Infinite Automata 2025/26

Lecture Notes 3

Henry Sinclair-Banks

The *encoding* of a CM, VAS, or VASS is important. There are two common encoding types: *unary encoding* and *binary encoding*. If we say that a VASS is encoded in, we mean that the integer vectors on the transitions are encoded in unary. Let (Q, T) be a VASS. The size of a transition $(p, \mathbf{x}, q) \in T$ is $\log(|Q|) + \|\mathbf{x}\|_1 + \log(|Q|)$ (the $\log(|Q|)$ terms are for writing down the states p and q, and $\|\mathbf{x}\|_1$ is for writing down each of the components of the vector. Note that the components of the vector are *not* written in binary. So, for example, the size of the vector $(7, -3, -11)$ is $\|(7, -3, -11)\|_1 = 21$. On the other hand, if we say that a VASS is encoded in binary, we mean that the integer vectors on the transitions are encoded in binary. Thus, when binary encoding is used, the size of a transition $(p, \mathbf{x}, q)$ is bounded above by $\log(|Q|) + d\log(\|\mathbf{x}\|_\infty) + \log(|Q|)$.

It is always true that the computational complexity of a problem may increase when binary encoding is used to specify the input to the problem. It could be the case that the complexity does not increase; but for VAS(S) of low dimension, the encoding can make big difference. The following theorem is Exercise 2.1.

Theorem 3.1. Reachability in binary-encoded 1-VASS is NP-hard.

Proof hint. Reduce from the subset sum problem. □

To give another perspective on binary-encoding versus unary-encoding, consider the subset sum problem. Subset sum is a classic NP-complete problem. This is true when the input to the subset sum problem is encoded in binary. It is also true that there is a straightforward $\mathcal{O}(nt)$-time dynamic programming algorithm that can decide subset sum. Here t is the target value of subset sum. One just maintains a two-dimensional dynamic programming table with the current index of the item being considered and the every possible subset sum value that is between 0 and t. This algorithm does not place the binary-encoded variant subset sum problem in P because the t is exponential compared to the number of bits required to write down t. However, if the input to the subset sum problem is encoded in unary, then really the size of t is just t and so this algorithm is a polynomial time algorithm. It is therefore true that unary-encoded subset sum is in P (in fact it is in NL).

We will now prove a tight upper bound of the complexity of reachability in binary-encoded 1-VASS.

Theorem 3.2. Reachability in binary-encoded 1-VASS is in NP.

Before proving Theorem 3.2, recall Exercise 3.3.

Definition 3.3. Let $X \subseteq \mathbb{Z}$ be a finite set, then

$$\text{cone}_{\mathbb{Z}}(X) \coloneqq \{n_1x_1 + \ldots + n_kx_k : n_1, \ldots, n_k \in \mathbb{N} \text{ and } x_1, \ldots, x_k \in X\}.$$

Lemma 3.4. Let $X \subseteq \mathbb{Z}$ be a finite set and let $b \in \text{cone}_{\mathbb{Z}}(X)$. Let $M \coloneqq \max\{|x| : x \in X\}$. If $|X| > \log(2M|X| + 1)$, then there exists a proper subset $X \subset Y$ such $b \in \text{cone}_{\mathbb{Z}}(X')$.

We will also use the fact that if there exists a run from a given starting configuration $(p, 0)$ to a target configuration $(q, 0)$, counter value need not exceed an exponential value.

Lemma 3.5. Let V be a binary-encoded 1-VASS and let $(p, 0)$, $(q, 0)$ be two configurations. Let n be the number of states in V and let M be the greatest absolute value of any transition update. There exist a polynomial f such that if $(p, 0) \xrightarrow{*}_V (q, 0)$, then there exists a run from $(p, 0)$ to $(q, 0)$ such that all configurations in the run have counter values at most $f(n) \cdot M$.

Proof. Follows from Lemma 1.10 after directly converting the given binary 1-VASS into a unary-encoded 1-VASS (or 1-CM) by replacing a transition (p, x, q) with x many incremental (or decremental) transitions. □

Definition 3.6. The *nadir* of a cycle is the state in which the cycle attains its lowest value. For example, suppose there is a cycle with overall effect $+8$ that takes the transitions $(p, 3, q)$, $(q, -5, r)$, and $(r, 10, p)$, then the lowest value will be attained at state r which makes r the nadir of this cycle. Note that if a cycle is positive and has nadir r, then the cycle can be taken from $(r, 0)$. If a cycle is negative and has nadir r, then the cycle can be taken from (r, x) where $-x$ is the effect of the cycle.

