

EXP-полные задачи

Валиев Айдар

Декабрь 2020

Аннотация

В этой работе будут рассмотрены общие понятия, относящиеся к **EXP**-полным языкам, некоторые примеры **EXP**-полных задач, а также приведены доказательства их принадлежности к **EXP**. Для одного языка будет доказана принадлежность к классу **EXP**-полных.

1. Введение

Начнем с определения сложностного класса **EXP**(**EXPTIME**):

$$\mathbf{EXPTIME} = \bigcup_{k=1}^{\infty} \mathbf{DTIME}(2^{n^k})$$

В число задач из этого класса входят множественные задачи определения выигрышной позиции в различных играх. В качестве примера можно привести обобщенные крестики-нолики, на вход дается бинарное представление размера стороны игровой доски, а также позиция на доске, которая не обязательно должна быть корректной (на доске могут стоять только крестики или только нолики). В этой задаче достаточно перебрать всевозможные позиции на доске, которых будет экспоненциальное от размера входа количество, по увеличению количества пустых клеток на доске и для каждой позиции найти ответ.

EXPTIME-трудные (**EXPTIME** – **hard**), так же, как и в случае других сложностных классов, включает в себя задачи, к которым может быть сведена за полиномиальное время любая задача из **EXPTIME**, то есть

$$\forall A \in \mathbf{EXPTIME}, B \in \mathbf{EXPTIME} - \mathbf{hard}$$

$$\exists f : x \in A \Leftrightarrow f(x) \in B$$

Построение такой функции называют сведением

И, соответственно, класс **EXPTIME** – **complete** представляет собой $\text{EXPTIME} \cap \text{EXPTIME} - \text{hard}$

Конкретные примеры **EXP**-полных

2. Некоторые **EXPTIME** – **complete** задачи

Как и многие другие "полные" классы, **EXPTIME** – **complete** представляет собой особый интерес, так как для доказательств различных гипотез, достаточно использовать лишь одну задачу из этого класса. Такие задачи называют "самыми сложными" в классе.

2.1. Задача останковки

Постановка задачи:

Задача L_1 состоит в определении того, завершит ли данная машина Тьюринга с данным входом, работу за данное количество шагов (количество шагов записано в бинарной форме). Ее обобщенный вариант об останковке (или зависании) машины Тьюринга с данным входом является неразрешимой, что было доказано Аланом Тьюрингом в 1936 году.

Доказательство полноты:

Чтобы проверить, что тройка (M, x, k) лежит в языке можно просто смоделировать k шагов машины M на входе x . Таким образом будет произведено $\mathcal{O}(2^k)$ шагов, значит $L_1 \in \text{EXPTIME}$.

Теперь рассмотрим некоторый язык $A \in \text{EXPTIME}$. Для него существует машина Тьюринга M , эффективно вычисляемая функция T такие, что $M(x)$ завершает работу не более, чем за $T(|x|)$ шагов, и $T(n) \leq 2^{n^t}$, для некоторого t . Тогда построим машину M' , которая возвращает 1, если M возвращает 1, иначе закидывается. В таком случае $(M, x) \in A \Leftrightarrow (M', x, T(|x|) + 1) \in L_1$. Значит $L_1 \in \text{EXPTIME} - \text{hard}$. Таким образом доказано, что $L_1 \in \text{EXPTIME} - \text{complete}$

2.2. Обобщенные шахматы

Постановка задачи:

В задаче L_2 дается число n , позиция на таблице $n \times n$ (у каждой стороны ровно один король), нужно определить, есть ли выигрышная стратегия у белых.

Чтобы показать, что $L_2 \in \mathbf{EXPTIME}$, можно построить орграф всех позиций, в котором будет $\mathcal{O}(2^n)$ вершин и для каждой определить, является ли она выигрышной, проигрышной или ведет в ничью (нужно посмотреть на метки сыновей, ничья может возникнуть, когда оптимальные ходы зацикливаются или был поставлен пат)

2.3. Игра на ограниченном графе

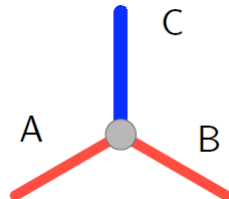
Необходимые определения:

- 1) Ограниченный граф - взвешенный ориентированный граф, удовлетворяющий следующему:

- Каждое ребро имеет вес 1 или 2 (далее красные и синие ребра, соответственно).
- Каждая вершина имеет вес 1 или 2 (на картинках будут обозначаться, как маленькие и большие круги, соответственно).

