Problem Sheet 8: Model Checking
Sample Solutions

Chris Poskitt and Carlo A. Furia
ETH Zürich

1 Evaluating LTL Formulae on Automata

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

ii. No: a counterexample is pull push.

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

iv. No: the empty word is a counterexample (∗ p demands the existence of a future position in the word for which 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 turn off or push, and hence the second disjunct will be satisfied.

vi. No: a counterexample is the empty word; or turn on turn off.
2 Equivalence of LTL Formulae

i.

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

ii.

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

iii.

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