

Master's Thesis

VR Sathiyandarayana ✉🏠
Chennai Mathematical Institute, India

Abstract

My master's thesis!

2012 ACM Subject Classification Replace ccsdesc macro with valid one

Keywords and phrases Logic

Digital Object Identifier 10.4230/LIPIcs.CVIT.2016.23

Acknowledgements I want to thank my advisors Dr. Paul Gastin and Dr. Aiswarya Cyriac for helping me through this project

1 Introduction

The separation property, invented by Dov Gabbay, is a strangely influential consequence of the design of popular temporal languages. Simply put, it requires all formulas in the language to be equivalent to a variant made up of formulas purely concerned with one *region* of the flow of time. Surprisingly, this property is linked to expressive completeness: a sufficiently expressive temporal language with the separation property can express any first-order property. In the next few sections, we will detail the separation property over linear time, its consequences on functional completeness, and discuss the generalization described in [4].

2 Preliminaries

Before discussing separation, we define some standard notions. A flow of time is simply a non-empty set T partially ordered by the binary relation $<$. We symbolically refer to these flows by the pair $(T, <)$. Examples include $(\mathbb{N}, <)$ and $(\mathbb{R}, <)$ with their natural ordering, unordered trees with the descendant relation, and Mazurkiewicz traces. We will consider the truth values of propositions (from a fixed set \mathcal{P}) at points on these flows.

The first-order vocabulary over these structures contains the ordering relation $<$ and a collection of *monadic* relations Q_1, Q_2, \dots that match the propositions q_1, q_2, \dots in \mathcal{P} . An assignment h of atoms in a time flow $(T, <)$ assigns to each Q_i a subset of T where the atom q_i is true. Augmented with the assignment, the triplet $(T, <, h)$ is called a *temporal structure*. First-order formulas are evaluated over these structures in the usual way. In this discussion, we pay special attention to first-order formulas with a single free-variable; they quite naturally mirror temporal formulas.

Instead of free variables and quantification, temporal languages employ *connectives* to reason through time. Popular connectives include F , P , G , H , U , and S , known as *future*, *past*, *globally*, *history*, *until* and *since* respectively. In this paper, we will limit our discussion to connectives that are definable by monadic first-order formulas.

Temporal formulas are evaluated at points in time. In a temporal structure $\mathcal{M} = (T, <, h)$, atoms are evaluated as

$$\mathcal{M}, t \models p \iff (T, <, h[x \mapsto t]) \models p(x) \iff t \in h(p)$$

As per the standard notation, the assignment $h[x \mapsto t]$ assigns the time point t to the first-order variable x . For a generic connective \sharp of arity n , let $\varphi_\sharp(t, X_1, \dots, X_n)$ be the



© Jane Open Access and Joan R. Public;
licensed under Creative Commons License CC-BY 4.0

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:9



Leibniz International Proceedings in Informatics

LIPICs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

monadic first-order formula defining it. Here, t is the point in time that the connective is evaluated at, and the X_i are monadic (second-order) variables. These variables expect a single-variable first-order formula, as shown below

$$\mathcal{M}, t \models \sharp(A_1, \dots, A_n) \iff (T, <, h[x \mapsto t]) \models \varphi_{\sharp}(x, \alpha_{A_1}, \dots, \alpha_{A_n})$$

Here, A_i are temporal formulas and α_{A_i} are their first-order translations. Notably, φ_{\sharp} can only quantify over elements in the domain T ; it cannot use second order quantifiers.

