Modelling Concurrent Systems Notes

Made by Leon :)

1 Process Algebras

Definition 1.1.1: ACP, CCS, CSP

The syntax of ACP, CCS, and CSP is defined as:

Operation	ACP	CCS	CSP
Termination	ϵ	0	STOP
Deadlock	δ		
Action	a		
Sequential Composition	$P.Q$		
Action Prefixing		$a.P$	$a \to P$
Alternative Choice	$P+Q$	$P+Q$	
External Choice			$P \Box Q$
Internal Choice			$P \sqcap Q$
Parallel Composition	$P \,\|\, Q$	$P \mid Q$	$P \,\|_A\, Q$
Restriction	$\partial_H(P)$	$P \backslash a$	
Abstraction	$\tau_I(P)$		P/a
Relabelling		$P[f]$	$P[f]$

Differences between ACP, CCS, and CSP

- **Action**: CCS and CSP require Action Prefixing, while ACP can perform sequential composition on any process. This also requires CCS and CSP processes to feature the inaction 0/STOP, while ACP is not restricted to this.
- **Choice**: ACP and CCS have an operator which can perform both External and Internal Choice. CSP Differentiates internal choices from external ones, and internal actions within $\Box$ do not count as a choice in CSP.
- **Parallel Composition**:
  - ACP actions follow a communication function to decide what to synchronise, i.e. $\gamma(a, b)$
  - CCS actions can only synchronise with its conjugate counterpart, i.e. a and $\overline{a}$
  - CSP actions can only synchronise over the same action, i.e. a and a
- **Restriction, Abstraction, Relabelling**:
  - Relabelling just doesn't exist in base ACP, lol
  - CCS combines communication and abstraction into one step - every synchronisation results in a τ.
  - CSP combines Parallel Composition and Restriction into one step, as CSP Parallel Composition doesn't feature left-over Left Merges.

Definition 1.1.2: The GSOS Format

General Structured Operational Semantics (GSOS) operations are compositional.
Rules of GSOS

- Its source has the form $f(x_1, \ldots, x_{ar(f)})$ with $f \in \Sigma$ and $x_i \in V$
- The left hand sides of its premises are variables x_i with $1 \leq i \leq ar(f)$
- The right hand sides of its positive premises are variables that are all distinct, and that do not occur in its source
- Its target only contains variables that also occur in its source or premises

GSOS Semantics of ACP

$$(a.P) \xrightarrow{\alpha} P \qquad P+Q \xrightarrow{\alpha} P \qquad P+Q \xrightarrow{\alpha} Q$$

$$\frac{P \xrightarrow{\alpha} P'}{P \,\|\, Q \xrightarrow{\alpha} P' \,\|\, Q} \qquad \frac{P \xrightarrow{a} P' \quad Q \xrightarrow{b} Q' \quad {\scriptstyle a|b=c}}{P \,\|\, Q \xrightarrow{a} P' \,\|\, Q'}$$

$$\frac{Q \xrightarrow{\alpha} Q'}{P \,\|\, Q \xrightarrow{\alpha} P \,\|\, Q'} \qquad \frac{P \xrightarrow{\alpha} P' \quad {\scriptstyle (\alpha \notin A)}}{\partial_H(P) \xrightarrow{\alpha} \partial_H(P')} \qquad \frac{\langle \mathcal{S}_X \mid \mathcal{S} \rangle \xrightarrow{a} P'}{\langle X \mid \mathcal{S} \rangle \xrightarrow{a} P'}$$

GSOS Semantics of CSP

$$(a \to P) \xrightarrow{a} P \qquad P \sqcap Q \xrightarrow{\tau} P \qquad P \sqcap Q \xrightarrow{\tau} Q$$

$$\frac{P \xrightarrow{a} P'}{P \Box Q \xrightarrow{a} P'} \qquad \frac{P \xrightarrow{\tau} P'}{P \Box Q \xrightarrow{\tau} P' \Box Q} \qquad \frac{Q \xrightarrow{a} Q'}{P \Box Q \xrightarrow{a} Q'}$$

$$\frac{Q \xrightarrow{\tau} Q'}{P \Box Q \xrightarrow{\tau} P \Box Q'} \qquad \frac{P \xrightarrow{\alpha} P'}{f(P) \xrightarrow{f(\alpha)} f(P')} \qquad \frac{P \xrightarrow{\alpha} P' \quad {\scriptstyle (\alpha \notin A)}}{P \|_A Q \xrightarrow{\alpha} P' \|_A Q}$$

