**Question 1**

1. Advantage: Can verify systems with more states, i.e. suffers less from the state explosion problem. Disadvantage: Are there any? I suppose implementation will be more difficult.

OBDD-based approaches for symbolic model checking require that the entire state space is computed first, which becomes infeasible when the state space gets to about 1030. Also, many of the operations that we have to run on OBDDs, for example *exists*, are not particularly efficient.

* 1. (M1, s0) ⊧ qUp

For all paths ρ = w0, w1, … where w0 = s0: ρ⊧ qUp

For all paths ρ = w0, w1, … where w0 = s0: ∃j >= 0: ρ^j ⊧ p and ∀0 <= k < j: ρ^j ⊧ q

True

Take j=0 for any path ρ, ρ ⊧ p (since s0 ∈ π(p)) and the for all quantification over k becomes empty and so is true trivially

* 1. (M1, s0) ⊧ FGt

For all paths ρ = w0, w1, … where w0 = s0: ρ⊧ FGt

For all paths ρ = w0, w1, … where w0 = s0: ∃i >= 0: ρ^i ⊧ Gt

For all paths ρ = w0, w1, … where w0 = s0: ∃i >= 0: ∀j >= 0: ρ^(i + j) ⊧ t

False

Take ρ = (s0 s3)^ω, t never holds in this path (since s0 ∉ π(t) and s3 ∉ π(t)) and so there doesn’t exist an i

* 1. (M1, s0) ⊧ GXXr

For all paths ρ = w0, w1, … where w0 = s0: ρ⊧ GXXr

For all paths ρ = w0, w1, … where w0 = s0: ∀i >= 0: ρ^i ⊧ XXr

For all paths ρ = w0, w1, … where w0 = s0: ∀i >= 0: ρ^(i+1) ⊧ Xr

For all paths ρ = w0, w1, … where w0 = s0: ∀i >= 0: ρ^(i+2) ⊧ r

False

Take ρ = (s0 s3)^ω, i = 0, ρ^2 ⊭ r (since s0 ∉ π(r))

* 1. (M1, s0) ⊧ AFt

For all paths ρ = w0, w1, … where w0 = s0: ∃i >= 0: wi ⊧ t

False

Take ρ = (s0 s3)^ω, t never holds in this path (since s0 ∉ π(t) and s3 ∉ π(t)) and so there doesn’t exist an i

* 1. (M1, s0) ⊧ AGEXr

For all paths ρ = w0, w1, … where w0 = s0: ∀i >= 0: wi ⊧ EXr

For all paths ρ = w0, w1, … where w0 = s0: ∀i >= 0: exists a path ρ’ = q0, q1, … where q0 = wi: q1⊧ r

False

Take ρ = s0s1s2^ω, i = 1, wi = s1 and s1⊭ EXr (since the only successor of s1 is s2 and s2 ∉ π(r))

* 1. (M1, s0) ⊧ E((p ∨ r) U (EXr))

There exists a path ρ = w0, w1, … where w0 = s0: ∃j >= 0: wj ⊧ EXr and ∀0 <= k < j: wj ⊧ p ∨ r

True

Take ρ =(s0 s3)^ω, j = 0

RTP: s0 ⊧ EXr

There exists a path ρ’ = q0, q1, … where q0 = s0: q1 ⊧ r

Take ρ’ =(s0 s3)^ω, s3⊧ r (since s3 ∈ π(r))

