1: Построить явную m-сводимость языка $L_1 = 0^n 1^n$ к языку $L_2 = 0^n$.

При решении такой (и подобных задач) следует сразу понять вот что: $L_1 \in \mathcal{R}$. Вспомним, что сводимость по определению означает вычислимую функцию f, отображающую слова между языками – ключевое слово здесь “вычислимую”.

У нас есть одновременно два факта:

– язык разрешимый, то есть его характеристическая функция вычислима,

– сводимость допускает любые вычислимые функции.

В этом случае требуемая сводимость очень простая: для слова $x \in \Sigma^*$ значение $f(x) = y$, где y определяется по следующему правилу:

– если x имеет вид $0^n 1^n$, то $y = 0$,

– иначе $y = 1$.

Такая сводимость может показаться тривиальной (она и есть тривиальна), но ключевым моментом здесь является разрешимость L_1 – если бы L_1 не был бы разрешим, кусок определения f “если x имеет вид” не мог бы быть вычислен, потому что было бы невозможно проверить, имеет x подходящий вид или нет – сравните с проблемой останова.

В качестве дополнительного упражнения выясните, для каких языков $L \subset \Sigma^*$ в принципе существует сводимость $L_1 \leqslant^m L$.

2: Исследовать язык $L = \{\langle M\rangle \mid \forall x \in \Sigma^* \, (|x| \text{ – простое число} \to x \notin \mathsf{Accept}(M))\}$ на принадлежность классам $\mathcal{R}$, $\mathcal{RE}$, co$\mathcal{RE}$.

Подобные задачи требуют для решения некоторого опыта и интуиции, но в целом схема одна и та же. Вам нужно определить (неформально) две вещи:

– Есть ли такое **проверяемое** условие что, если оно выполнено, то дальше "испортить его выполненность" нельзя. К примеру "машина принимает хотя бы три слова", "машина принимает 15 слов попарно различных длин", "две машины принимают кодировку друг друга". Проверяемость важна – положительный ответ на вопрос должно быть возможно алгоритмически проверить – к примеру "машина не принимает хотя бы три слова" проверить нельзя, а вот "машина отвергает хотя бы три слова" можно. Под "испортить" здесь я понимаю вот что: если некоторая машина уже приняла три каких-то слова, то что бы она ни делала на всех остальных словах, она не перестанет удовлетворять условию "машина принимает хотя бы три слова".

– Есть ли такое **проверяемое** условие что, если оно нарушено, то дальше "испортить его нарушенность" нельзя. К примеру "машина принимает не более трёх слов", "машина не принимает ни одного слова вида 0^n", "машина не принимает ни одного слова".

Выполнение первого из условий должно натолкнуть вас на мысль о принадлежности $\mathcal{RE}$, второго – co$\mathcal{RE}$.

Вернёмся к задаче. Обратите внимание на $\notin$ – здесь есть простое условие нарушения – если данная вам машина примет хотя бы одно слово, длина которого простое число, эта машина не подходит, это никак не исправить и машину можно отвергнуть. Это прямое доказательство принадлежности co$\mathcal{RE}$. Более формально: следующая машина N будет ко-перечислять язык L.

На входе x машина N

1) определяет машину M такую, что $x = \langle M\rangle$,

2) запускает M параллельно на всех входах простой длины,

3) если M приняла какое-то слово простой длины, машина N отвергает вход.

Иначе то же самое можно переписать в Π_1 нотации, наш язык $L = \{x \mid \forall y P(x,y)\}$, где P вычислимый бинарный предикат – P интерпретирует x как кодировку машины M, y – как пару: слово w и число t. Если w имеет не простую длину, он принимает (x,y). Иначе он моделирует работу M на слове w на t шагов. Принимает, если M не приняла w за t шагов. В этом случае для слова x из языка любой y подойдёт. Для слова не из языка не подойдет такая пара (w,t), что соответствующая машина принимает слово w за t шагов.

Ещё одно возможное доказательство принадлежности co$\mathcal{RE}$ – сводимость языка L к $\mathsf{NotHalt}$. Поскольку $\mathsf{NotHalt}$ является co$\mathcal{RE}$-полным языком, сводимость существует. Типично, она выглядит так: функция сводимости $f(\langle M\rangle) = (\langle N\rangle, \langle M\rangle)$. Машина N та же, что и раньше. Очевидно, f вычислима.

Докажем корректность сводимости:

– если $\langle M\rangle \in L$, то M не принимает ни одного слова простой длины, шаг 2 машины N будет работать бесконечно, так как слов простой длины бесконечное количество, шаг 3 машины N никогда не наступит, тогда машина N не остановится на входе $\langle M\rangle$, тогда пара $(\langle N\rangle, \langle M\rangle) \in \mathsf{NotHalt}$;

– если $\langle M\rangle \notin L$, то M принимает какое-то слово w простой длины за какое-то конечное число шагов t, шаг 2 машины N будет работать до тех пор, пока параллельный запуск не дойдёт до нужного слова и нужного числа шагов, машина N перейдёт к шагу 3 и остановится на входе $\langle M\rangle$, тогда пара $(\langle N\rangle, \langle M\rangle) \notin \mathsf{NotHalt}$.

Для доказательства непринадлежности $\mathcal{RE}$ докажем сводимость в обратную сторону: $\mathsf{NotHalt} \leqslant^m$

L. Определим $f((\langle M\rangle,\omega)) = \langle M_\omega\rangle$, где машина M_ω на входе x

1) моделирует работу M на слове ω,

2) если M остановилась, принимает x.