$$\frac{P \xrightarrow{a} P' \quad Q \xrightarrow{a} Q' \quad {\scriptstyle (a \in A)}}{P \|_A Q \xrightarrow{a} P' \|_A Q'} \qquad \frac{Q \xrightarrow{\alpha} Q' \quad {\scriptstyle (\alpha \notin A)}}{P \|_A Q \xrightarrow{\alpha} P \|_A Q'}$$

$$\mu p.P \xrightarrow{\tau} P[\mu p.P/p]$$

Definition 1.1.3: CSP Expansion Theorem

Let $P := \sum_{i \in I} \alpha_i P_i$ and $Q := \sum_{j \in J} \beta_j . Q_j$. Then

$$P \mid Q = \sum_{i \in I} \alpha_i (P_i \mid Q) + \sum_{j \in J} \beta_j (P \mid Q_j) + \sum_{\substack{i \in I, j \in J \\ \alpha_i = \bar{b}_j}} \tau.(P_i \mid Q_j)$$

Any guarded CCS expression can be written into a bisimulation equivalent CCS expression of the form $\sum_{i \in I} \alpha_i . P_i$. This is called **head normal form**

Definition 1.1.4: CCS Axioms

- Axioms of CCS
$$(P+Q)+R = P+(Q+R) \quad \text{(associativity)}$$
$$P+Q = Q+Q \quad \text{(commutativity)}$$
$$P+P = P \quad \text{(idempotence)}$$
$$P+0 = P \quad \text{(0 is a neutral element)}$$
- Axiomatisation of Rooted Weak Bisimulation
$$\alpha.\tau.P = \alpha.P \quad \text{(T1)}$$
$$\tau.P = \tau.P + P \quad \text{(T2)}$$
$$\alpha.(\tau.P + Q) = \alpha.(\tau.P+Q) + \alpha.P \quad \text{(T3)}$$
- Axiomatisation of Branching Bisimularity
$$\alpha.(\tau.(P+Q)+Q) = \alpha.(P+Q) \quad \text{(P)}$$
- Axiomatisation of strong partial trace equivalence
$$\alpha.(P+Q) = \alpha.P + \alpha.Q$$
- Axiomatisation of weak partial trace equivalence
$$\tau.P = P$$

Definition 1.1.5: Axioms of ACP

is this really necessary.. who knows

2 Semantics and shit like that

Definition 2.0.1: Trace Semantice

- **Completed Trace**: A start to finish trace of a process.
- **Partial Trace**: From the start of a process to any point, including the end. Clearly, $CT(P) \subseteq PT(Q)$
- **Strong vs Weak**: Weak PT and CT means that two processes are equivalent with all instances of τ omitted.
- **Infinite Trace Semantics**: Differs from different types of divergence. Stronger than CT and PT

Definition 2.0.2: Bisimulation Semantice

- **True Bisimulation** ($\leftrightarrows$):
  - if sRt and $s \xrightarrow{a} s'$ then $\exists t'$ s.t. $t \xrightarrow{a} t'$ and $s'Rt'$
  - if sRt and $t \xrightarrow{a} t'$ then $\exists s'$ s.t. $s \xrightarrow{a} s'$ and $s'Rt'$
  - if sRt then $s \models p \iff t \models p$ for all $p \in P$
- **Branching Bisimilarity** ($=_{RBB}$)
  - if sRt and $s \xrightarrow{a} s'$ then either:
    - $a = \tau$ and $s'Rt$
    - $\exists t_1$, t' such that $t \Rightarrow t_1 \xrightarrow{a} t'$, sRt_1 and $s'Rt'$
  - if sRt and $t \xrightarrow{a} t'$ then either:
    - $a = \tau$ and $t'Rs$
    - $\exists s_1$, s' such that $s \Rightarrow s_1 \xrightarrow{a} s'$, s_1Rt and $s'Rt'$
  - if sRt and $s \models p$ then $\exists t_1$ s.t. $t \Rightarrow t_1 \models p$, and sRt_1
  - if sRt and $t \models p$ then $\exists s_1$ s.t. $s \Rightarrow s_1 \models p$, and s_1Rt
- Other notions:
  - **Rooted Branching Bisimilarity**: The same as Branching Bisimilarity except the first action is Strongly bisimilar. (This makes RBB a congruence on $+$)
  - **Delay Bisimilarity**: Same as *branching bisimilarity*, but with the requirements sRt_1 and s_1Rt dropped.
  - **Weak Bisimilarity**: The same as *delay bisimularity* except the action requirements are also relaxed:
    - If sRt and $s \xrightarrow{a} s'$ then either:
      - $a = \tau$ and $s'Rt$
      - $\exists t_1$, t_2, t' such that $t \Rightarrow t_1 \xrightarrow{a} t_2 \Rightarrow t'$ and $s'Rt'$
    - If sRt and $t \xrightarrow{a} t'$ then either:
      - $a = \tau$ and sRt'
      - $\exists s_1$, s_2, s' such that $s \Rightarrow s_1 \xrightarrow{a} s_2 \Rightarrow s'$ and $s'Rt'$