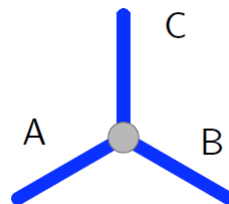
Выбор направлений всех ребер в графе будем называть допустимым положением, если для каждой вершины графа выполнено, что сумма весов входящих ребер не меньше веса самой вершины.

- 2) Вершина "И" - вершина веса 2, с которой смежны 2 красных ребра и одно синее.



Название объясняется тем, что если в допустимом положении синее ребро направлено от этой вершины, то оба красных ребра должны быть направлены в эту вершину.

- 3) Вершина "ИЛИ" - вершина веса 2, с которой смежны 3 синих ребра.



Аналогично с вершиной "И", если в допустимом положении одно из синих ребер направлено из вершины, то хотя бы одно из оставшихся направлено внутрь.

- 4) "И/ИЛИ" граф - ограниченный граф, каждая вершина которого является вершиной "И" или "ИЛИ"

Постановка задачи L_3 :

Дано:

- 1) "И/ИЛИ" граф $G(V, E)$, находящийся в допустимом положении
- 2) Разбиение множества ребер этого графа $(E_W, E_B) : E_W \cap E_B = \emptyset$ и $E_W \cup E_B = E$
- 3) $e_W \in E_W, e_B \in E_B$ - финальные ребра

Есть два игрока, белый и черный. Они по очереди, начиная с белого, ходят. Белый(черный) за ход переворачивает не более одного ребра из $E_{W(B)}$ (после поворота граф должен находиться в допустимом положении), если он перевернул ребро $e_{W(B)}$, то выигрывает. Задача состоит в определении, есть ли у белого игрока выигрышная стратегия.

$L_3 \in \mathbf{EXPTIME}$, аналогично обобщенным шахматам. Докажем, что $L_3 \in \mathbf{EXPTIME} - \mathbf{complete}$ окончательно

3. Доказательство трудности "игры на ограниченном графе"

Чтобы доказать принадлежность этой задачи к $\mathbf{EXPTIME} - \mathbf{hard}$, сведем к ней одну из логических игр G_6 , принадлежность которой к $\mathbf{EXPTIME} - \mathbf{hard}$ было доказано в 1976-ом году Стокмейером.

G_6 :

Дано: Логическая формула в КНФ F , зависящая от переменных X, Y (тут X и Y - наборы переменных), и начальные значения наборов X и Y .

Задача: Определить есть ли у первого игрока выигрышная стратегия. Два игрока ходят по очереди, на каждом ходу меняя значение не более одной переменной из X для первого игрока и Y для второго игрока. Первый игрок выигрывает, если после его хода F выполняется на текущих наборах X и Y .

Заметим, что в этой задаче нет варианта выигрыша второго игрока, все, что он может делать не давать выиграть первому.

Чтобы построение сведения было более легким, будем строить по формуле "И/ИЛИ" граф с дополнительными конструкциями, но перед этим покажем, что такой граф будет эквивалентен стандартному "И/ИЛИ" графу.

3.1. "Удобный" граф

"Удобным" графом будем называть граф, в котором помимо "И" и "ИЛИ" вершин разрешены такие конструкции, как:

- Терминальные ребра - ребра, на одном конце которых нет вершины. Также могут присутствовать фиксированные терминальные ребра - терминальные ребра, направление которых фиксировано и направлено от единственной смежной вершины.

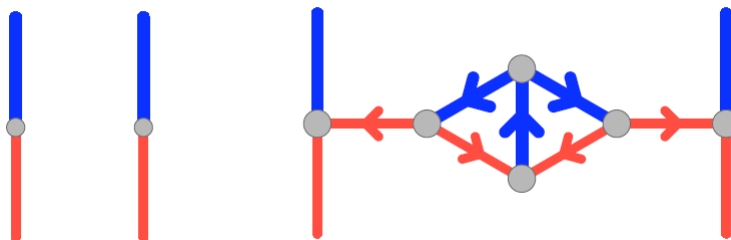
КАРТИНКА

- Вершины степени два.