We illustrate this behaviour with an example. The connective F is defined by the formula

$$\varphi_F(t, X) \triangleq \exists x. (t < x) \wedge X(x)$$

Hence, we have

$$\mathcal{M}, t \models Fp \iff (T, <, h[x \mapsto t]) \models \exists y. (x < y) \wedge p(y)$$

We similarly define the other main connectives

$$\varphi_P(t, X) \triangleq \exists x. (x < t) \wedge X(x)$$

$$\varphi_G(t, X) \triangleq \forall x. (t < x) \wedge X(x)$$

$$\varphi_H(t, X) \triangleq \forall x. (x < t) \wedge X(x)$$

$$\varphi_U(t, X_1, X_2) \triangleq \exists x. [(t < x) \wedge X_1(x) \wedge \forall y ((t < y < x) \rightarrow X_2(y))]$$

$$\varphi_S(t, X_1, X_2) \triangleq \exists x. [(x < t) \wedge X_1(x) \wedge \forall y ((x < y < t) \rightarrow X_2(y))]$$

Note that, unlike the typical definition of U , φ_U doesn't rely on the present point t . Such an until is referred to in the literature by either the *strict* until (see [2]) or the *strong* until (see [1]). This particular behaviour makes observing separation much easier.

► **Definition 1** (Expressive Completeness). *A temporal language is **first-order expressively complete** over a class of time flows iff there exists a temporal formula A for any first-order formula with one free variable $\varphi(t)$ such that*

$$\mathcal{M}, t \models A \iff \mathcal{M}[x \mapsto t] \models \varphi(x)$$

*For any flow \mathcal{M} in the class. Relatedly, a flow of time $(T, <)$ is termed to be **expressively complete** if there exists an expressively complete temporal language over it.*

3 Linear Flows

In [4], Gabbay showed how the temporal language **L** with the strict until U and since S connectives satisfies the separation property over the integer time flow $(\mathbb{Z}, <)$.

To discuss this further, we need the notion of *regions* and *pure formulas*. Informally, the flow of time $(T, <)$ is partitioned into a set of regions. The positions of these regions depends on the position of the time point t where the temporal formula is being evaluated. For linear flows, Gabbay selected three regions:

- The *past* of t , formally defined as $\{x \mid x \in \mathbb{Z} \wedge x < t\}$.
- The *present*, which is simply $\{t\}$.
- The *future* of t , which naturally is $\{x \mid x \in \mathbb{Z} \wedge t < x\}$

Note that these regions are disjoint, and that the union of these regions produces the entire flow. Also, notice that these regions are first-order definable.

Now, we define *pure formulas*. For any flow $(T, <)$, we denote two assignments h and h' to be in agreement over a region $R \subset T$ iff for any atom $q \in \mathcal{P}$ and any point $s \in R$,

$$s \in h(q) \iff s \in h'(q)$$

Now, call a temporal formula A *pure* with respect to a region R iff* for any two assignments h and h' that agree on that region,

$$(T, <, h), t \models A \iff (T, <, h'), t \models A$$

In other words, A is true on h' iff A is true on h . We use the terms *pure past*, *pure present*, or *pure future* to denote pure formulas in the past, present, and future regions respectively.

Finally, call a formula A **separated** if it is a Boolean combination of pure formulas. Now, we can state the separation property

► **Theorem 2** (Separation Theorem). *Every temporal formula A in the language of S and U over linear time can be equivalently represented by a separated formula.*

The proof of this theorem is quite involved, and is presented in full detail in [4]. In the next few sections, I'll give a high-level overview of Gabbay et. al.'s scheme. To mirror their notation, I'll write U formulas as $U(p, q)$ instead of $q\mathcal{U}p$.

3.1 Separating S and U over linear time

It's simple enough to notice that, over linear time, a formula composed only of U s is a pure future formula, a formula composed of S s is a pure past formula. The task, therefore, is to transform formulas with both U s and S s.

Over the integer time flow $(\mathbb{Z}, <)$, these connectives naturally possess the following properties

$$\begin{aligned} U(\alpha \vee \beta, \gamma) &\equiv U(\alpha, \gamma) \vee U(\beta, \gamma) \\ U(\alpha, \beta \wedge \gamma) &\equiv U(\alpha, \beta) \wedge U(\alpha, \gamma) \end{aligned} \tag{1}$$