Definition 2.0.3: Compositionality, Congruence

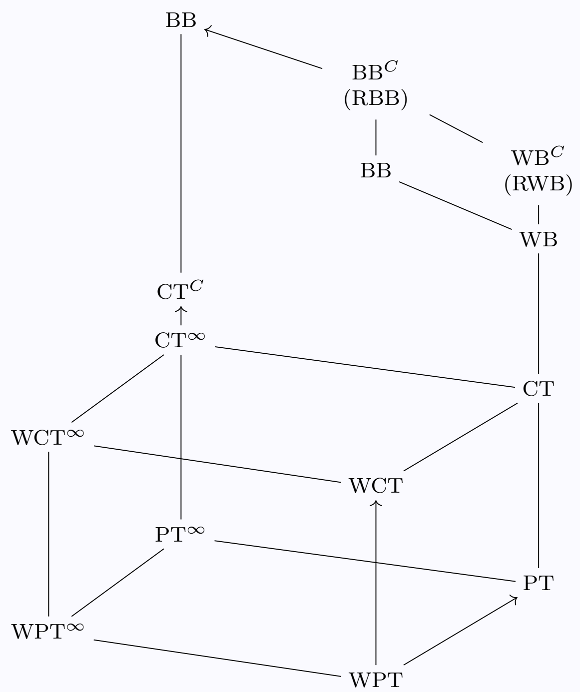
An equivalence $\sim$ is a **congruence**[a] for a language L if $P \sim Q$ implies that $C[P] \sim C[Q]$ for every context $C[\ ]$, where $C[\ ]$ is an L-expression with a **hole** in it, and $C[P]$ is the result of plugging in P for the hole. An alternative definition for a congruence $\sim$ is if every n-ary operator f of L, we have

$$P_i \sim Q_i \text{ for } i = 1, \ldots, n \text{ implies } f(P_1, \ldots, P_n) \sim f(Q_1, \ldots, Q_n)$$

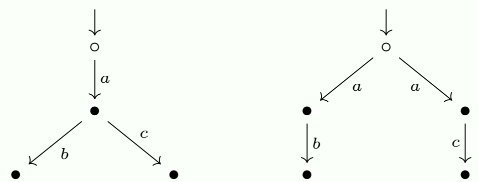
The **Congruence closure** of a language, denoted $\sim^c$ of a language is a modification to a language that isn't compositional to turn it compositional. The *congruence closure* of Branching Bisimilarity is Rooted Branching Bisimilarity.

[a] We can also say **the language is compositional for the equivalence**

Theorem 2.0.4: Semantic Equivalence Spectrum



Definition 2.0.5: The GOAT of Process algebra



Definition 2.0.6: Failure Semantics

Definition 2.0.7: Consistent Colouring

Definition 2.0.8: Safety

3 Other models of Concurrency

Definition 3.0.1: Hennessy-Milner Logic

The syntax of HML is given by:

$$\phi, \psi ::= \top \mid \bot \mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi \mid \langle\alpha\rangle\phi \mid [\alpha]\phi$$

Infinitary HML (HML^∞) has an infinitary conjunction: $\bigwedge_{i \in I} \phi_i$ HML in set form. If a process P has a property Φ, we write $P \models \Phi$.