Proof of Theorem 3.2. Let $(V, (p, 0), (q, 0))$ be an arbitrary instance of reachability in binary-encoded 1-VASS. Let n be the number of states in V and let M be the greatest absolute value of any transition update. Suppose $(p, 0) \xrightarrow{*}_V (q, 0)$ and let $(p, 0) \xrightarrow{\pi}_V (q, 0)$ be a run in which the counter does not exceed $f(n) \cdot M$ (Lemma 3.5).

First, for each control state q, we shall "mark" the (up to) two configurations in which q is first and last visited in the run $(p, 0) \xrightarrow{\pi}_V (q, 0)$. For convenience, we'll call c_q^+ the marked configuration for the first occurrence of q and c_q^- the marked configuration for the last occurrence of q. For every state r, we shall define multisets S_r^+ and S_r^- that will contain positive and negative cycles, respectively. The multiset S_r^+ is associated with the marked configuration c_r^+ and S_r^- is associated with c_r^-. Now, we will repeatedly apply the following procedure from left to right. Search for a simple cycle in which there does not exist a marked configuration. Recall Exercise 2.5; it is true that if the length of the run is greater than $2n^2 + n$, then there must exist a simple cycle that does not contain a marked configuration. Suppose such a cycle γ has been obtained by the procedure and suppose that r is the nadir of γ. We shall remove γ from $(p, 0) \xrightarrow{\pi}_V (q, 0)$ and place it in S_r^+ if the effect of is positive, otherwise we will place it in S_r^-. Intuitively speaking, S_r^+ contains all of the positive simple cycles that can be taken from state r as early as possible in the run and S_r^- contains all of the negative simple cycles that can be taken from state r as late as possible in the run.

Once the above procedure has been completed, we are left with at most n multisets of positive effect cycles, at most n multisets of negative effect cycles, and a "skeleton" of the run that has length at most $2n^2 + n$ that contains the marked configurations. Here, we are using "skeleton" to refer to the underlying transitions taken as the result of the procedure will remove sections of the run which makes some configuration sequences invalid. Importantly, the marks on the configurations are still preserved in the skeleton. First, observe that the following run can be executed: follow the skeleton, and whenever a marked configuration c_r^+ is reached, immediately take all cycles from its associated multiset S_r^+ (in any order). Similarly, whenever a marked configuration c_r^- is reached, immediately take all cycles from S_r^- (in any order). The reason that this is a valid run is because: cycle are being taken from their nadirs, positive cycles are moved earlier in the run, negative cycles are moved to later in the run (which together helps to increase the counter in the middle of the run), and the fact that marked configurations are never moved means that the run is always connected in the underlying directed graph.

Now we will apply Lemma 3.4 to replace every S_r^+ and every S_r^- with an alternative multiset of cycles. Consider any set S_r^+. Let b be the sum of all effects of all cycles in S_r^+, $b \coloneqq \sum_{\gamma \in S_r^+} \text{effect}(\gamma)$. First, since the length of $(p, 0) \xrightarrow{\pi}_V (q, 0)$ is at most $n(f(n) \cdot M + 1)$, we know that $b \leq n(f(n) \cdot M + 1)$ as there cannot be more than $n(f(n) \cdot M + 1)$ many simple cycles in $(p, 0) \xrightarrow{\pi}_V (q, 0)$. Let E be the *set* of all distinct cycle effects of S_q^+, $E \coloneqq \{\text{effect}(\gamma) : \gamma \in S_r^+\}$. Clearly, $b \in \text{cone}_{\mathbb{Z}}(E)$. Note also that $E \subseteq [1, nM]$ because the greatest possible effect of a positive simple cycle is bounded by the greatest positive transition effect M times the number of states n; hence $|E| \leq nM$. Now, we can repeatedly use Lemma 3.4 to obtain a subset $E' \subseteq E$ such that $b \in \text{cone}_{\mathbb{Z}}(E')$ and $|E'| \leq \log(2n^2M^2 + 1)$; here $|E| \leq nM$ and $\max\{e \in E\} \leq nM$. Suppose $E' = \{e_1, \ldots, e_k\}$; then we know that there exists

$n_1, \ldots, n_k \in \mathbb{N}$ such that $b = \sum_{i=1}^{k} n_i e_i$. Moreover, since $b \leq n(f(n) \cdot M + 1)$, we know that all $n_i \leq n(f(n) \cdot M + 1)$ as well.