In addition, their negations can be productively rewritten as

$$\begin{aligned} \neg U(\alpha, \beta) &\equiv G(\neg\alpha) \vee U(\neg\alpha \wedge \neg\beta, \neg\alpha) \\ \neg S(\alpha, \beta) &\equiv H(\neg\alpha) \vee S(\neg\alpha \wedge \neg\beta, \neg\alpha) \end{aligned}$$

where the semantics of G and H are

$$\begin{aligned} \mathcal{M}, t \models G(\alpha) &\iff \mathcal{M}, t \models \forall t'. t' > t \rightarrow \varphi_\alpha(t') \\ \mathcal{M}, t \models H(\alpha) &\iff \mathcal{M}, t \models \forall t'. t' < t \rightarrow \varphi_\alpha(t') \end{aligned}$$

Here, φ_α is the first-order translation of α .

Our strategy involves *pulling-out* U s from inside S and vice versa. We accomplish this by writing all temporal formulas in a standard notation, and then applying a sequence of *elimination* rules. In the next section, we describe these rules.

3.1.1 Eliminations

Let α , β , φ and ψ be boolean combinations of propositional atoms. In the following subsections, we pull out a $U(\varphi, \psi)$ from inside a S under a variety of minimal configurations. In later sections, we show that these configurations suffice.

$$S(\alpha \wedge U(\varphi, \psi), \beta)$$

This formula requires $U(\varphi, \psi)$ to be true at a point t' in the past of t . This in turn implies φ at some point t'' ahead of t' . This naturally breaks down into three cases: $t'' > t$, $t'' = t$, and $t' < t'' < t$. The translation is

$$\begin{aligned} & S(\varphi \wedge \beta \wedge S(\alpha, \psi \wedge \beta), \beta) \\ \vee & (S(\alpha, \psi \wedge \beta) \wedge (\varphi \vee (\psi \wedge U(\varphi, \psi)))) \end{aligned}$$

$$S(\alpha \wedge \neg U(\varphi, \psi), \beta)$$

In this case, we immediately rewrite $\neg U(\varphi, \psi)$ as $G(\neg\alpha) \vee U(\neg\alpha \wedge \neg\beta, \neg\alpha)$. This gives us

$$\begin{aligned} & S(\alpha \wedge \neg U(\varphi, \psi), \beta) \equiv \\ & S(\alpha \wedge G(\neg\alpha), \beta) \\ \vee & S(\alpha \wedge U(\neg\alpha \wedge \neg\beta, \neg\alpha), \beta) \end{aligned}$$

where each individual case can be translated using the ideas used to rewrite $S(\alpha \wedge U(\varphi, \psi), \beta)$.

$$S(\alpha, U(\varphi, \psi))$$

It's instructive to recognize how $S(\alpha, U(\varphi, \psi))$ could be translated. Unlike the previous cases, the Until fragment needs to be true at each point in the path to α . This could involve multiple segments in this path where ψ is true till φ is true. Wonderfully, this is *indistinguishable* from the case where, at each point in the path, either φ or ψ is true. This formula is translated to

$$\begin{aligned} & S(\alpha, \perp) \\ \vee & S(\alpha, \varphi \vee \psi) \wedge [\varphi \vee (\psi \wedge U(\varphi, \psi))] \end{aligned}$$

Here, $S(\alpha, \perp)$ can only be true if α is true at the previous point. Otherwise, we'll need $U(\varphi, \psi)$ to be satisfied at the previous location, hence the $\varphi \vee (\psi \wedge U(\varphi, \psi))$ at the present. At each point t' in the path to α , if $t' + 1 \models \varphi$, $t' \models U(\varphi, \psi)$. Otherwise, $t' + 1 \models \psi$. At this point, we can use an inductive argument, starting from the previous point, to prove the correctness of this translation.

$$S(\alpha, \beta \vee U(\varphi, \psi))$$

The idea is to attempt to enforce $U(\varphi, \psi)$ at each point in the path *iff* we can detect an earlier point in the path which needed to satisfy it. A simple way to detect these points is to look for the moment where $\neg\beta$ was true, and check whether, along the way to that point, $\neg\varphi$ was true at each step. Accordingly, $S(\neg\beta \wedge \neg\alpha, \neg\varphi \wedge \neg\alpha)$ does the trick. Here, the $\neg\alpha$ is to ensure that we specifically look for points in the future of α , the leftmost point in our consideration.