- $P \models \top$
- $P \not\models \bot$
- $P \models \Phi \wedge \Psi$ iff $P \models \Phi$ and $P \models \Psi$
- $P \models \Phi \vee \Psi$ iff $P \models \Phi$ or $P \models \Psi$
- $P \models [K]\Phi$ iff $\forall Q \in \{P' : P \xrightarrow{a} P' \text{ and } a \in K\}.\ Q \models \Phi$
- $P \models \langle K\rangle\Phi$ iff $\exists Q \in \{P' : P \xrightarrow{a} P' \text{ and } a \in K\}.\ Q \models \Phi$

Deadlock can be represented as $P \models [\text{Act}]\bot$, where Act is the set of all actions.

Definition 3.0.2: Preorder

A **preorder** is a relation that is *transitive* and *reflexive*, but not *symmetric*. Preorders are denoted with $\sqsubseteq$.
Preorders are used just like equivalence relations to compare specifications and implementations. We write

$$\mathit{Spec} \sqsubseteq \mathit{Impl}$$

For each preorder $\sqsubseteq$, there exists an associated equivalence relation $\equiv$ called its **kernel**, defined by

$$P \equiv Q \iff (P \sqsubseteq Q \wedge Q \sqsubseteq P)$$

Definition 3.0.3: Simulation Equivalence

One process simulates the other when P can do all the same moves as Q. We write $P \sqsubseteq_S Q$ if Q can be simulated by P. Two processes are **simulation equivalent**, $P =_S Q$ if one simulates the other, and vice versa. This is two one-sided equivalences, and therefore is not the same as bisimulation, which needs both processes to be equivalent at the same time

Definition 3.0.4: Kripke Structure

Kripke Structures are defined on states rather than actions, called **atomic predicates**.
Let AP be a set of **atomic predicates**. A **Kripke structure** over AP is a tuple $(S, \rightarrow, \models)$ with S a set of states, $\rightarrow \subseteq S \times S$, the **transition relation**, and $\models \subseteq S \times AP$. The relation $s \models p$ says that predicate $p \in AP$ **holds in state** $s \in S$.

A **path** in a Kriple structure is a nonempty finite or infinite sequence $s_0, s_1, \ldots$ of states, such as $(s_i, s_{i+1}) \in \rightarrow$ for each adjacent pair of states s_i, s_{i+1} in the sequence. A path is **complete** if it is either infinite or ends in deadlock (a state without outgoing transitions)

Definition 3.0.5: Petri Nets

they exist

Definition 3.0.6: CTL

Computational Tree Logic is defined on

$$\phi, \psi ::= p \mid \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi \mid \phi \rightarrow \psi \mid$$
$$\mathbf{EX}\phi \mid \mathbf{AX}\phi \mid \mathbf{EF}\phi \mid \mathbf{AF}\phi \mid \mathbf{EG}\phi \mid \mathbf{AG}\phi \mid \mathbf{E}\psi\mathbf{U}\phi \mid \mathbf{A}\psi\mathbf{U}\phi$$

$p \in AP$ is an atomic predicate. CTL is defined on states, the relation $\models$ between states s in a Kripke structure and CTL formulae ϕ is inductively defined by

- $s \models p,\ p \in AP$ iff $(s, p) \in \models$
- $s \models \top$ always holds, and $s \models \bot$ never
- $s \models \neg\phi$ iff $s \not\models \phi$
- $s \models \phi \wedge \psi$ iff $s \models \phi$ and $s \models \psi$
- $s \models \phi \vee \psi$ iff $s \models \phi$ or $s \models \psi$
- $s \models \phi \rightarrow \psi$ iff $s \not\models \phi$ or $s \models \psi$
- $s \models \mathbf{EX}\phi$ iff there is a state s' with $s \rightarrow s'$ and $s' \models \psi$
- $s \models \mathbf{AX}\phi$ iff for each state s' with $s \rightarrow s'$ one has $s' \models \psi$
- $s \models \mathbf{EF}\phi$ iff some complete path starting in s contains a state s' with $s \models \phi$
- $s \models \mathbf{AF}\phi$ iff each complete path starting in s contains a state s' with $s \models \phi$
- $s \models \mathbf{EG}\phi$ iff all states s' on some complete path starting in s satisfy $s' \models \phi$
- $s \models \mathbf{AG}\phi$ iff all states s' on all complete path starting in s satisfy $s' \models \phi$
- $s \models \mathbf{E}\psi\mathbf{U}\phi$ iff some complete path π starting in s contains a state s' with $s' \models \phi$, and each state s'' on π prior to s' satisfies $s'' \models \phi$
- $s \models \mathbf{A}\psi\mathbf{U}\phi$ iff each complete path π starting in s contains a state s' with $s' \models \phi$, and each state s'' on π prior to s' satisfies $s'' \models \phi$