Функция сводимости очевидно вычислима (это просто маленький кусок универсальной машины Тьюринга), докажем корректность сводимости.

– если $(\langle M\rangle,\omega) \in \mathsf{NotHalt}$, то M не останавливается на ω, тогда для любого входа x машина M_ω никогда не перейдёт ко второму шагу, а значит никогда не остановится, тогда машина M_ω не принимает никакой вход x, в том числе она не принимает никакой x простой длины, следовательно $\langle M_\omega\rangle \in L$;

– если $(\langle M\rangle,\omega) \notin \mathsf{NotHalt}$, то M останавливается на ω, тогда для любого входа x машина M_ω перейдёт ко второму шагу и примет вход x, в том числе машина M_ω примет любой x простой длины, следовательно $\langle M_\omega\rangle \notin L$.

Из этого следует, что $L \notin \mathcal{RE}$ и в частности $L \notin \mathcal{R}$.

Непринадлежность $\mathcal{R}$ можно доказать проще: через теорему Райса. Пусть $\mathcal{L}$ это множество языков, таких, что в них нет слов простой длины. Очевидно, существует машина, принимающая язык из $\mathcal{L}$ – это машина, которая не принимает ни одного слова, $\varnothing$ лежит в $\mathcal{L}$. Очевидно, существует машина, принимающая язык не из $\mathcal{L}$ – это машина, которая принимает все слова, Σ^* не лежит в $\mathcal{L}$. Следовательно, язык $L_{\mathrm{in}\mathcal{L}}$, который равен исходному L, не является разрешимым.

Итого, $L \notin \mathcal{R}$, $L \notin \mathcal{RE}$, $L \in \mathrm{co}\mathcal{RE}$.

3: Исследовать язык $L = \{\langle M\rangle \,|\, \mathsf{Accept}(M)$ бесконечен$\}$ на принадлежность классам $\mathcal{R}$, $\mathcal{RE}$, co$\mathcal{RE}$.

Проще всего с принадлежностью $\mathcal{R}$ – это прямое применение теоремы Райса, здесь $\mathcal{L}$ это множество бесконечных языков, машина, принимающая все слова, принимает язык из множества, машина, не принимающая ни одного слова, принимает язык не из множества, значит L неразрешим.

С $\mathcal{RE}$ и co$\mathcal{RE}$ ситуация сложнее: не видно простых способов ни однозначно принять, ни однозначно отвергнуть.

Докажем непринадлежность $\mathcal{RE}$: $\mathsf{NotHalt} \leqslant^m L$. Определим $f((\langle M\rangle, \omega)) = \langle N\rangle$. Машина N на входе x

1) моделирует работу M на ω в течение $|x|$ шагов,

2) если M остановилась за $|x|$ шагов, отвергнуть x, иначе принять x.

Функция f вычислима, так как это снова кусок универсальной МТ. Докажем корректность сводимости:

– если $(\langle M\rangle, \omega) \in \mathsf{NotHalt}$, то M не останавливается на ω, тогда для любого входа x машина N на втором шаге получит результат “M не остановилась” и примет x, тогда $\mathsf{Accept}(N) = \Sigma^*$ – бесконечный язык, следовательно $\langle N\rangle \in L$;

– если $(\langle M\rangle, \omega) \notin \mathsf{NotHalt}$, то M останавливается на ω за некоторое число t шагов, тогда для любого входа x длины больше или равной t машина N на втором шаге получит результат “M остановилась” и отвергнет x, тогда $\mathsf{Accept}(N) = \Sigma^{t-1}$ – конечный язык, следовательно $\langle N\rangle \notin L$.

Значит $L \notin \mathcal{RE}$. Этого в том числе хватит, чтоб доказать $L \notin \mathcal{R}$.

Докажем теперь непринадлежность co$\mathcal{RE}$: $\mathsf{Halt} \leqslant^m L$. Определим $g((\langle M\rangle, \omega)) = \langle S\rangle$. Машина S на входе x

1) моделирует работу M на ω в течение $|x|$ шагов,

2) если M остановилась за $|x|$ шагов, принять x, иначе отвергнуть x.

Функция g вычислима. Докажем корректность сводимости:

– если $(\langle M\rangle, \omega) \in \mathsf{Halt}$, то M останавливается на ω за некоторое число t шагов, тогда для любого входа x длины больше или равной t машина S на втором шаге получит результат “M остановилась” и примет x, тогда $\mathsf{Accept}(S) = \Sigma^* \setminus \Sigma^{t-1}$ – бесконечный язык, следовательно $\langle S\rangle \in L$;

– если $(\langle M\rangle, \omega) \notin \mathsf{Halt}$, то M не останавливается на ω, тогда для любого входа x машина S на втором шаге получит результат “M не остановилась” и отвергнет x, тогда $\mathsf{Accept}(S) = \varnothing$ – конечный язык, следовательно $\langle S\rangle \notin L$.

Значит $L \notin$ co$\mathcal{RE}$. Этого (отдельно) также хватит, чтоб доказать $L \notin \mathcal{R}$.

Итого L не лежит ни в одном из классов $\mathcal{R}$, $\mathcal{RE}$, co$\mathcal{RE}$. Дополнительный вопрос: определить тот уровень иерархии, на котором лежит представленный язык.