And for all quantification over k becomes empty and trivially true

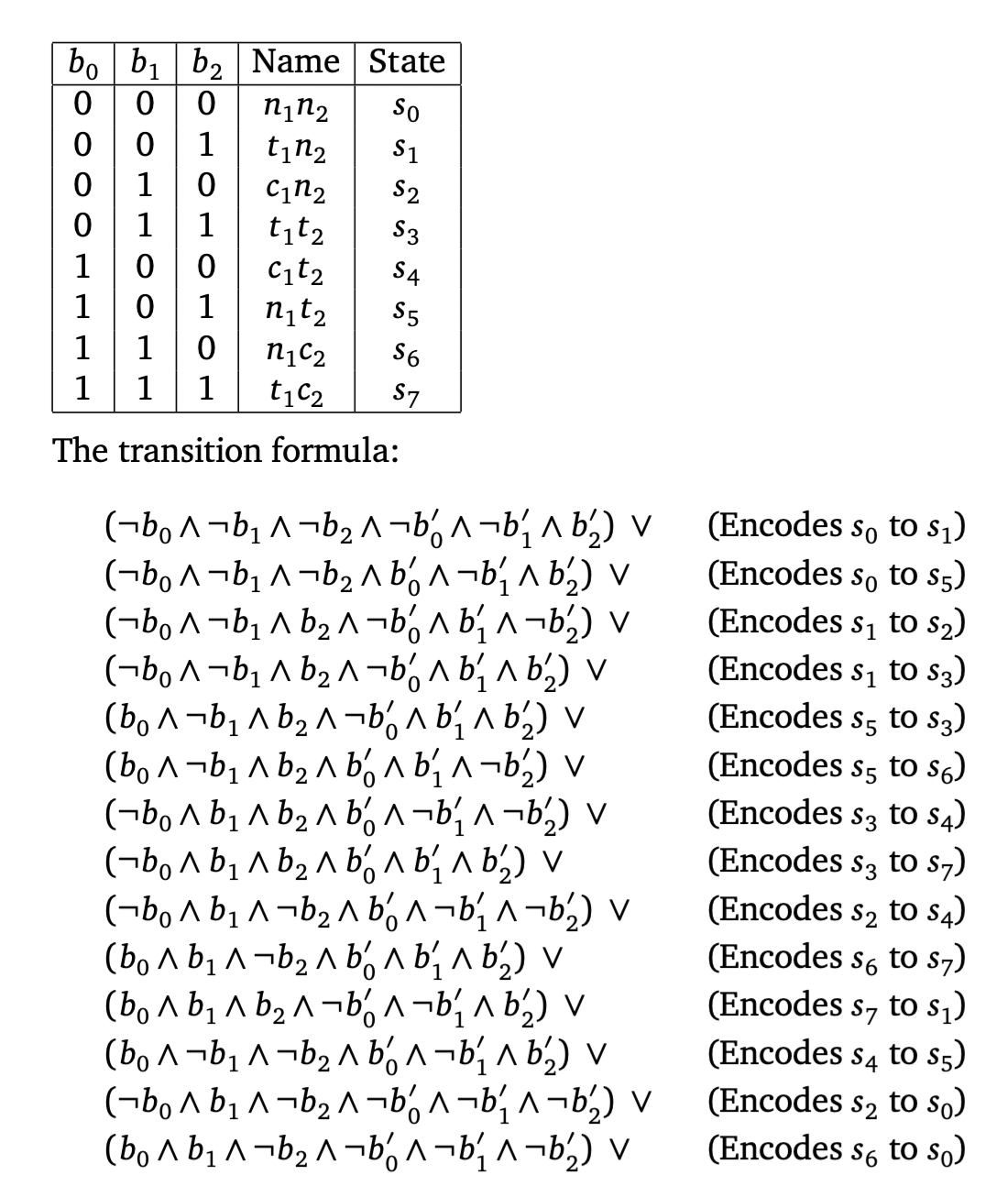
1. A minimal triple of CTL operators is a set of three CTL operators such that every other operator in CTL can be expressed in terms of these three CTL operators. (?)

EFφ ≡ E (true U φ)

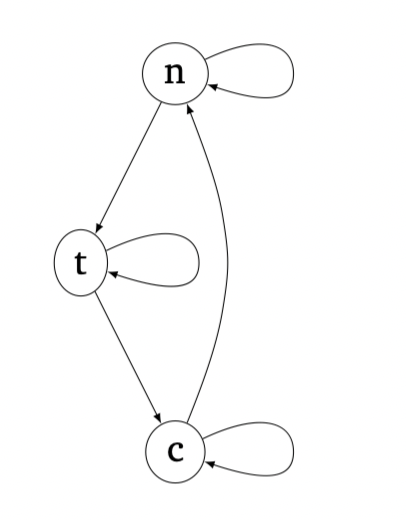
AGφ ≡ ㄱEFㄱφ ≡ ㄱE (true U ㄱφ)