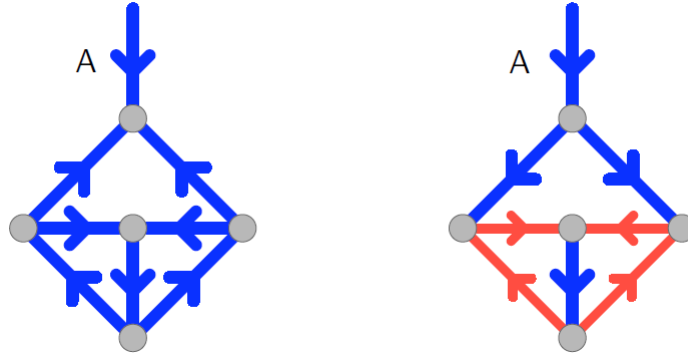
КАРТИНКА

Приведем, конструкции, допустимые в "И/ИЛИ" графе, которые эквивалентны приведенным выше особенностям "удобного" графа.

- 1) Для начала избавимся от (фиксированных) терминальных красных ребер. Добавим на свободный конец этого ребра вершину веса 2, из которой исходит (фиксированное) терминальное синее ребро
- 2) Заменяем вершины степени два с смежными ребрами одинакового цвета, на вершину веса 2, из которой дополнительно исходит терминальное синее ребро. Получится вершина "И" или "ИЛИ".
- 3) Чтобы заменить вершины степени два, смежные с красным и синим ребром, нужно заметить, что если в графе оставить только красные ребра, то граф будет состоять из изолированных вершин, простых циклов и простых путей. Тогда вершины степени два это концы простых путей, из чего следует, что их четное количество. Воспользуемся такой конструкцией:



- 4) Теперь осталось только исправить терминальные синие ребра таким образом



Легко заметить, что в первом случае соответствующее синее ребро может быть направлено в обе стороны, а во втором только в направлении от первоначальной вершины.

3.2. Сведение к "удобному" графу

Так как в построенном "удобном" графе у каждого игрока будет свое множество ребер, ориентацию которых он может менять, на рисунках будем отображать ребра белого(первого) игрока белой заливкой, а черного(второго) игрока черной.

Пусть есть некоторая формула F , зависящая от наборов переменных X и Y . Для каждой переменной $x \in X$ создадим такой граф:

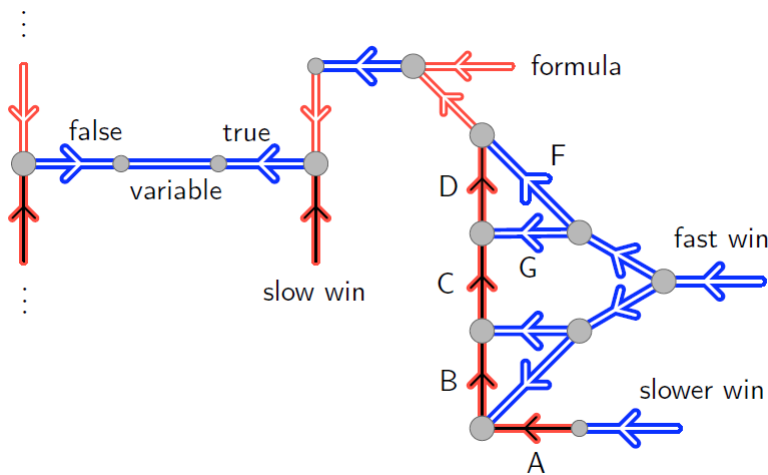
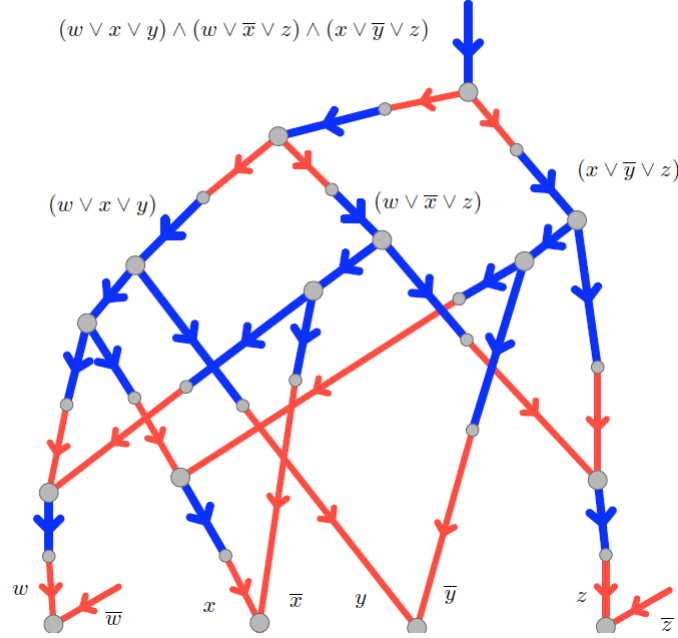