It's important to recognize that we are capable of recognizing such points at each step of the path to α . This means that, if we recognized such a point that's 3 steps away, we recognized it at 2 and 1 step away too. This allows us a simple fix: $S(\neg\beta, \neg\varphi \wedge \neg\alpha) \rightarrow \varphi \vee \psi$. If φ was true, we will not see this point in our next search. Otherwise, ψ would be true, allowing for the possibility of enforcement in the future.

The overall translation now is

$$\begin{aligned} & S(\alpha, \neg\alpha \wedge (S(\neg\beta \wedge \neg\alpha, \neg\varphi \wedge \neg\alpha) \rightarrow \varphi \vee \psi)) \\ \wedge & S(\neg\beta \wedge \neg\alpha, \neg\varphi \wedge \neg\alpha) \rightarrow (\varphi \vee (\psi \wedge U(\varphi, \psi))) \end{aligned}$$

$$S(\alpha, \beta \vee \neg U(\varphi, \psi))$$

This case is very similar to the previous case. The points we search for must be in danger of satisfying $U(\varphi, \psi)$; hence, we look for $S(\neg\beta \wedge \neg\alpha, \psi \wedge \neg\alpha)$. We fix these points by requiring φ to be false. In the worst-case, we've dragged on the possible *until* to the present, at which point we can extinguish all hope. This gives us the overall translation:

$$\begin{aligned} & S(\alpha, \neg\alpha \wedge (S(\neg\beta \wedge \neg\alpha, \psi \wedge \neg\alpha) \rightarrow \neg\varphi)) \\ \wedge & S(\neg\beta \wedge \neg\alpha, \psi \wedge \neg\alpha) \rightarrow ((\neg\psi \wedge \neg\varphi) \vee (\neg U(\varphi, \psi))) \end{aligned}$$

$$S(\alpha \wedge U(\varphi, \psi), \beta \vee U(\varphi, \psi))$$

This is a neat combination of $S(\alpha \wedge U(\varphi, \psi), \beta)$ and $S(\alpha, \beta \vee U(\varphi, \psi))$. The translation is simple. *I believe Gabbay made a typo in this particular example. [7] mentions this.*

$$\begin{aligned} & S(\alpha, \psi) \wedge (\varphi \vee (\psi \wedge U(\varphi, \psi))) \\ \vee & S(\varphi \wedge S(\alpha, \psi), S(\neg\beta, \neg\varphi) \rightarrow \varphi \vee \psi) \\ \wedge & S(\neg\beta, \neg\varphi) \rightarrow (\varphi \vee (\psi \wedge U(\varphi, \psi))) \end{aligned}$$

3.1.2 Putting it all together

The eliminations presented in the previous section lend initial credence to the idea of separation. Amazingly, Gabbay presents a neat induction scheme that builds off these rules to separate *any* temporal formula in the language. In this subsection, we present an overview of his arguments (presented in more detail in [4]).

► **Lemma 3.** *Let α and β be pure-present temporal formulas and φ and ψ be formulas where the only appearance of U is $U(\alpha, \beta)$ and doesn't appear nested inside a S . Then $S(\varphi, \psi)$ can be written as a syntactically separated formula where the only appearance of U is $U(\alpha, \beta)$.*

Proof. We start by writing φ and ψ in their conjunctive and disjunctive normal form respectively. In this transformation, we treat all top-level instances of U and S in φ and ψ as atomic propositions. This gives us

$$\begin{aligned} \varphi &\equiv \bigvee_i (\varphi_{i,1} \wedge \varphi_{i,2} \wedge \cdots \wedge \varphi_{i,m_i}) \\ \psi &\equiv \bigwedge_i (\psi_{i,1} \vee \psi_{i,2} \vee \cdots \vee \psi_{i,n_i}) \end{aligned}$$

