Problem Sheet 8: Model Checking
Sample Solutions

Chris Poskitt and Carlo A. Furia
ETH Zürich

1 Evaluating LTL Formulae on Automata

i. Yes: whenever \texttt{start} occurs, \texttt{stop} must occur eventually since it is the only means of getting to the accepting state.

ii. No: a counterexample is \texttt{pull push}.

iii. Yes: the formula asserts that from every position in a word (if there are any), eventually either \texttt{turn off} or \texttt{push} will occur. One of these events must occur to return to the accepting state.

iv. No: the empty word is a counterexample (\texttt{\textcircled{d} p} demands the existence of a future position in the word for which \texttt{p} holds — the empty word cannot possibly satisfy it as it has no positions).

v. Yes: if the word is empty, then it will satisfy the first disjunct (“always false” holds simply because there are no positions in the empty word to check against): if the word is non-empty, the final position in the word must be \texttt{turn off} or \texttt{push}, and hence the second disjunct will be satisfied.

vi. No: a counterexample is the empty word; or \texttt{turn on turn off}.

2 Equivalence of LTL Formulae

i. \[ w, i \models \texttt{true} \cup F \]
   iff for some \[ i \leq j \leq n \] we have \[ w, j \models F \]
   and for all \[ i \leq k < j \] we have \[ w, k \models \texttt{true} \] \[ \text{[definition of until]} \]
   iff for some \[ i \leq j \leq n \] we have \[ w, j \models F \] \[ \text{[semantics of true]} \]

ii. \[ w, i \models \neg \texttt{\textcircled{d} \neg F} \]
   iff \[ w, i \not\models \texttt{\textcircled{d} \neg F} \] \[ \text{[definition of not]} \]
   iff it is \textit{not} the case that for some \[ i \leq j \leq n \] we have \[ w, j \models \neg F \] \[ \text{[semantics of eventually]} \]
   iff for all \[ i \leq j \leq n \] it is not the case that \[ w, j \models \neg F \] \[ \text{[semantics of quantifiers]} \]
   iff for all \[ i \leq j \leq n \] it is not the case that \[ w, j \not\models F \] \[ \text{[semantics of negation]} \]
   iff for all \[ i \leq j \leq n, w, j \models F \] \[ \text{[simplify double negation]} \]
iii.

$$w, i \models \Diamond \Diamond p$$

iff for some $$i \leq j \leq n$$ we have $$w, j \models \Diamond p$$  \[\text{semantics of eventually}\]

iff for some $$i \leq j \leq h \leq n$$ we have $$w, h \models p$$  \[\text{sem. eventually; merging intervals}\]

iff for some $$i \leq h \leq n$$ we have $$w, h \models p$$  \[\text{a fortiori}\]

iff $$w, i \models \Diamond p$$  \[\text{semantics of eventually}\]
3 Automata-Based Model Checking

i. The automaton we build from the temporal formula is the following.

![Automaton Diagram]

ii. The intersection automaton is the following:

![Intersection Automaton Diagram]

iii. Any accepting run is a counterexample to the LTL formula being a property of the microwave oven automaton. There are several, for example: pull push, pull push pull push, ...