Definition 3.0.7: LTL

Linear-Time Temporal Logic is defined on

$$\phi, \psi ::= p \mid \top \mid \bot \mid \neg\phi \mid \phi \wedge \psi \mid \phi \vee \psi \mid \neg\phi \mid \phi \rightarrow \psi \mid$$
$$\mathbf{X}\phi \mid \mathbf{F}\phi \mid \mathbf{G}\phi \mid \psi\mathbf{U}\phi \mid$$

$p \in AP$ is an atomic predicate. The modalities X, F, G, U are called **next-state**, **eventually**, **globally**, and **until**. The relation $\models$ between paths and LTL formulae, with $\pi \models \phi$ saying that the path π satisfies the formula ϕ, or that ϕ is valid on π, or **holds** on π, is inductively defined by

- $\pi \models p,\ p \in AP$ iff $(s, p) \in \models$
- $\pi \models \top$ always holds, and $\pi \models \bot$ never
- $\pi \models \neg\phi$ iff $s \not\models \phi$
- $\pi \models \phi \wedge \psi$ iff $\pi \models \phi$ and $\pi \models \psi$
- $\pi \models \phi \vee \psi$ iff $\pi \models \phi$ or $\pi \models \psi$
- $\pi \models \phi \rightarrow \psi$ iff $s \not\models \phi$ or $\pi \models \psi$
- $\pi \models \mathbf{X}\phi$ iff $\pi' \models \phi$, where π' is the suffix of π obtained by omitting the first state
- $\pi \models \mathbf{F}\phi$ iff $\pi' \models \phi$ for some suffix π'
- $\pi \models \mathbf{G}\phi$ iff $\pi' \models \phi$ for each suffix π'
- $\pi \models \psi\mathbf{U}\phi$ iff $\pi' \models \phi$ for some suffix π' of π, and $\pi'' \models \phi$ for each path $\pi'' \neq \pi'$ with $\pi \Rightarrow \pi'' \Rightarrow \pi'$

Here a path is seen as a state, namely the first staet on that path, together with a future that has been chosen already when evaluating an LTL formula on that state. Traditionally, these judgements $\pi \models \phi$ were defined only for infinite paths π. When also applying them to finite paths, we only have to make one adaptation, namely $\pi \models \mathbf{X}\phi$ never holds if π has only one state. So $\mathbf{X}\phi$ says that there is a next state, and that ϕ holds in it.
$s \models \phi$ iff $\pi \models \phi$ for all complete paths π starting in state s. Here a path is **complete** if it is either infinite or ends in deadlock.

Lemma 3.0.8: Comparing LTL to CTL

LTL and CTL can be shown to be incomparable by proving that there cannot exist an LTL formula that is equivalent to the CTL formula $\mathbf{AGEF}a$, and by showing that there cannot exist a CTL formula equivalent to the LTL formula $\mathbf{FG}a$

Theorem 3.0.9: Satisfaction of Strong Bisimilarity

- Two processes are strongly bisimlar iff they satisfy the same infinitary HML formulas. Therefore, to show that two processes are not strongly bisimilar, it suffices to find an infinitary HML formula that is satisfied by one, but not the other.
- Two processes P and Q satisfy the same CTL formulas if they are strongly bisimilar. For finitely branching processes, we have “iff”. For general processes, we have “iff” if we use CTL with infinite conjunctions.
- Two divergence-free processes satisfy the same CTL_{-X} formulas if they are branching bisimulation equivalent. We have “iff” if we use CTL_{-X} with infinite conjunctions.

