16+S;
poly red_T = T^16+T;
poly red_X = P^4 + P;
// red_all: all bit-level vars and red_S
ideal red_all = v0^2+v0, v1^2+v1, v2^2+v2, v3^2+v3, v4^2+v4, v5^2+v5,red_S,red_X;
poly tran = T+(X^2+X+1)*v0*v1*v3*v5+(X^3+X)*v0*v1*v3+(X+1)*v0*v1*v5
+(X^3+1)*v0*v3*v5+(X^3+X)*v0*v3+(X+1)*v0*v5+(X^2)*v0+(X^2+X+1)*v1*v3*v5+(X^3+X)*v1*v3+(X^2+X+1)*v1*
v5+(X)*v1+(X^3+1)*v3*v5+(X^3+X)*v3+(X^3+1)*v5+X^2;

poly init_S = S+X^2;
poly reached = T+X^2;

ideal I1 = preprocess(def_S, red_all, A_in);
poly unitran = conv_word(tran,I1);
I1 = preprocess(def_X, red_all, X_in);
unitran = conv_word(unitran,I1);

int i = 1;
ideal from_I,to_I,new_I;
from_I[1] = init_S;
while(1)
{
	i++;
	to_I[i] = transition(from_I[i-1],unitran,red_all);
	"Iteration #",i-2;
	"Next State(s): ",to_I[i];
	new_I[i] = redWord(to_I[i]+compl(reached,red_T), red_T);
	"Newly reached states: ",new_I[i];
	if ((redWord(new_I[i],red_T) == 1) or (i>25))
	{
		"*************** TERMINATE! ***************";
		break;
	}
	reached = redWord(reached * new_I[i],red_T);
	"Currently reached states: ",reached;
	from_I[i] = subst(new_I[i],T,S);
}
"BFS depth: ",i-2;
"Final reachable states: ",reached;
```