Рис. 1. Подграф для вершины

Левая часть, которая отсутствует на картинке, совпадает с правой частью. Для переменных $y \in Y$ единственным отличием будет то, что ребро соответствующее переменной (на рисунке variable) будет черным, а не белым. Ребро отмеченное, как formula направлена в часть графа, построенную из вершин "И" и "ИЛИ" и соответствующую F . Пример такой части графа:



Такую часть графа будем называть формальной. Все ребра белые, а на вершине конструкции находится ребро, которое ведет в финальное ребро белого игрока.

Ребра, помеченные на рис. 1 как "fast win", почти сразу ведут в финальное ребро белого игрока. Ребра, помеченные как "slow win", ведут по черным ребрам к финальному ребру черного игрока, при этом длина этого пути больше, чем количество ребер, которое надо перевернуть белому игроку в части графа, соответствующей формуле, чтобы дойти до финального ребра. Ребра, помеченные как "slower win", ведут по белым ребрам к финальному ребру белого игрока, но количество ребер, которое необходимо повернуть на пути, больше, чем на пути из "slower win".

3.3. Корректность сведения

Перед тем, как доказать, что построенное сведение корректно, заметим, что если в какой-то момент белый игрок повернет ребро, помеченное на

рис. 1 как "true" или "false" (он всегда может повернуть ровно одно из этих ребер, такое действие будем называть фиксацией переменной), то ему откроется проход к ребру "fast win", и единственный вариант для черного это повернуть ребро "D" соответствующей конструкции. Но, перед этим, надо повернуть ребра "C", "B" и "A", иначе белый может сразу пройти к ребру "fast win" и выиграть. Таким образом, после фиксации какой-либо переменной, черный обязательно поворачивает соответствующее ребро "A" (причем он не мог повернуть это ребро, до фиксации, так как в этом случае белый бы повернул ребро "slower win" и дошел бы до финального ребра, не давая черному возможности повернуть ребро "slow win"), что означает, что белый, фактически, не тратит ход на фиксацию. Белый, в свою очередь, может фиксировать какую-либо переменную только вместе со всеми, причем в этот момент набор значений переменных должен быть выполняющим для формулы, иначе черный пойдет по пути "slow win" а у белого не будет доступа к финальному ребру через формальную часть графа.

Также заметим, что до момента, когда белый игрок в первый раз зафиксирует переменную, все, что может делать черный игрок - поворачивать доступные ему ребра "variable" (белый игрок до фиксации переменных также будет поворачивать только ребра "variable")

Теперь покажем корректность сведения.

3.3.1. \Rightarrow

Пусть, у первого игрока в игре G_6 есть выигрышная стратегия. Будем повторять ее в игре на графе, поворачивая ребра "variable", соответствующие нужным переменным. Черный будет поворачивать доступные ему ребра "variable", что будет соответствовать действиям второго игрока в игре G_6 . Когда останется одно или ноль действий для белого игрока, он зафиксирует все переменные по очереди и начнет поворачивать ребра в формальной части и доберется до финального ребра раньше черного, так как путь из "slow win" до финального ребра для черного игрока дольше по построению.

3.3.2. \Leftarrow

Аналогичные рассуждения действуют и в обратную сторону. Вплоть до фиксации переменных повороты ребер будут соответствовать изменению значения переменных, и так как после фиксации белый доберется до финального ребра быстрее, значит, на момент фиксации полученный набор значений выполнял формулу. Соответственно первый игрок имеет выигрышную стратегию в игре G_6 . Корректность сведения, а, соответственно, и принадлежность данной задачи к классу **NP – complete** доказаны.

3.4. Заключение

Как можно было заметить, некоторые **EXPTIME** – **complete** задачи, такие как шахматы, шашки, го, и другие игры, знакомы многим людям. В этой работе помимо них были разобраны более нестандартные примеры задач из этого класса, а также было приведено достаточно красивое сведение для игры на "И/ИЛИ"-графе.

3.5. Источники

- 1) Games, Puzzles, and Computation by Robert Aubrey Hearn B.A., Rice University (1987) S.M., Massachusetts Institute of Technology (2001)
- 2) <https://en.wikipedia.org/wiki/EXPTIME>