Now, we can use equation 1 to write $S(\varphi, \psi)$ as

$$\begin{aligned} S(\varphi, \psi) &\mapsto S\left(\bigvee_i (\varphi_{i,1} \wedge \varphi_{i,2} \wedge \cdots \wedge \varphi_{i,m_i}), \psi\right) \\ &\mapsto \bigvee_i S(\varphi_{i,1} \wedge \cdots \wedge \varphi_{i,m_i}, \psi) \\ &\mapsto \end{aligned}$$

◀

► **Lemma 4 (Lorem ipsum).** *Vestibulum sodales dolor et dui cursus iaculis. Nullam ullamcorper purus vel turpis lobortis eu tempus lorem semper. Proin facilisis gravida rutrum. Etiam sed sollicitudin lorem. Proin pellentesque risus at elit hendrerit pharetra. Integer at turpis varius libero rhoncus fermentum vitae vitae metus.*

Proof. Cras purus lorem, pulvinar et fermentum sagittis, suscipit quis magna.

■ **Listing 1** Useless code.

```
for i:=maxint to 0 do
begin
  j:=square(root(i));
end;
```

Just some paragraph within the proof. Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.

▷ **Claim 5.** content...

Proof. content...

1. abc abc abc



► **Corollary 6** (Curabitur pulvinar, [5]). *Nam liber tempor cum soluta nobis eleifend option congue nihil imperdiet doming id quod mazim placerat facer possim assum. Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed diam nonummy nibh euismod tincidunt ut laoreet dolore magna aliquam erat volutpat.*

► **Proposition 7.** *This is a proposition*

Proposition 7 and Proposition 7 ...

3.2 Curabitur dictum felis id sapien

Curabitur dictum Corollary 6 felis id sapien Corollary 6 mollis ut venenatis tortor feugiat. Curabitur sed velit diam. Integer aliquam, nunc ac egestas lacinia, nibh est vehicula nibh, ac auctor velit tellus non arcu. Vestibulum lacinia ipsum vitae nisi ultrices eget gravida turpis laoreet. Duis rutrum dapibus ornare. Nulla vehicula vulputate iaculis. Proin a consequat neque. Donec ut rutrum urna. Morbi scelerisque turpis sed elit sagittis eu scelerisque quam condimentum. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Aenean nec faucibus leo. Cras ut nisl odio, non tincidunt lorem. Integer purus ligula, venenatis et convallis lacinia, scelerisque at erat. Fusce risus libero, convallis at fermentum in, dignissim sed sem. Ut dapibus orci vitae nisl viverra nec adipiscing tortor condimentum [3]. Donec non suscipit lorem. Nam sit amet enim vitae nisl accumsan pretium.

3.3 Proin ac fermentum augue

Proin ac fermentum augue. Nullam bibendum enim sollicitudin tellus egestas lacinia euismod orci mollis. Nulla facilisi. Vivamus volutpat venenatis sapien, vitae feugiat arcu fringilla ac. Mauris sapien tortor, sagittis eget auctor at, vulputate pharetra magna. Sed congue, dui nec vulputate convallis, sem nunc adipiscing dui, vel venenatis mauris sem in dui. Praesent a pretium quam. Mauris non mauris sit amet eros rutrum aliquam id ut sapien. Nulla aliquet fringilla sagittis. Pellentesque eu metus posuere nunc tincidunt dignissim in tempor dolor. Nulla cursus aliquet enim. Cras sapien risus, accumsan eu cursus ut, commodo vel velit. Praesent aliquet consectetur ligula, vitae iaculis ligula interdum vel. Integer faucibus faucibus felis.

- Ut vitae diam augue.
- Integer lacus ante, pellentesque sed sollicitudin et, pulvinar adipiscing sem.
- Maecenas facilisis, leo quis tincidunt egestas, magna ipsum condimentum orci, vitae facilisis nibh turpis et elit.

► Remark 8. content...

