Recall: Decidability of Satisfiability/Realisability from 0

Theorem 3.6. The satisfiability problem for RDC with discrete time is decidable.

Theorem 3.9. The realisability problem for RDC with discrete time is decidable.

Sketch: Proof of Theorem 3.6

• Give a procedure to construct, given a formula $F$, a regular language $L(F)$ such that $I, [0,n] \models F$ if and only if $w \in L(F)$ where $w$ describes $I$ on $[0,n]$ (suitability of the procedure: Lemma 3.4)

• Then $F$ is satisfiable in discrete time if and only if $L(F)$ is not empty (Lemma 3.5)

• Theorem 3.6 follows because
  • $L(F)$ can effectively be constructed,
  • the emptiness problem is decidable for regular languages.
Lemma 3.8. We have $F(k, r)$. If $F$ is regular, then $L$ is realisable from 0 in $F$. 

Definition 3.2. For $x, y, z$ basic state variables, $F$ contains exactly state variables.

Lemma 3.4. Construction of RDC in Continuous Time.

(1) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(2) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(3) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(4) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(5) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(6) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(7) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(8) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(9) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(10) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(11) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.

(12) $F$ contains exactly state variables. If $F$ contains exactly state variables, then $F$ can be transformed into an equivalent DNF.
Theorem 3.11. The realizability from 0 problem for DC with final state is undecidable.

Sketch: Proof of Theorem 3.10

Recall: Two-counter machines

- The emptiness problem for DC with final state is decidable.
- The satisfiability problem for DC is decidable.
- The emptiness problem for 2C is undecidable.
- The satisfiability problem for 2C is undecidable.
- The realizability from 0 problem for 2C with final state is undecidable.

Proof: We reduce from the emptiness problem for 2C to the realizability problem for DC with final state.

1. Construct a DC formula $\phi$ such that if $M$ is a 2C with final state, then $\phi$ is a DC formula.
2. If $M$ is empty, then $\phi$ is satisfiable.
3. If $M$ is not empty, then $\phi$ is not satisfiable.
4. Therefore, the realizability problem for DC with final state is undecidable.

Theorem 3.10. The emptiness problem for DC with final state is decidable.

Proof: We construct a DC formula $\psi$ such that if $M$ is a DC with final state, then $\psi$ is a DC formula.

1. If $M$ is empty, then $\psi$ is a DC formula.
2. If $M$ is not empty, then $\psi$ is not a DC formula.
3. Therefore, the emptiness problem for DC with final state is decidable.

# Two-Counter Machines and Configurations

- The emptiness problem for 2C is undecidable.
- The satisfiability problem for 2C is undecidable.
- The realizability problem for 2C with final state is undecidable.

Proof: We reduce from the emptiness problem for 2C to the realizability problem for 2C with final state.

1. Construct a 2C formula $\phi$ such that if $M$ is a 2C, then $\phi$ is a 2C formula.
2. If $M$ is empty, then $\phi$ is a 2C formula.
3. If $M$ is not empty, then $\phi$ is not a 2C formula.
4. Therefore, the realizability problem for 2C with final state is undecidable.
∀x, RDC
Contradiction. is valid.

Note: the DC fragment defined by the following grammar is not even semidecidable.

Formulae used in the reduction are abbreviations:

\[ F \land \neg \Box M (F) \iff \neg F \lor \neg \neg F \leq \neg F \land \neg \neg F \]
\[ F \lor \neg \Box M (F) \iff \neg F \land \neg \neg F \leq \neg F \lor \neg \neg F \]
\[ F \land \neg \Box M (F) \iff \neg F \lor \neg \neg F \leq \neg F \land \neg \neg F \]
\[ F \lor \neg \Box M (F) \iff \neg F \land \neg \neg F \leq \neg F \lor \neg \neg F \]

(i) If zero
\[ \neg F \land \neg \neg F \leq \neg F \land \neg \neg F \]
\[ \neg F \lor \neg \neg F \leq \neg F \lor \neg \neg F \]

(ii) Decrement counter
\[ \neg F \land \neg \neg F \leq \neg F \land \neg \neg F \]
\[ \neg F \lor \neg \neg F \leq \neg F \lor \neg \neg F \]

(iii) Increment counter
\[ \neg F \land \neg \neg F \leq \neg F \land \neg \neg F \]
\[ \neg F \lor \neg \neg F \leq \neg F \lor \neg \neg F \]

Theorem 3.11. This yields fin in M.

Following [Chaochen and Hansen, 2004] the DC formula