We can accordingly define a new multiset of cycles T_r^+ such that $\sum_{\gamma \in S_r^+} \text{effect}(\gamma) = \sum_{\gamma \in T_r^+} \text{effect}(\gamma)$. To do this, we have to look back at the definition of E. First, observe that since E was the *set* of effects of cycles in S_r^+, then we can, for every $e \in E$, record a cycle γ_e such that γ_e has nadir r and effect e. It may well be the case that there are multiple different cycles in S_r^+ that have nadir r and effect e, but it suffices to record just one. Then, notice that since $E' \subseteq E$, we can obtain a record of cycles that have effects in E' (we just keep the same γ_e for every $e \in E'$. Now, give than $b = \sum_{i=1}^{k} n_i e_i$, we can define the multiset T_r^+ by adding n_i copies of γ_{e_i} for each i. It is therefore true that $\sum_{\gamma \in S_r^+} \text{effect}(\gamma) = b = \sum_{\gamma \in T_r^+} \text{effect}(\gamma)$.

Note that the previous arguments can be symmetrically applied to the multisets S_r^- as well to obtain new multisets T_r^-.

Overall we have proved that if reachability holds in a binary-encoded 1-VASS, then there must be a run with the following structure: there is a short path to a small collection of short cycles, each of which is taken up to an exponential number of times, then there is another short path to another small collection of short cycles, each of which is taken up to an exponential number of times, then etc. That is because the multiset of cycles S_r^+ can be replaced with the equivalent multiset of cycles T_r^+ To be precise, when we say "small" collection of cycles, we have proved that in each collection there need not be more than $\log(2nM^2)$ many cycles. Clearly $\log(2nM^2) = \log(2n) + \log(M^2) \leq 2n + 2\log(M)$ which is linear in the size of the binary-encoded 1-VASS V. It remains to argue that reachability is in fact in NP.

We shall use a very similar NP polynomial sized certificate and polynomial time checking algorithms as was used in Exercise 3.1. (Reachability in binary-encoded 1-dimensional linear path schemes is in NP.) We just guess a run with the aforementioned structure. Such a guess is polynomially sized because (i) the skeleton has polynomial length, (ii) the number of simple cycles take in total is polynomial, and (iii) the number of times each cycle is taken is at most $n(f(n) \cdot M + 1)$ which, when written in binary, only required polynomially many bits.

We first check underlying (skeleton) path is connected; we can do this by one-by-one checking that transitions connect. Then we check that the run specified by the cycles and iteration counts is in fact a valid run (i.e. that the counter remains nonnegative throughout). For this, when following the underlying (skeleton) path, we calculate the current configuration step-by-step, then when we get to a collection of cycles, we do the following check. If the cycle is positive, we first attempt to take *the first* iteration of the cycle (step-by-step) to make sure that the counter does not drop below zero. If the first iteration is successful, we know by monotonicity that the cycle can be iterated as many times as specified without concern for dropping below zero. So we just calculate the configuration reached after this block of cycle iterations by multiplying its effect by the remaining iteration count and adding that to the current counter value. However, if the cycle is negative, we do the same as previous except that we take *the last* iteration of the cycle step-by-step to make sure that the counter does not drop below zero. The total number of checks is equal to: the length of the underlying (skeleton) path plus the length of every cycle iterated. This means that the checking algorithm indeed runs in polynomial time. □

Corollary 3.7. Reachability in binary-encoded 1-CMs is in NP.

Proof sketch. Let $(M, (s, 0), (t, 0))$ be an arbitrary instance of reachability in binary-encoded 1-CMs; M is the 1-CM, $(s, 0)$ is the starting configuration, and $(t, 0)$ is the target configuration. Notice that in a minimal length run from $(s, 0)$ to $(t, 0)$, a zero-testing transition need not be taken more than once. That is because if it ever is taken more than once, then there must be a zero-effect cycle inside the run which can be removed to shorten the run and this would contradict the minimality of the length of the run. This means that apriori, the order in which a run takes the zero-testing transitions can be guessed and then between zero-testing transitions, one just needs to check that reachability holds (without using zero-testing transitions). These reachability checks can, by Theorem 3.2, be done in NP. □