Cheat sheet Specification and verification

Beau De Clercq

lecture 2

Reach(T) = {s$\in S | \exists s_0 \in I \wedge s_0 \rightarrow ... \rightarrow s_n = s\}$ = Post*(I)

program graph: digraph with conditions on the edges, PG=(Locations, Actions, Effect, transition relation, Initial locations, initial

lecture 3

safety (something bad never happens): ¬F(formula), liveness (something good will happen): GF(...), persistence (ensure property holds forever): FG(...), unconditional fairness: for all i $\wedge_i$ GF(...), strong fairness: for all i $\wedge_i$ GF(...) → GF(...), weak fairness: for all i $\wedge_i$ FG(...) → GF(...)

temporal operators: ○, U, G, F

derived operators: Fϕ = TUϕ, Gϕ = ¬F¬ϕ, ϕWψ = (ϕUψ)∨Gϕ, ϕRψ = ¬(¬ϕU¬ψ)

Words(ϕ) = {w=$a_0a_1a_2$... $\in (2^p)^\omega | w \models \phi\}$, $\pi \in$Paths(T): $\pi \neg \models \phi \leftrightarrow \pi \models \neg\phi$, TS T $\neg \models \phi \leftarrow$ T $\models \neg\phi$ (Traces(T) ∈Words(¬ϕ))

lecture 4

NFA NBA concatenation: I = I^1 if $I^1 \cap F^1$ is empty, $I^1 \cup I^2$ otherwise; transition: (q, A) = δ^1(q, A) if q$\in Q^1$ and δ^1(q, A)$\cap F^1$ is empty; δ^1(q, A)$\cup I^2$ if q$\in Q^1$ and δ^1(q, A)$\cap F^1$ is not empty; δ^2(q, A) if q$\in Q^2$

ω-operator for NFA: add transitions from q_{new} to all states directly reachable by q_0; for all states in F, if δ(q, q_f, α) then add δ(q, q_{new}, α); the new set $I' = F' = I$; remove useless states

L(NBA) $\neq \phi$ iff there is an accepting state on a reachable cycle

Accepting run for NBA: sequence of states such that $q_i \in F$ for infinitely many indices i, L(NBA) = {all words for which there is an accepting run}

lecture 5

Closure(ϕ) = set of all sub-formulas and their negation; a set of sub-formulas B ∈ Closure is elementary if B is logically and locally consistent as well as maximal; states for a GNBA for LTL are all elementary sets, if ϕ is in B B is initial, the initial sets are those with an Until or the second part of an Until

Persistence checking: compute reachable SCCs and check if one contains a state satisfying ¬ϕ OR construct T and $A_{\neg\phi}$ in parallel and simultaneously construct the reachable fragment of the product via nested DFS

lecture 6

State formula: $\Phi = T|a|\Phi \wedge \Psi|\neg\Phi|\exists\phi|\forall\phi$; path formula: $\phi = \bigcirc\Phi|\Phi$UΨ

CTL derived operators: $\exists\Diamond\phi = \exists(TU\phi)$, $\forall\Diamond\phi = \forall(TU\phi)$, $\exists\Box\phi = \neg\forall\Diamond\neg\phi$, $\forall\Box\phi = \neg\exists\Diamond\neg\phi$

$Sat_T(\Phi) = s \in S | s \models \Phi$; $s \in S\neg \models \Phi \leftrightarrow s \models \neg\Phi$, TS T $\neg \models \Phi \leftarrow$ T $\models \neg\Phi$

lecture 7

define boolean function with truth table, can only use $\wedge, \vee, \neg$, negating BDD: replace 0 and 1 constant value leaves, from decision tree to BDD: merge/remove useless subtrees, merge equivalent nodes

lecture 8

Symbolic model checking via BDDs: encode states as bitvectors, represent transitions as boolean functions (==switching functions) = characteristic function $\chi_R(s) = 1$ if $s \in R$ and 0 otherwise, represent labeling via $\chi_{Sat(s)}$; at the end of the model checking process check that $\neg\chi_I \vee f_{Sat(\Phi)} = 1$

lecture 10

Given a circuit with automaton A, check if there exists a function μ such that the language of Axμ is empty

AIGER: aag M I L O A; every variable and AND-gate is represented by an even index i ≥2, negations by i+1; i ≤2M+1 for every index i

Games played on automata: states Q just keep track of latch values, Σ corresponds to valuations of the inputs, δ respects the latch next-step functions

LTL to games: construct NBA for ϕ, determinize it to a DPA and attempt to synthesize a strategy for player in the parity game