1 Recap

1. The Manna & Pnueli Hierarchy
2. Characterization of Languages accepted by DBW/NBW
3. New acceptance criteria: Generalized Büchi and coBüchi.

2 Continuing with Other Acceptance Conditions

Recall the definition of coBüchi acceptance.

Definition 2.1. A coBüchi acceptance condition is a subset \( F \subseteq Q \). A run \( r \) is accepting according to the coBüchi condition iff

\[
\inf(r) \cap F = \emptyset.
\]

Example 2.2. The following coBüchi automaton accepts the language \( \infty a \)'s.

![coBüchi automaton diagram]

Theorem 2.3. \( L \in DBW \iff L \in DCW \).

Proof. Proof sketch Just consider the same structure as DCW.

Earlier we showed that there is no DBW accepting \( L_{\infty a} \).

Corollary 2.4. There is no DCW for the language \( \infty a \)'s

Question Can this language be accepted by an NCW?

Answer No.

Theorem 2.5. \( NCW = DCW \).

Proof later....

Question Can we translate an NCW to an NBW?

Theorem 2.6. An NCW with \( n \) states can be translated to an NBW with at most \( 2n \) states

Proof Sketch. }
Question In those cases when an NBW has an equivalent NCW, can one be defined on the same structure?

Example 2.7. Example where this holds: Consider the NBW $B_1$ defined as follows:

Then $|B_1| = \{ w \in \{a, b\}^\omega : |w|_a \geq 1 \text{ and } |w|_b \geq 1 \}$ and we can define a DCW $C_1$ on the same structure by setting $F = \{1, 2, 3\}$. It holds that $|C_1| = |B_1|$.

Example 2.8. Example where this does not hold. Let $L_{gb \geq 1} = a^* b(a + b)^\omega$. Then $L_{gb \geq 1}$ is in NCW since $|C_2| = L_{gb \geq 1}$ as well as in NBW since $|B_2| = L_{gb \geq 1}$.

However, no equivalent NCW can be defined on the structure of $B_2$.

3 Back to Characterizations

Theorem 3.1. $L \in \text{DCW} \iff \text{there exists a regular language } R \text{ such that } L = \mathcal{P}_{\text{pref}}(R)$.

Proof Sketch. Let $L \in \text{DCW}$. Then there exists a DCW $D = (\Sigma, Q, Q_0, \delta, F)$ such that $L = |D|$. Define $DFW D' = (\Sigma, Q, Q_0, \delta, \emptyset \setminus F)$. We claim that $L = \mathcal{P}_{\text{pref}}(|D'|)$.

4 Streett and Rabin automata

Definition 4.1. A Rabin acceptance condition is a set of pairs of sets of states $\{ \langle G_1, B_1 \rangle, \ldots, \langle G_k, B_k \rangle \}$. A run $r$ is accepting according to the Rabin condition iff

Claim 4.2. Obviously $\text{NRW} \supseteq \text{NBW}$.

Proof Sketch. $B = (\Sigma, Q, Q_0, \delta, F) \rightsquigarrow R = (\Sigma, Q, Q_0, \delta, \{ \{F, \emptyset\} \})$
Claim 4.3. Obviously \( \text{NRW} \supseteq \text{NCW} \).

Proof Sketch. Let \( \mathcal{C} = (\Sigma, Q, Q_0, \delta, F) \leadsto \mathcal{R} = (\Sigma, Q, Q_0, \delta, \{(Q, F')\}) \)

Claim 4.4. If \( q \in G_i \cap B_i \) we can replace this pair with \( (G_i \setminus \{q\}, B_i) \) without changing the language.

Definition 4.5. A Streett acceptance condition is a set of pairs of sets of states \( \{(G_1, B_1), \ldots, (G_k, B_k)\} \).

A run \( r \) is accepting according to the Streett condition iff

\[ \forall i \text{ if } \inf(r) \cap G_i \neq \emptyset \text{ then } \inf(r) \cap B_i \neq \emptyset \]

4.1 From Rabin and Streett to Büchi

Example 4.6. Two examples on the same structure:

\[ R_1 : \alpha = \{(2), 0\} \quad L_1 = \infty b \]

\[ R_2 : \alpha = \{(1, 2)\} \quad L_2 = \sim \infty b \]

Claim 4.7. \( \text{NSW} \supseteq \text{NGBW} \).