**Question 2**

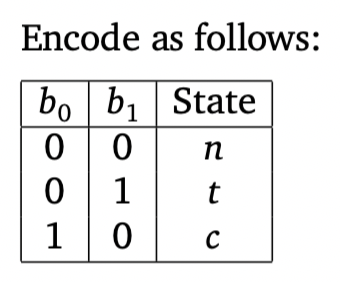
Part A:

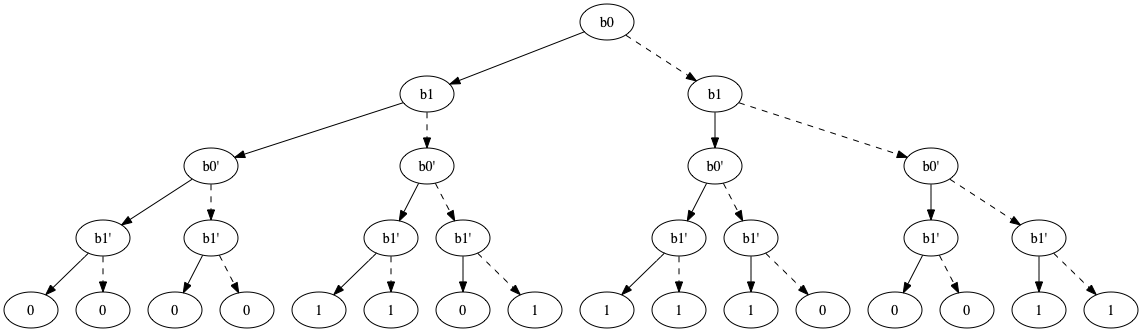
****

Part B:



Part C:

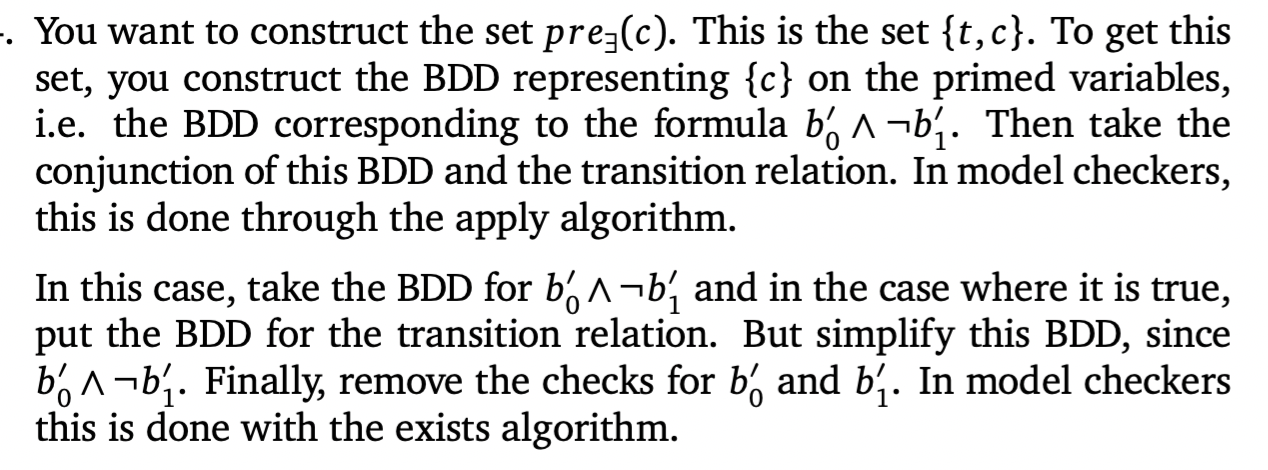


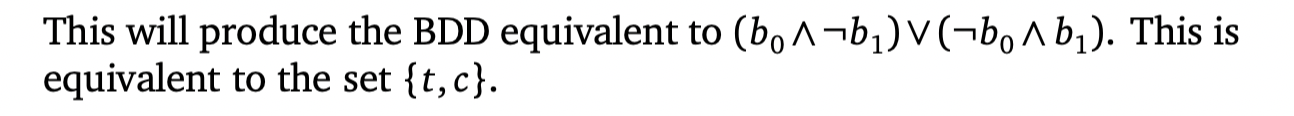


Part D:

I cba to do this.

Part E:



****

**Question 3**