Theorem 3.0.10: De Nicola-Vaandrager

Translates Krikpe structures :D

Definition 3.0.11: Partition Refining

Partition refining is an algorithm to turn a process into its minimal state, making it easier to compare bisimularity. Works with

$$\texttt{split}(B, a, P)$$

where B is an equivalence class, a is an action, and P is the process. `split` splits an equivalence class into two, ones with the action and ones without it.

$$[\{s \xrightarrow{a} \bullet\}]_P$$

means “state s does action a outside the equivalence class in process P”

Listing 1: Pseudocode for `split`

```
split(B, a, P) → ({B_i} a set of blocks)
  choose s ∈ B
  B_1 = ∅ (B_1 contains states equivalent to s∗)
  B_2 = ∅ (B_2 contains states inequivalent to s∗)
  for each s' ∈ B do
  begin
    if [{s -a-> •}]_P = [{s' -a-> •}]_P then
      B_1 = B_1 ∪ {s'}
    else
      B_2 = B_2 ∪ {s'}
  end
  if B_2 = ∅ then
    return {B_1}
  else
    return {B_1, B_2}
```

Methodology:

- Start with one equivalence class for all states
- Run `split` on the outermost states, this now splits into R_1 and R_2, where R_2 are outside states
- Run `split` on states with outgoing actions to states in R_2, this now splits R_1 into R_1 and R_3, where R_3 are second-most outer states
- Keep on running until root state is partitioned
- If needed, the minimal graph will have exactly one of each equivalence class

---

Running partition refinement for Branching Bisimularity

- When checking whether a state in block B has an α transition to block C, it is okay if we can move through block B by doing only τ-transitions, and then reach a state with an α transition to block C
- We never check τ-transitions from block B to block B (τ-transitions to another block are fair game though)

This implies running the rule on

$$[\{s \Rightarrow\xrightarrow{a} \bullet\}]_P$$

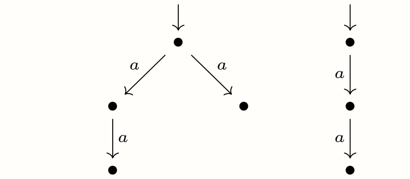
Definition 3.0.12: Alternating Bit Protocol

no idea this works tbh

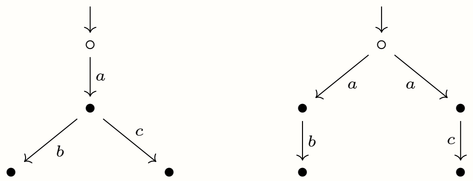
4 Example Catalogue

Example 4.0.1: Trace Semantics

A process that is PT equivalent but not CT equivalent

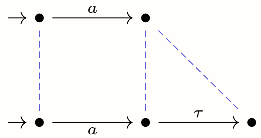


A process that is CT equivalent but not Bisimulation equivalent

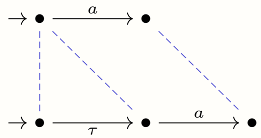


Example 4.0.2: Bisimulation Semantics

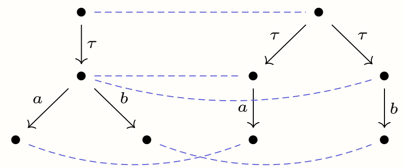
Two processes that are Branching Bisimulation equivalent



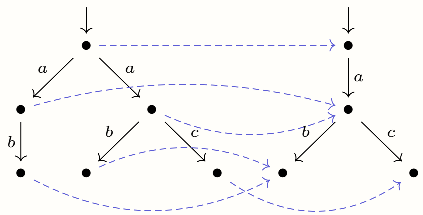
Two processes that are Branching Bisimulation equivalent but not Rooted BB equivalent

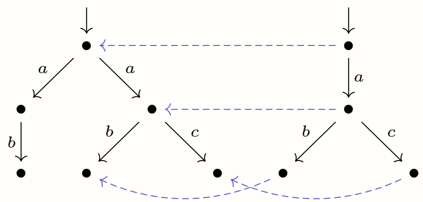


Two processes that are Weak Bisimulation equivalent but not Branching Bisimulation equivalent?



Two processes that are Simulation equivalent but not Bisimulation equivalent





Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasellus adipiscing semper elit. Proin fermentum massa ac quam. Sed diam turpis, molestie vitae, placerat a, molestie nec, leo. Maecenas lacinia. Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui.

Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Suspendisse vel felis. Ut lorem lorem, interdum eu, tincidunt sit amet, laoreet vitae, arcu. Aenean faucibus pede eu ante. Praesent enim elit, rutrum at, molestie non, nonummy vel, nisl. Ut lectus eros, malesuada sit amet, fermentum eu, sodales cursus, magna. Donec eu purus. Quisque vehicula, urna sed ultricies auctor, pede lorem egestas dui, et convallis elit erat sed nulla. Donec luctus. Curabitur et nunc. Aliquam dolor odio, commodo pretium, ultricies non, pharetra in, velit. Integer arcu est, nonummy in, fermentum faucibus, egestas vel, odio.