Proof Sketch. \( \mathcal{G} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\}) \leadsto \mathcal{S} = (\Sigma, Q, Q_0, \delta, \{(F_1, 0), \ldots, (F_k, 0)\}) \)

Claim 4.8. Let \( \mathcal{R}_i = (\Sigma, Q, Q_0, \delta, \{(G_i, B_i)\}) \).

Then \( \mathcal{S} = \cup_{1 \leq i \leq k} [\mathcal{R}_i] \).

Proof Sketch. \( \mathcal{G} = (\Sigma, Q, Q_0, \delta, \{F_1, \ldots, F_k\}) \leadsto \mathcal{R} = (\Sigma, Q, Q_0, \delta, \{(F_1, 0), \ldots, (F_k, 0)\}) \)

Claim 4.9. \( \text{NRW}[1] \subseteq \text{NBW} \).

Proof Sketch.

\[ \begin{array}{c}
Q \\
\cup \\
Q \setminus B \\
\cup \\
Q \\
\end{array} \]

Corollary 4.10. we can translate \( \text{NRW}[k] \) to \( \text{NBW} \) using \( 2nk \) states.

Claim 4.11. \( \text{NSW} \subseteq \text{NBW} \).

Proof Sketch. For all \( i \) we need to visit \( G_i \) finitely often or \( B_i \) infinitely often. Thus \( \exists I \subseteq [1..k] \) such that for all \( j \in I \), \( G_i \) is visited only finitely often, and for all \( j \notin I \), \( B_j \) is visited infinitely often. So \( \mathcal{S} = \cup_{j \in I} [\mathcal{G}_j] \) where \( G_j \) is an NGBW recognizing the words that do not visit \( \cup_{j \in I} G_j \) from some point onwards, and visits each of the sets \( \{B_{j_1}, B_{j_2}, \ldots, B_{j_\ell}\} \) for \( j_i \notin I \) infinitely often.
5 Fundamental questions

Question When is an automaton un-interesting?

Definition 5.1. Given an automaton \( A \)

- The \textit{emptiness} question is to decide whether \( |A| = \emptyset \).
- The \textit{universality} question is to decide whether \( |A| = \Sigma^* \) or \( |A| = \Sigma^\omega \) depending on the context.

Theorem 5.2. The non-emptiness problem for NFW is decidable in linear time and is NLOGSPACE-complete.

Proof Sketch. The language of an NFW \( A \) is non-empty if there exists a state \( q \in F \) such that \( q \) is reachable from an initial state.
Thus, we can compute the set of states \( R \) reachable from an initial states of \( A \) using BFS (breadth-first search), and conclude that \( A \) is non-empty iff \( R \cap F \neq \emptyset \).
BFS constructs \( R \) in linear time.

For space complexity, we note that graph reachability can be tested in non-deterministic log-space. The TM tries to construct a path of reachable states ending in a state in \( F \). At each state the algorithm needs to remember the current state and the next state, for which \( 2(\log n) \) bits suffice.
The completeness part of the theorem follows since graph reachability is also NLOGSPACE hard.

Theorem 5.3. The non-emptiness problem for NBW is decidable in linear time and is NLOGSPACE-complete.

Proof Sketch. The language of an NBW \( A \) is non-empty if there exists a state \( q \in F \) such that \( q \) is reachable from an initial state, and \( q \) is reachable from itself. We say that a strongly connected component (SCC) is \textit{non-trivial} if it contains at least one edge. Since it is strongly connected this implies that it also contains a cycle. Thus \( A \) is non-empty iff there exists a non-trivial SCC that intersects \( S_0 \), from which a non-trivial SCC that intersects \( F \) is reachable. Therefore, non-emptiness of NBWs also reduces to reachability.

It is possible to construct the decomposition of a graph into SCCs using depth-first-search (DFS) in linear time.

For the space result, we again use a non-deterministic TM that guesses first a path from an initial state \( q_0 \) to an accepting state \( q \) vertex by vertex, and then guesses a path from \( q \) to itself vertex by vertex. At any moment in time it needs to remember at most 3 vertices, the current one, the guessed one and \( q \). It thus suffices with logarithmic space. Hardness again follows from hardness of the reachability problem.