4 Pellentesque quis tortor

Nec urna malesuada sollicitudin. Nulla facilisi. Vivamus aliquam tempus ligula eget ornare. Praesent eget magna ut turpis mattis cursus. Aliquam vel condimentum orci. Nunc congue, libero in gravida convallis [6], orci nibh sodales quam, id egestas felis mi nec nisi. Suspendisse tincidunt, est ac vestibulum posuere, justo odio bibendum urna, rutrum bibendum dolor sem nec tellus.

► **Lemma 9** (Quisque blandit tempus nunc). *Sed interdum nisl pretium non. Mauris sodales consequat risus vel consectetur. Aliquam erat volutpat. Nunc sed sapien ligula. Proin faucibus sapien luctus nisl feugiat convallis faucibus elit cursus. Nunc vestibulum nunc ac massa pretium pharetra. Nulla facilisis turpis id augue venenatis blandit. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus.*

Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.

5 Morbi eros magna

Morbi eros magna, vestibulum non posuere non, porta eu quam. Maecenas vitae orci risus, eget imperdiet mauris. Donec massa mauris, pellentesque vel lobortis eu, molestie ac turpis. Sed condimentum convallis dolor, a dignissim est ultrices eu. Donec consectetur volutpat eros, et ornare dui ultricies id. Vivamus eu augue eget dolor euismod ultrices et sit amet nisi. Vivamus malesuada leo ac leo ullamcorper tempor. Donec justo mi, tempor vitae aliquet non, faucibus eu lacus. Donec dictum gravida neque, non porta turpis imperdiet eget. Curabitur quis euismod ligula.

References

- 1 Michael Benedikt and Clemens Ley. Limiting until in ordered tree query languages. *ACM Trans. Comput. Logic*, 17(2), mar 2016. doi:10.1145/2856104.
- 2 Volker Diekert and Paul Gastin. Pure future local temporal logics are expressively complete for mazurkiewicz traces. In Martín Farach-Colton, editor, *LATIN 2004: Theoretical Informatics*, pages 232–241, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg.
- 3 Edsger W. Dijkstra. Letters to the editor: go to statement considered harmful. *Commun. ACM*, 11(3):147–148, 1968. doi:10.1145/362929.362947.
- 4 Dov M. Gabbay, Ian Hodkinson, and Mark Reynolds. *Temporal Logic (Vol. 1): Mathematical Foundations and Computational Aspects*. Oxford University Press, Inc., USA, 1994.
- 5 Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
- 6 John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space and related problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley, California, USA, October 13-15, 1975*, pages 57–64. IEEE Computer Society, 1975. doi:10.1109/SFCS.1975.23.

- 7 Maarten Marx. Conditional xpath, the first order complete xpath dialect. In *Proceedings of the Twenty-Third ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '04, page 13–22, New York, NY, USA, 2004. Association for Computing Machinery. doi:10.1145/1055558.1055562.

A Styles of lists, enumerations, and descriptions

List of different predefined enumeration styles:

```
■ \begin{itemize}...\end{itemize}
■ ...
■ ...

1. \begin{enumerate}...\end{enumerate}
2. ...
3. ...

(a) \begin{alphaenumerate}...\end{alphaenumerate}
(b) ...
(c) ...

(i) \begin{romanenumerate}...\end{romanenumerate}
(ii) ...
(iii) ...

(1) \begin{bracketenumerate}...\end{bracketenumerate}
(2) ...
(3) ...
```

Description 1 `\begin{description} \item[Description 1] ... \end{description}`
Description 2 Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.
Description 3 ...
 Proposition 13 and Proposition 13 ...

B Theorem-like environments

List of different predefined enumeration styles:

► **Theorem 10.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Lemma 11.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Corollary 12.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Proposition 13.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Conjecture 14.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Observation 15.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Exercise 16.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Definition 17.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Example 18.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Note 19.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Note.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Remark 20.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

► **Remark.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▷ **Claim 21.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

▷ **Claim.** *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.*

Proof. *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.* ◀

Proof. *Fusce eu leo nisi. Cras eget orci neque, eleifend dapibus felis. Duis et leo dui. Nam vulputate, velit et laoreet porttitor, quam arcu facilisis dui, sed malesuada risus massa sit amet neque.* ◀