Sed commodo posuere pede. Mauris ut est. Ut quis purus. Sed ac odio. Sed vehicula hendrerit sem. Duis non odio. Morbi ut dui. Sed accumsan risus eget odio. In hac habitasse platea dictumst. Pellentesque non elit. Fusce sed justo eu urna porta tincidunt. Mauris felis odio, sollicitudin sed, volutpat a, ornare ac, erat. Morbi quis dolor. Donec pellentesque, erat ac sagittis semper, nunc dui lobortis purus, quis congue purus metus ultricies tellus. Proin et quam. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos hymenaeos. Praesent sapien turpis, fermentum vel, eleifend faucibus, vehicula eu, lacus.

Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Donec odio elit, dictum in, hendrerit sit amet, egestas sed, leo. Praesent feugiat sapien aliquet odio. Integer vitae justo. Aliquam vestibulum fringilla lorem. Sed neque lectus, consectetuer at, consectetuer sed, eleifend ac, lectus. Nulla facilisi. Pellentesque eget lectus. Proin eu metus. Sed porttitor. In hac habitasse platea dictumst.

Suspendisse eu lectus. Ut mi mi, lacinia sit amet, placerat et, mollis vitae, dui. Sed ante tellus, tristique ut, iaculis eu, malesuada ac, dui. Mauris nibh leo, facilisis non, adipiscing quis, ultrices a, dui.

Morbi luctus, wisi viverra faucibus pretium, nibh est placerat odio, nec commodo wisi enim eget quam. Quisque libero justo, consectetuer a, feugiat vitae, porttitor eu, libero. Suspendisse sed mauris vitae elit sollicitudin malesuada. Maecenas ultricies eros sit amet ante. Ut venenatis velit. Maecenas sed mi eget dui varius euismod. Phasellus aliquet volutpat odio. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia Curae; Pellentesque sit amet pede ac sem eleifend consectetuer. Nullam elementum, urna vel imperdiet sodales, elit ipsum pharetra ligula, ac pretium ante justo a nulla. Curabitur tristique arcu eu metus. Vestibulum lectus. Proin mauris. Proin eu nunc eu urna hendrerit faucibus. Aliquam auctor, pede consequat laoreet varius, eros tellus scelerisque quam, pellentesque hendrerit ipsum dolor sed augue. Nulla nec lacus.

Suspendisse vitae elit. Aliquam arcu neque, ornare in, ullamcorper quis, commodo eu, libero. Fusce sagittis erat at erat tristique mollis. Maecenas sapien libero, molestie et, lobortis in, sodales eget, dui. Morbi ultrices rutrum lorem. Nam elementum ullamcorper leo. Morbi dui. Aliquam sagittis. Nunc placerat. Pellentesque tristique sodales est. Maecenas imperdiet lacinia velit. Cras non urna. Morbi eros pede, suscipit ac, varius vel, egestas non, eros. Praesent malesuada, diam id pretium elementum, eros sem dictum tortor, vel consectetuer odio sem sed wisi. Sed feugiat. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Ut pellentesque augue sed urna. Vestibulum diam eros, fringilla et, consectetuer eu, nonummy id, sapien. Nullam at lectus. In sagittis ultrices mauris. Curabitur malesuada erat sit amet massa. Fusce blandit. Aliquam erat volutpat. Aliquam euismod. Aenean vel lectus. Nunc imperdiet justo nec dolor.

Etiam euismod. Fusce facilisis lacinia dui. Suspendisse potenti. In mi erat, cursus id, nonummy sed, ullamcorper eget, sapien. Praesent pretium, magna in eleifend egestas, pede pede pretium lorem, quis consectetuer tortor sapien facilisis magna. Mauris quis magna varius nulla scelerisque imperdiet. Aliquam non quam. Aliquam porttitor quam a lacus. Praesent vel arcu ut tortor cursus volutpat. In vitae pede quis diam bibendum placerat. Fusce elementum convallis neque. Sed dolor orci, scelerisque ac, dapibus nec, ultricies ut, mi. Duis nec dui quis leo sagittis commodo.